Travel Time Prediction for Urban Networks

Hao Liu

Delft University of Technology

21 October, 2008

Cover illustration: Hao Liu

Travel Time Prediction for Urban Networks

Proefschrift
ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 21 oktober 2008 om 10:00 uur
door

Hao LIU

Master of Science in Engineering, RIOH, Beijing, P.R. China
geboren te Dechang, Sichuan Province, P.R. China

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. H.J. van Zuylen
Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. H.J. van Zuylen	Technische Universiteit Delft, promotor
Prof. dr. ir. F.M. Sanders	Technische Universiteit Delft
Prof. dr. ir. S.P. Hoogendoorn	Technische Universiteit Delft
Dr. ir. J.W.C. van Lint	Technische Universiteit Delft, co-promotor
Prof. dr. ir. X. Chen	Southeast University, P.R. China
Prof. dr. ir. B. Immers	Katholieke Universiteit Leuven, Belgium
Dr. Y. Wang	Monash University, Australia
Prof. dr. Cees Witteveen	Technische Universiteit Delft

TRAIL dissertation Series no. T2008/12, The Netherlands TRAIL Research School

This dissertation is the result of a Ph.D. study carried out from November 2003 to June 2008 at Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Transport and Planning. The research was sponsored partialy by the Sino-Netherlands ITS training center and the ATMO research project, and partialy by the Research Institute of Highway, Ministry of Communications, P.R.China.

Published and distributed by:

TRAIL Research School
P.O. Box 5017
2600 GA Delft
The Netherlands
T: +31 (0) 15 278 6046
F: +31 (0) 15 278 4333
E: info@rsTRAIL.nl

ISBN: 978-90-5584-106-6

Printed in The Netherlands

Dedicated to my parents
Dekang and Chaolin

Preface

Five years ago, I faced a difficult choice of one of three universities where I would continue my Ph.D. study. Now, I am very proud that I made a wise choice of Delft University of Technology. Also, I would like to thank my dear promoter, Professor Henk van Zuylen because of your choice of me as a Ph.D. student.

As I approach the end of my Ph.D. study, which has been the toughest but yet the most enjoyable work of my life, I do realize that I have absorbed a great deal of knowledge and experience. Henk, you taught me how to work independently and gave me the freedom to formulate my research approach. The most valuable gift I obtained during these years is not only this dissertation but the way to do the 'real' scientific research. From your personality, I also learned how to operate a research group. You are a good example for my future career. I cannot find a better word to express my special gratitude in English. Thus, I quote a Chinese sentence in Pinyin "Yi Ri Wei Shi, Zhong Sheng Wei Fu". As well, I would like to thank my daily supervisor, Hans van Lint, for his daily support and effort of reading and reviewing my papers and chapters. The most impressive thing I have learned from you is that conducting good research and making wonderful music are possible at the same time.

Furthermore, I thank all my colleagues at the Transport and Planning Department. With your effort, I do really enjoy the friendly and happy research atmosphere. I thank Professor Piet Bovy for reviewing my papers and providing me advice. Also, many thanks go to all the supporting staff, especially Nicole Fontein, Cees Landman for their support. Here, I have to emphasize my dear roommates, Francesco, Saskia and Huizhao. With your smiles, jokes and talk, I always have a sunny mood even if it was rainy. Very special thanks should be sent to Adam and Pascal for your translation. I also thank all the table tennis people, Chris, Enide, Victor, Theo and many more.

I would like to thank all the committee members for their time and effort, useful comments and suggestions, and their approval of my draft dissertation.

I am particularly grateful to my Chinese colleagues at Research Institute of Highway, Professor Xiaojing Wang, Ke, Tongyan, Jiantong, and many more.

I am grateful to all my Chinese friends in the Netherlands. Huizhao&Hao Li, no more complimentary words than I am so lucky to meet you and have much wonderful time with you. Guoping, likes my elder brother, always pays the bill for coffee and drives me to wherever I need to go. Xiaohui&Xin, provided me many loving memories, particularly when we crowded into your room, hung out in a bar, and traveled on a train for fun. Lao Yuan and your wife, thanks for your help in dealing with many Dutch affairs and super

delicious dinners at your home. Thanks also go to Yusen, Xiong, Gang, Ming, Peng, Minwei, Yufei and many more.

A very special thank goes to my academic sister, Wei. The stimulation story you made always encourages me during tough time.

Last but not least, I would like to dedicate this thesis to my parents for their love and support, and my dear fiancee F.F.Y for your tolerance and encouragement during my hard times. A Chinese saying in Pinyin ”Wu Sheng Sheng You Sheng” expresses all my thanks. You will see from my activities how much I appreciate.

Hao Liu

Beijing, September 2008

Contents

List of Figures

List of Tables

Notations

The main symbols, variables, parameters and abbreviations that are used in this dissertation are presented as follows:

Statistical symbols and parameters

MARE	:	Mean Absolute Relative Error
MSE	:	Mean Squared Error
RMSE	:	Root Mean Squared Error
RMSEP	:	Root Mean Squared Error Proportional
SSE	:	Sum Squared Error

General abbreviations

ATIS	:	Advanced Traveler Information Systems
RGS	:	Route Guidance Systems
FCD	:	Floating Car Data
VMS	:	Variable Message Sign
GPS	:	Global Position System
AVI	:	Automatic Vehicle Identification
MDP	:	Markov Decision Process
HCM	:	Highway Capacity Manual
KNN	:	K-Nearest Neighbor
GIS	:	Geography Information System
SSNN	:	State Space Neural Network
ANN	:	Artificial Neural Network
USEG	:	Urban Segment Travel Time Prediction Model
UROU	:	Urban Route Travel Time Prediction Model
FNN	:	Feed-forward Neural Network
LM-BR	:	Levenberg-Marquardt and Bayesian Regulation
EKF	:	Extended Kalman Filter
GPRS	:	General Packet Radio Service
ANPR	:	Automatic License Number Plate Recognition
TTV	:	Travel Time Variability
DMI	:	Distance Measuring Instrument
LPM	:	License Plate Matching
ETC	:	Electronic Toll Collection
PM	:	Platoon matching
CPT	:	Cellular Phone Tracking
AS	:	Aerial Survey

Symbols, variables and parameters

TT	:	travel time
TTV	:	travel time variability
T_f	:	travel time in free flow conditions
$TT90th$, $TT10th$	:	90th and 10th percentile travel time
p	:	departure time period
k	:	time step
L	:	length of the route of interest
t, t', t^*	:	time or departure time
$TT_i^d(t)$	:	departure travel time of individual i
$TT_i^a(t)$	:	arrival travel time of individual i
x	:	space
$x_i(t)$	:	location of individual i as a function of time
v	:	speed
v_s, v_t	:	space mean speed and time mean speed
σ_s	:	the variance of space mean speed
v_f	:	speed in free flow conditions
$v_i(t)$	:	speed of individual i as a function of time
$\gamma_a(t), \gamma_d(t)$	:	acceleration and deceleration rate of individual i
$t_i(x)$	:	time of individual i as a function of space
r	:	index for route
q	:	flow rate
q^{in}, q^{out}	:	inflow and outflow rate
D	:	delay
d_s	:	stop delay
d_d	:	deceleration delay
d_1	:	uniform delay
d_2	:	incremental delay
d_3	:	initial queue delay
s_a	:	saturated flow rate
c_y	:	cycle time
g_e	:	effective green time
C_a	:	capacity of route or intersection approach
q	:	vehicle arrival flow rate
χ	:	volume-to-capacity ratio
f_r	:	adjustment factors for the quality of progression
f_t	:	adjustment factor for residual delay component
f_{PF}	:	adjustment factor for the platoon arriving in green interval
I	:	adjustment factor for upstream filtering/metering
T	:	evaluation period
N_q	:	the number of vehicles in a queue
L_q	:	the length of a queue
k_m	:	jam density
ϕ	:	occupancy
P_d	:	the percentage of green time at the upstream signal
P_u	:	the percentage of green time at the downstream signal
P_s	:	the ratio of the detector setback distance to the link length

$G(.)$	:	data driven model
U	:	the input of neural networks
X	:	the input or influencing factors
Y	:	the output of neural networks
S	:	the hidden layer of neural networks
$\mathbf{\Omega}$	:	data set
$\rho_{\min}, \rho_{\max}$	:	the minimal and maximal value of data set $\mathbf{\Omega}$
W	:	parameter (weight) vector
w	:	weight parameter
E	:	error vector
ε	:	error
$\mathbf{\Phi}$	:	transfer function
J	:	Jacobian matrix
H	:	Hessian matrix
$\alpha_l, \alpha_t, \alpha_r$	:	left-turning, throughput and right turning fractions

Chapter 1

Introduction

1.1 Background

Travel time is an important aspect of transportation. It reflects the performance of road networks and has a direct meaning for all people. Travel time is of interest for both the individual traveler and for the network authorities who manage the road networks. Providing travel time information to road users allows them to make more informed decisions on their choices (e.g. mode, route and departure time) (Bovy & Stern 1990). These individual behavioral changes have a positive impact on the network as a whole (potentially reduce congestion and improve network efficiency) (Ben-Akiva et al. 1991). From the perspective of the whole network, the desire of travelers to reduce their travel times over a road network has led to a necessity for network authorities to reduce the overall travel time on the road network.

Moreover, accurate and reliable travel time predictions can also benefit the road users by decreasing uncertainty and reducing stress (Adler & Blue 1998). Clearly, travel time information is becoming increasingly important for a variety of real-time transportation applications, such as Advanced Traveler Information Systems (ATIS), Route Guidance Systems (RGS), etc. As a direct result, refining or creating new travel time prediction models used in real time operations is a research area with growing interests (Paterson 2000).

Travel time prediction for *freeways* has been studied intensively in the past decade (e.g. Van Lint 2004, Vanajakshi 2004, Paterson 2000, and Bovy & Thijs 2000). However, limited work has been done for *urban networks* (Lin et al. 2004, Robinson 2005). This Ph.D. research concentrates on travel time prediction for urban networks based on the analysis of the differences between freeways and urban networks.

Traffic flows on freeways are often treated as uninterrupted flows. However, the traffic flows on urban networks are considered as interrupted flows. Vehicles traveling on urban networks are subject to not only queuing delays but also signal delays as well as delays caused by vehicles entering from the cross streets (Lin et al. 2004). The mechanism of traffic propagation along urban networks is quite different from that on freeways. Since freeway traffic operations are considerably different from urban traffic operations, travel

time prediction models for freeways are not directly usable in an urban context. This motivates our intention to create a model which can be applicable in an urban context.

From the practical application point of view, the availability of traffic data will influence the methodologies used for urban travel time prediction. Besides the differences in traffic operations, urban networks differ also in terms of the available data from sensors. Urban networks are usually not as comprehensively covered by measurement equipments as freeway networks are (Van Lint 2004). A shortage of data slows down the development of models for urban networks since few data are available for the calibration and validation of these models.

Presently, the monitoring of traffic flows on urban networks is enhanced through the use of cameras that recognize license plate numbers. More and more cities, such as Beijing (China), Delft (The Netherlands) and Stockholm (Sweden), have installed license plate cameras for monitoring large-scale urban networks. Those direct travel time measurements are available for calibrating and validating models for the prediction of travel times. Utilizing these new travel time measurements along with traditional loop detector measurements for travel time prediction on urban arterials is the main motivation for the research described in this dissertation.

Another potential data resource is the Globle Position System (GPS). With the successful commercial marketing, GPS equipments on taxies can provide tremendous amounts of traffic data covering the whole urban network. These data, so called floating car data (FCD), can be a good complementary resource in the future. However, this thesis will not take the use of FCD into account.

Urban networks are usually equipped with single loop detectors, while double loop detectors are often installed on freeways. The single loop detectors are only able to provide volumes and occupancies, unlike the double loop detectors which also provide speeds. Moreover, freeways usually have a more uniform spacing of detector locations (e.g. 500 meters in the Netherlands), but for urban streets the detector locations vary with the length of the urban links.

In this Chapter, the background of this Ph.D. research is described in order to identify the intention of this effort. Next, research objectives are presented and the research scope is defined in order to describe a clear boundary of the research problem. Then, the main contributions and relevance of this dissertation are summarized. The final section outlines the structure of the entire dissertation.

1.2 Research Objectives and Scope

In this section, research objectives and scope will be elaborated in order to make a clear description of the urban travel time prediction problem.

Travel times are the results of traffic flow operations, which in turn are governed by the interactions between traffic demand (e.g. commute traffic demand, etc.) and traffic supply characteristics (e.g. road capacity, traffic signal timing, weather, etc.). In other words, traffic conditions result in different travel times. The traffic conditions are governed by

complex non-linear interactions of heterogeneous groups of driver-vehicle-road combinations. The drivers, vehicles and roads are characterized by their own specific technical and behavioral properties, such as vehicle dimensions and acceleration characteristics, drive-style (aggressive, conservative), and traffic signal controls. Predicting travel times requires predicting traffic conditions in the future. However, in a practical situation we only know the traffic conditions up to the current time instant. Thus, predicting future traffic conditions is a major issue of addressing the travel time prediction problem. For applicability purposes, not all (inter)relationships between these demand and supply factors will be taken into account.

1.2.1 Research Objectives

The main objective of this dissertation is to develop a methodology that can provide robust and accurate travel time predictions for urban networks.

First, the *accuracy of* travel time predictions is an important criterion for evaluating the proposed model. The smaller the difference between predicted travel times and actual travel times, the better the model performs. To quantify this, a simple baseline model, which is widely used in practice, will be compared with the proposed model. The baseline model simply uses measured travel times as predicted travel times (see details in Chapter 5). Obviously, the purpose is to develop a model which outperforms the baseline model. In addition, one or more existing models will be used for comparison.

Secondly, the proposed model should be *robust.* The robustness requires the proposed model to be capable of coping well with variations in its operating environment. Robustness is a quality which is difficult to assess quantitatively. In this dissertation, robustness is defined two-fold: the ability of coping with the variations of traffic conditions (free flow and congestion); and the ability of coping with the variations of the quality of input data. The former is assessed by comparing the accuracy of travel time predictions under different traffic conditions. The later is addressed by assessing how well the model developed in this research with corrupted and missing (bad quality) data, which are common problems in a real time situation.

1.2.2 Research Scope

Travel time prediction is a broad research problem. In the literature, a wide variety of travel time prediction models have been developed. Those models can be classified in terms of road type, spatial scope, prediction horizon, input traffic data, etc. In this research, the scope will be limited in the following way.

First, this research only focuses on *signalized urban arterials*. Unsignalized urban streets are not considered. Also, urban streets with roundabouts are not addressed. The spatial scope and road type will be restricted to route level and signalized urban networks (details of the definitions of the link, segment, route, network will be presented in Chapter 2). Of course, the concept of predicting urban route travel time can be easily extended to a network level by disaggregating the network into routes. Thus, from the spatial application perspective, this research will focus on signalized urban routes.

Although there are a lot of factors influencing urban travel times, such as road geometry, public transit, traffic composition, weather, the focus here is on two important factors, namely, *traffic demand* and *intersection control*. Apparently, travel times are strongly influenced by traffic demand. A common experience is that drivers take long travel time with the increase of traffic volumes. As mentioned above, the interrupted operation of urban arterials causes stochastic delays, which constitute a large part of travel time on urban streets (Viti & Zuylen 2004).

The availability of data dominantly influences the modeling approach and methodology which will be used for travel time prediction. Single loop detectors are the common detection equipments installed on urban networks. In most Dutch cities, single loop detectors are installed at signalized intersections for the vehicle actuated signal control. Those single loop detectors are located at either just upstream of the stop line or even further upstream. Those single loop detectors provide *volume* data. The *signal timing* data can be obtained from signal controllers. *Measured travel times* are provided from license plate camera systems. Note that the layout of detectors might be different for networks with different traffic control methods, e.g. SCOOT (Hunt et al. 1981), SCATS (Wilson et al. 2006), UTOPIA (www.peektraffic.nl), etc. In this research, we use the Dutch layout of detectors.

Finally, this research focuses on *short term* travel time prediction. In general, short term travel time prediction is used for real time route guidance, while long term travel time prediction is widely used for transport planning purpose. Obviously, the longer the prediction horizon, the more models rely on either statistical or theoretical assumptions regarding future traffic conditions (Van Lint 2004). In fact, there is no a clear boundary between the short term and the long term. In this dissertation, 30 minutes is selected as the maximal prediction ahead for testing the performance of the proposed model because most urban trips are shorter than 10km, a distance that can be traveled within 30 minutes.

1.3 Contributions and Relevance of the Dissertation

The main contributions of this dissertation are listed below:

1. It develops a neural network based traffic flow model for urban route travel time prediction (Chapter 4). The approach is a hybrid of data-driven and model-based approaches. The concept of a hybrid model for a neural network is applied for the first time in this dissertation. Some of the innovative aspects of this model are specified as follows:

    - A single segment model based on the State Space Neural Network is developed for modeling the traffic flow on a single signalized segment. The segment model is generic and not location-specific, at least in terms of its mathematical structure and the overall input-output relationship.
    - The segment based model can be extended to predict urban travel times from segment level to route level. The extension can be accomplished by concatenating separate segment models which correspond to the segments of the route

of interest. The traffic flow on the urban routes are modelled by propagating from (upstream) segments to (downstream) segments.

- The model is able to take traffic signal timings into account.

2. Two strategies have been proposed to preprocess raw data. A procedure of dealing with corrupted volume data collected by single loop detectors has been proposed (Chapter 6). In addition, a method to detect the outliers of travel time observations, and then fill in the empty gaps, has been developed (Appendix E).

1.3.1 Scientific Relevance

The use of neural networks for freeway travel time prediction is certainly not new. But, the applications of neural networks on urban networks have not been found in the literature. A neural network based traffic flow model for urban route travel time prediction, as described in this dissertation, has been presented. By considering the main features of urban networks, the proposed model decomposes modelling complex urban networks into modelling urban segments and concatenating them afterwards. Modelling travel time on urban segments is based on a neural network, while the concatenation of those models is based on traffic flow theory. In this sense, it shows that the domain knowledge (in our case traffic flow theory) can be integrated into neural networks. It is a promising attempt to combine a data-driven method with a model-based approach.

1.3.2 Practical Relevance

The research results can be applied for Advanced Traveler Information Systems (ATIS) and Route Guidance Systems (RGS), which are two parts of the Intelligent Transportation Systems (ITS). In Beijing situation, for example, an increasing number of variable message signs (VMSs) have been set up at strategic bifurcations in the urban network, where drivers could choose alternative routes. In addition, two real-time traffic information websites (one is based on loop detector data, the other is based on floating car data) are open to the public. Now, both the VMSs and the two websites only display speeds in colors (red represents congestion, yellow represents slight congestion, green represents free flow). Those displayed speeds only represent rough traffic states in the urban network. However, there is a clear need for accurate travel time predictions, which can be displayed on the VMSs, online websites, or in-car navigation systems. The practical relevance of this dissertation is to develop a urban travel time prediction model, which is applicable in a real-time environment.

We demonstrate how the proposed model can be deployed in a single loop detector based data collection system. Since the proposed model is essentially a data driven approach, it could also be applied for other traffic data collection systems which measure other quantities that are physically or statistically related to travel times. For example, more and more cities in China build up floating car data collection systems (based on GPS and GSM), and automatic vehicle identification systems (AVI).

In practice, missing and corrupted data frequently occur. There is a need for dealing with the quality of loop detector data and measured travel times. Two methods have been

presented to address the problem of raw data. Certainly, the methods can be used by other researchers who need a good-quality database from loop detectors and license plate recognition systems.

1.4 Dissertation Outline

Apart from the introduction in Chapter 1, the dissertation is composed of the following chapters and appendixes. Figure 1.1 shows the entire structure of this dissertation.

Chapter 2 gives some definitions of travel times (the theme of this dissertation) and urban networks (the application focus). Those definitions will be used through the whole dissertation. Since delays dominate the main part of urban travel times, a detailed explanation of the delays caused by different sources is included. The complexity of many influencing factors results in the variability of urban travel times. In the end of this Chapter, some basic relationships between travel times and traffic quantities are presented.

Chapter 3 provides a comprehensive overview of existing travel time prediction approaches solely for urban networks. Those approaches are categorized into two groups: model based and data driven. The advantages and disadvantages of each approach will be discussed. Based on this overview, a state space neural network for predicting urban travel times is proposed.

Chapter 4 describes the development of the proposed model for predicting urban travel times. First, a basic and generic model is developed at the urban segment level. Then, it shows how to concatenate the generic models to predict travel times at urban route level.

Chapter 5 applies the proposed model in a simulation environment. Three typical traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions) have been generated to test this proposed model. This Chapter also addresses the issues of sensitivity and robustness.

Chapter 6 uses empirical date obtained from RegioLab Delft project (www.regiolab-delft.nl) to assess the proposed model. The practical applications, not like simulations (100% correct data), require extensive strategies to deal with the quality of actual observations. This Chapter presents methods to generate good quality data. Then, those data will be used for test the proposed model.

Finally, Chapter 7 summarizes the main conclusions of this research and offers the directions of future research.

Appendix A lists common performance indicators used in Chapter 5 and 6.

In this dissertation, the license plate recognition system is used as a travel time collection system. To have an overviews of various travel time collection systems, readers can find in Appendix B.

In Chapter 2, it is stated that the locations of loop detectors do have a significant influence on measured mean speeds, which can be used for deriving travel times. Details of the analytical calculation of mean travel times and time/space mean speeds with respect to locations can be found in Appendix C.

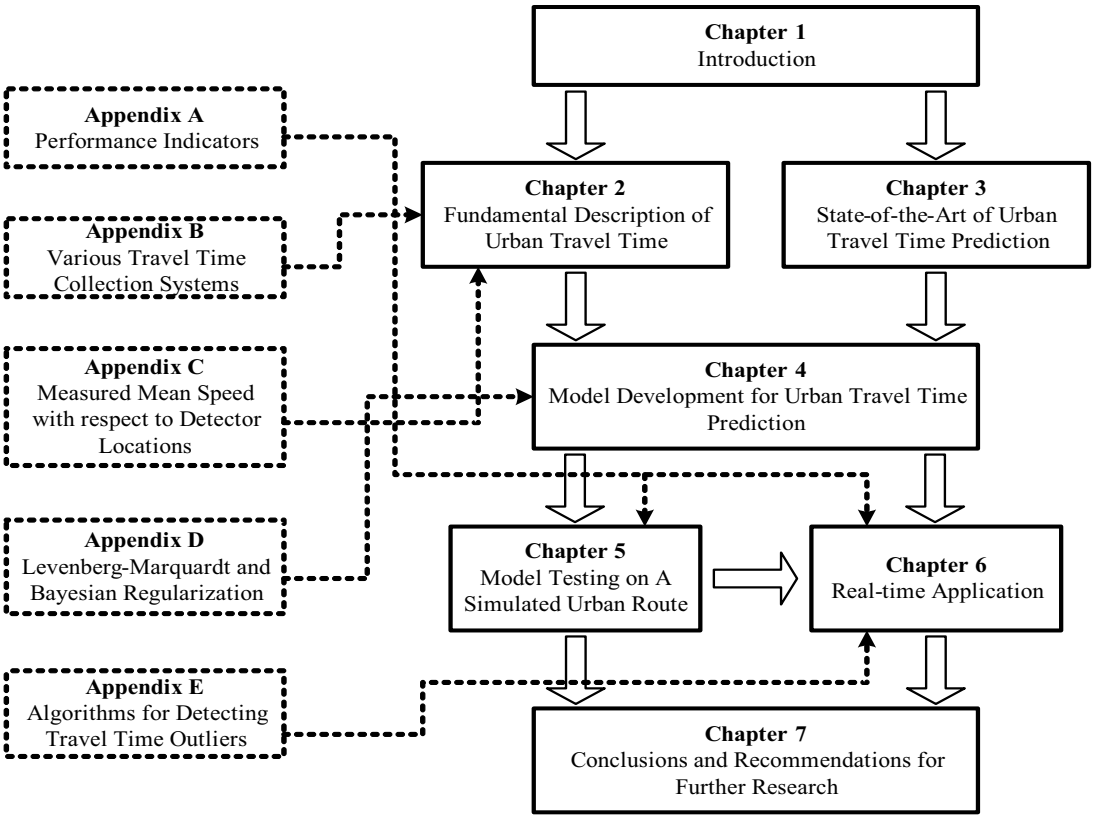


Figure 1.1: Schematic overview of the structure of the dissertation and appendixes.

In Chapter 4, we present a neural network model for urban travel time prediction. The detailed description of the batch training algorithm for the proposed model, Levenberg-Marquardt and Bayesian Regularization, is described in Appendix D.

In Chapter 6, we put the proposed model into practice. In reality, the measured travel times often consist of corrupted values, called outliers. Appendix E presents several algorithms for detecting travel time outliers.

Chapter 2

Fundamental Description of Urban Travel Time

2.1 Introduction

In the literature, two distinct road types for the application of travel time prediction can be easily identified: freeways and urban (signalized or unsignalized) streets. This is due to the fact that the characteristics of freeways are significantly different from those of urban streets. The focus here is on travel time prediction solely for urban streets. Before we explore this problem, we refer the definitions of freeways and urban streets as those in (Transportation Research Board 2000). *A freeway* is defined as a divided highway with full control of access and two or more lanes for the exclusive use of traffic in each direction. The term "*urban street*" refers to urban arterials and connectors, including those in downtown areas. There are no signalized or stop-controlled at-grade intersections along freeways, and access to and from the freeway is limited to ramp locations. The principal difference between freeway traffic and urban traffic is that the former is uninterrupted while urban traffic is interrupted at intersections. Depending on the control scheme and the degree of saturation, the interruption causes stochastic delays. These delays constitute a large part of travel times on urban streets (Viti & Zuylen 2004).

Apart from the principal difference, other important factors also influence the travel times for urban streets: (1) *Speed Condition*: Urban streets usually operate at lower speeds (less than 50km/h) compared to freeways (more than 80km/h). (2) *Bus Blockage*: The impact of local transit buses that stop to discharge or pick up passengers at near-side or far-side bus stops may result in increased delays. (3) *Pedestrian and Cyclist Disturbance:* Compared with freeways, vehicles running on urban streets can be disturbed by pedestrians and cyclists crossing streets. (4) *Transit Priority:* Transit priority signal timing gives green time with high priority to public transit, while it sacrifices time for personal cars. (5) *Parking:* The influences of parking are not only the reduction of the capacity of road facilities but also the disturbances of the traffic. The disturbances include the 'frictional effect' of a parking lane on the flow in an adjacent lane and the 'occasional blocking' of an adjacent lane by vehicles moving into or out of parking spaces. In this dissertation, we will not further investigate those factors.

In the next section, some basic definitions of travel times on urban streets are presented. As the dominant parts of travel times, delays that drivers experience on urban streets have been investigated in detail. The analysis of delays will be helpful in distinguishing urban travel times from freeway travel times. In the end, this Chapter introduces the basic relationship between travel times and traffic variables (volumes and speeds). Particularly, measured time mean speeds at different locations along an urban segment are analyzed. The results show that stationarity and homogeneity may be satisfied on freeways but do not hold on urban streets.

2.2 The Basic Definitions of Travel Time on Urban Streets

2.2.1 Definitions of the Elements of Urban Networks

In this dissertation, the following definitions are used.

Definition 1 *An intersection is a road junction where two or more roads either meet or cross at the same grade or level.*

Definition 2 *An urban link is a section of a urban street between two consecutive intersections.*

Definition 3 *An urban segment is a combination of one urban link and one intersection. Two types of urban segments can be defined according to the layout of the segment: type A, an intersection is connected to the start of a urban link; type B, the end of an urban link is followed by an urban intersection.*

Definition 4 *An urban street/route consists of a number of contiguous urban segments.*

Figure 2.1 schematically outlines the different elements of a typical urban route. Unless specifically stated otherwise, an intersection is denoted with index n, an urban link is denoted with index l, an urban segment is denoted with index s, and an urban route is denoted with index r.

2.2.2 Definitions of Travel Times

The following section presents some basic definitions of travel times used in this dissertation (based on Van Lint 2004 and Bovy & Thijs 2000).

Travel time is the duration of time that a driver takes from the start of a trip to the end of a trip. The spatial scope of this trip can be an intersection, or an urban link, or a segment or a route. Travel time can be expressed as

$$TT = t' - t \tag{2.1}$$

where t denotes the time instant when a driver departs from the start of the trip, t' denotes the time instant when the driver arrives at the end of the trip.

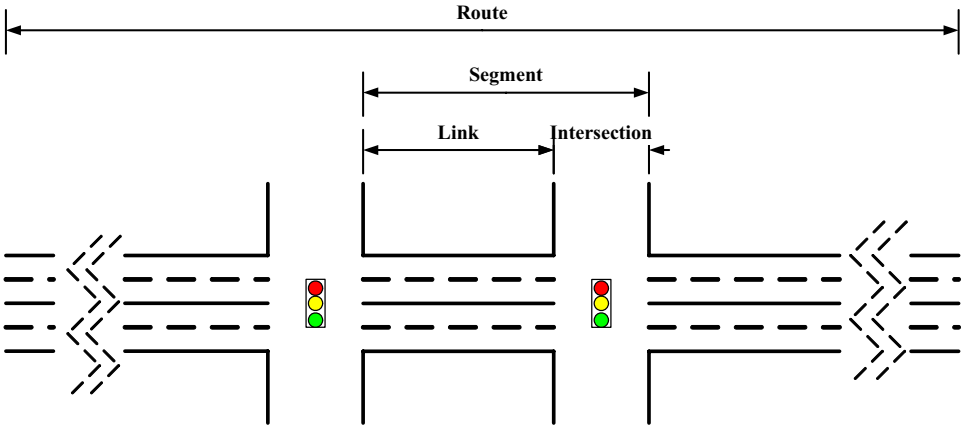


Figure 2.1: Schematic representation of a typical urban route and basic elements

Individual and Aggregated Travel Time

Definition 5 *The individual travel time $TT_i(t)$ is the time that a single vehicle/driver i traverses the route of interest at departure time instant t.*

Definition 6 *The aggregated travel time $TT(p)$ is defined as the arithmetic mean travel time of the population of all vehicles that traverse the route of interest during the departure time period $p = [t, t + \tau]$. The mean travel time of the aggregated interval can be expressed as the average of individual travel times:*

$$TT(p) = \frac{1}{N}\sum_{i=1}^{N} TT_i(t^*),\ t^* \in p$$

where N denotes the total number of vehicles departing during time period p.

Departure and Arrival Travel Time

Definition 7 *Individual departure travel time, $TT_i^d(t)$, is the travel time with departure time instant, t. Aggregated departure travel time, $TT^d(p)$, is the average travel time within departure time period p.*

Definition 8 *Individual arrival travel time, $TT_i^a(t')$, is the travel time with arrival time instant, t'. Aggregated arrival travel time, $TT^a(p)$, is the average travel time within arrival time period p.*

Note that each vehicle has both departure and arrival travel times when it traverses a finite length of a route. For this vehicle i, $TT_i^d(t) = TT_i^a(t') = t' - t > 0$. Figure 2.2 shows two vehicles' trajectories graphically, and the departure and arrival travel times for traveling from x_0 to x_1.

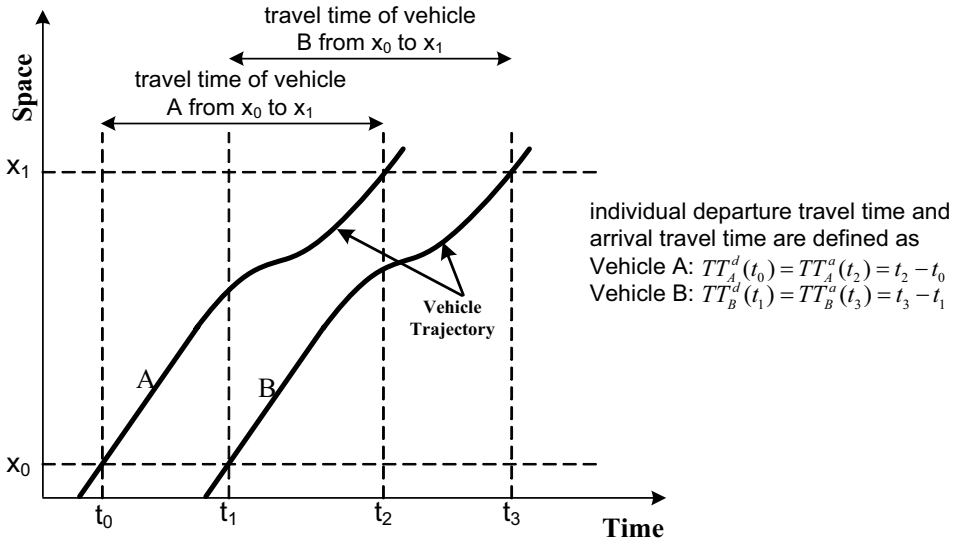


Figure 2.2: The individual departure and arrival travel time are obtained from vehicle trajectories.

Travel Time Observations

The use of cameras to capture vehicle license plate numbers is widely accepted for measuring travel times (other methods of measuring travel times seen in Appendix B). With license plate cameras, travel time observations are only available when the vehicles complete their trips. Therefore, each individual travel time observation cannot be calculated unless both of the departure time instant and arrival time instant are available. Normally, travel time collection systems provide only arrival travel times. But, departure travel times are more convenient for travel time prediction. Therefore, there is a need to convert arrival travel times to departure travel times. The simple way to obtain individual departure travel times is by shifting arrival travel times backward with the absolute value of travel times on the time axis. Figure 2.3(a) shows an example of the empirical individual travel time observations collected from an urban street, Kruithuisweg, in Delft, in the Netherlands.

It is interesting that the aggregated departure travel time observations are not the results of simply shifting the aggregated arrival travel time observations (seen in Figure 2.3(b)). This is due to the different speeds of vehicles and overtaking activities. For a simple example, vehicle A and B arrive at the end of a trip during the same arrival aggregated period, while vehicle A departs one departure aggregated period before vehicle B. Obviously, vehicle B runs faster than vehicle A. Thus, they have the same aggregated arrival time period, but separate contributions to two aggregated departure time periods.

Travel Time Estimation and Prediction

Travel time estimation refers to the calculation of the (mean) travel times of realized trips based on known traffic quantities (e.g. speeds, flows, and densities, etc.) (Bovy & Thijs 2000). That is, estimated travel times are derived from traffic measurements up to the present time instant/period (see Figure 2.4). Travel time estimation, by definition, is a technique used in cases where directly measured travel times are not available.

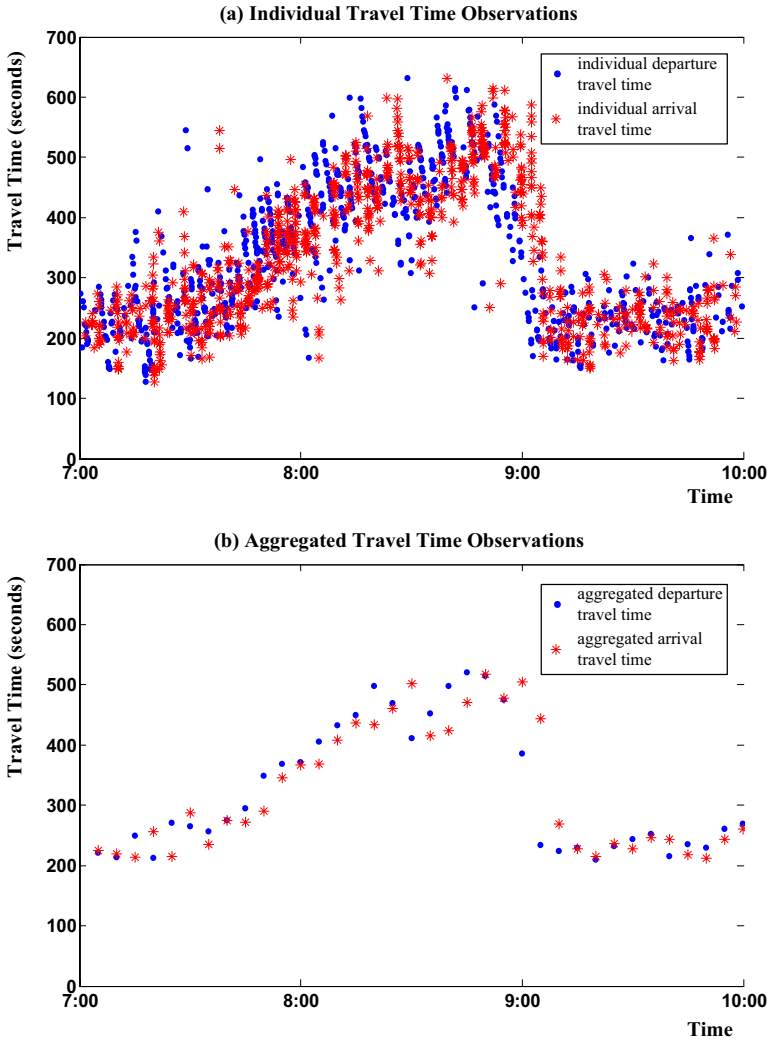


Figure 2.3: Travel time observations collected from a provincial road, Kruithuisweg, of Delft, in the Netherlands during the period of 7:00 to 10:00AM on April 22, 2006. (a) individual travel time observations (b) aggregated travel time with time interval of 5 minutes.

Travel time prediction refers to the calculation of the departure travel time for the future traffic conditions. Predicted travel times are calculated based on not only the past traffic conditions but also the future traffic conditions. The past traffic conditions are known and can be measured. However, the future traffic conditions are unknown. There is no doubt that travel time predictions are based on accurately predicting future traffic conditions.

Note that both estimated travel times and measured travel times can be expressed at the departure or arrival time instant, depending on different applications. But, predicted travel times always use the departure time instant unless explicitly stated. Predicted travel times are more useful for travelers because they provide information affecting travelers' decisions on, for example, modes, routes, and departure times. Measured travel times only contain past information when drivers have completed the trip of interest, which has less or even no value for future traffic conditions.

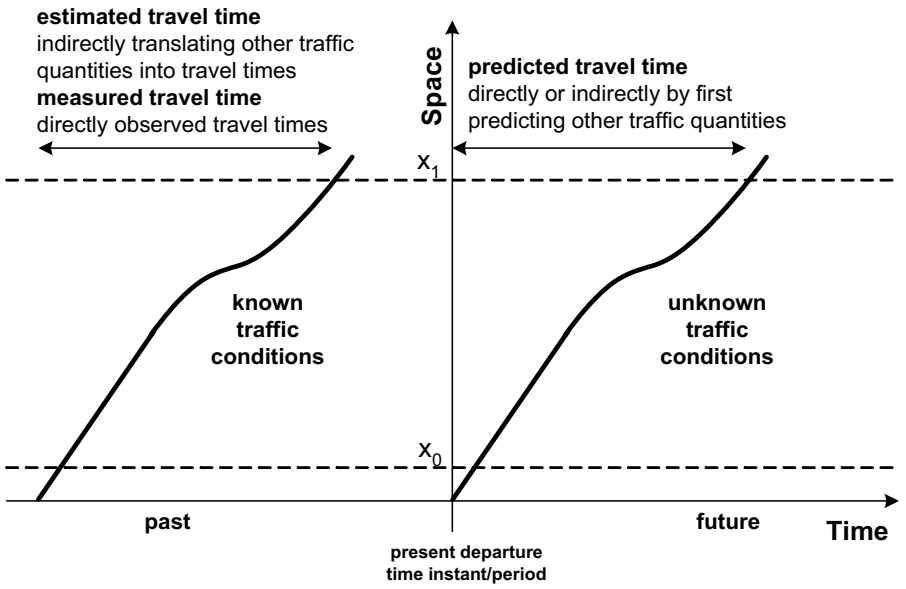


Figure 2.4: The difference between travel time observation, estimation and prediction (cite from Van Lint 2004).

Instantaneous Travel Time

The specific term of *instantaneous,* used for travel time, should be emphasized. Instantaneous travel time pertains to the sum of travel time estimations of each segment at the same time step. If it is assumed that the current traffic conditions remain stationary for an infinite period, the instantaneous travel time can be used as the predicted travel time. This is a simple way to 'predict' future traffic conditions, which are widely used in practice. Because of its simplicity and ease-of-implementation, this trick can be integrated into any model. However, this simplicity will result in an underestimation of future congestion-onset conditions and an overestimation of future congestion-dissolve conditions (Van Lint 2004).

2.3 Individual Travel Time and Trajectory Speed

In the trajectories the normal representation is that the position x of a vehicle is given as a function of the time t. The speeds of the vehicle is given as $v(t) = dx/dt$ and the distance travelled is given by $x = \int v(t)dt$. In order to obtain the travel time between two positions along a road, x_0 and x_5, the inverse is needed: the relation between the time as determined by the distance. As visible from Figure 2.5, this relation is not a continuous function, since at locations where a vehicle stops, the relation is multi-valued. That means that only for $v > 0$ the derivative of the time with respect to space exists. Otherwise, at the points where the vehicle stops, the derivative does not exist in a strict sense.

The time to travel from x_0 to x_5 can be given in the following expression:

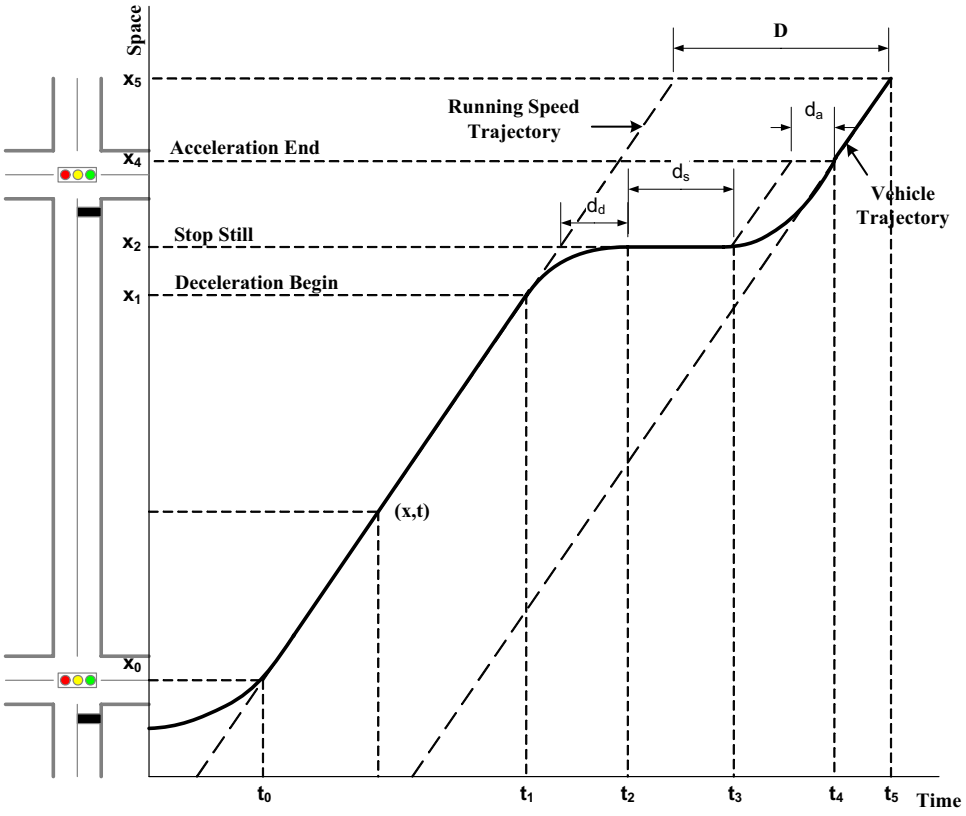


Figure 2.5: Schematic space-time diagram illustrating delay terms at a signalized intersection.

$$TT = \int_{x_0}^{x_5} \Psi(x)\, dx \tag{2.2}$$

where

$$\begin{aligned} \Psi(x) &= dt/dx = \tfrac{1}{dx/dt} \quad \text{for } dx/dt > 0 \\ \Psi(x) &= t_s \delta(x - x_s) \quad \text{for } dx/dt = 0 \end{aligned}$$

where the delta function is defined as

$$\begin{aligned} \delta(x) &= 0 \quad \text{for } x \neq 0 \\ \delta(x) &= \infty \quad \text{for } x = 0 \end{aligned}$$

and $\int \delta(x)\, dx = 1$.

Using this approach we can rewrite the travel time from x_0 to x_5 as

$$TT = \int_{x_0}^{x_5} \left[\frac{1}{dx/dt} + (t_3 - t_2)\delta(x - x_3) \right] dx \tag{2.3}$$

2.4 Components of Travel Times on Urban Streets

As stated earlier, delays at intersections play a dominant role in the travel times that drivers experience on urban streets. Thus, travel time TT on an urban street r can be subdivided into two main components as follows:

$$TT_r(t) = T_f(t) + D(t) \tag{2.4a}$$

where $T_f(t)$ denotes the travel time in free-flow conditions at time instant t, and $D(t)$ denotes the (mean) delay that drivers experience when departing at time instant t. Free-flow conditions refer to no other vehicles (volume is zero) and no intersections (no signal control). In free-flow conditions, drivers are able to choose their desired speeds. It is certain that the desired speeds vary with driving behavior, speed limit, weather conditions, etc. The investigation of the variability of desired speeds is out of the scope of this dissertation. Let v_f be constant. T_f can be expressed as a function of the route length L_r and the desired speed v_f in free flow conditions, that is $T_f = \frac{L_r}{v_f}$. This results in T_f being constant for 'free flow' conditions, while D varies with the effects of different influencing factors.

In the literature, several terms for delay have been used widely. In the interest of uniformity the following definitions are given based on (Hoeschen et al. 2005 and Skabardonis & Geroliminis 2005). The delays may (not exclusively) be from different sources: (1) full stop, acceleration and deceleration; (2) the position of vehicles in a platoon; (3) signal control strategies for isolated intersections; (4) signal (offset) coordinations between two intersections; (5) overflows due to high traffic demand. This implies that the influencing factors under consideration in this dissertation are traffic signals and traffic volumes, but do not include others, like weather, transit priority, pedestrian disturbance, etc.

2.4.1 Delays of stop, acceleration and deceleration

To illustrate the delays caused by a full stop, acceleration and deceleration, a hypothetical trajectory of a single vehicle is shown in Figure 2.5. As mentioned previously, the urban traffic is interrupted due to the presence of traffic signals. In most cases, vehicles decelerate when they approach a stop line and the traffic signal turns red. Then they stand still until the traffic signal turns green. Afterwards, vehicles accelerate to pass the intersection. These delay components are illustrated graphically in Figure 2.5 and are summarized below.

Definition 9 *The stop delay is the duration that a vehicle physically stops and waits for the signal to turn green at the signalized intersection. This is shown as d_s in Figure 2.5 and corresponds to the flat part (t_2 to t_3) of the space-time trajectory.*

$$d_s = t_3 - t_2 \tag{2.5}$$

The stop delay is strongly related to the time instant when the vehicle arrives at the traffic signal. Obviously, vehicles have to decelerate and stop in the red phase, while they continue to pass the intersection in the green phase if no queue is present.

Definition 10 *The deceleration time is the duration that a vehicle goes from running speed to standing still in the end of the queue at the signalized intersection. That is, the vehicle runs from x_1 to x_2, and deceleration time corresponds to the time period (t_1 to t_2) of the space-time trajectory. Decelerated delay, d_d, is the time difference between deceleration time and time running at free-flow speed from x_1 to x_2:*

$$d_d = (t_2 - t_1) - \frac{x_2 - x_1}{v_f} \tag{2.6}$$

If we consider that the vehicle decelerates at a constant deceleration rate, γ_d, then the deceleration time can be calculated as

$$t_2 - t_1 = \frac{1}{v_f}\left[(x_2 - x_1) - \frac{v_f^2}{2\gamma_d}\right] \tag{2.7}$$

Substituting equation 2.7 in 2.6 gives

$$d_d = \frac{v_f}{2\gamma_d} \tag{2.8}$$

Definition 11 *The acceleration time is the duration that a vehicle takes to speed up from zero to running speed. This corresponds to the time period (t_3 to t_4). Accelerated delay, d_a, is the time difference between the acceleration time and time running at free-flow speed from t_3 to t_4.*

Analogously to equation 2.6, the accelerated delay can be expressed

$$d_a = \frac{v_f}{2\gamma_a} \tag{2.9}$$

in which γ_a is the acceleration rate.

Thus, the travel time for a single vehicle, assuming no interaction with other vehicles, can be calculated by

$$\begin{aligned} TT_r(t) &= T_f(t) + D(t) && (2.10)\\ &= \frac{L_r}{v_f} + d_d + d_s + d_a && (2.11)\\ &= \frac{L_r}{v_f} + (t_3 - t_2) + \frac{v_f}{2\gamma_d} + \frac{v_f}{2\gamma_a} && (2.12) \end{aligned}$$

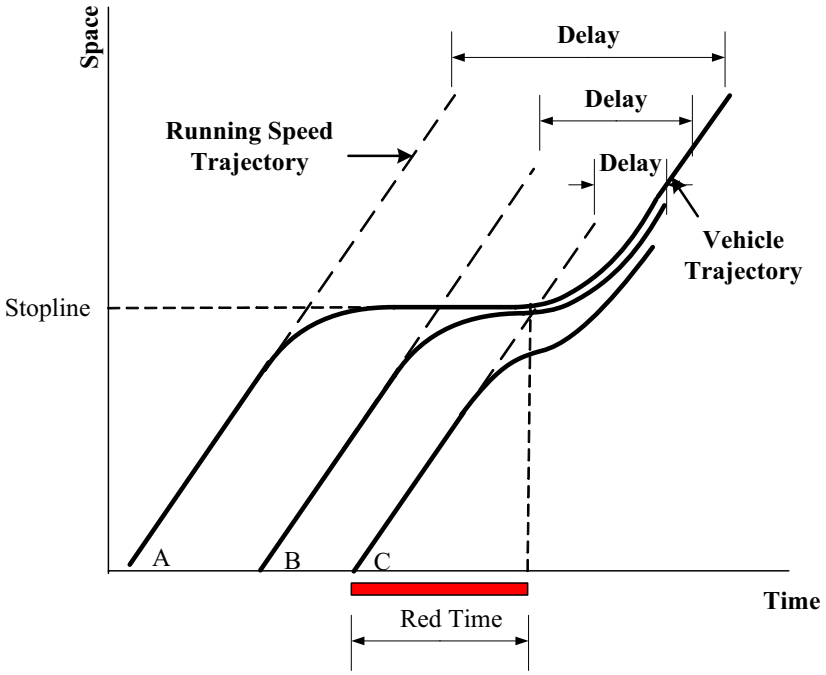


Figure 2.6: Trajectories of vehicles approaching a traffic signal. Vehicle A arrives at stopline when the traffic signal turns red, which cause him wait for an entire red time. Vehicle B arrives at stopline at the middel of red phase. Vehicle C do not need to stop still when he passes the intersection.

Example 1 *Suppose there is a free-flow condition and d_d is highly dependent on the time when the vehicle arrives at the signal (see in Figure 2.6). For example, vehicle A decelerates at deceleration rate γ_d and reaches the stop line exactly at the start of the red phase (this is the worse case that a vehicle might experience for the longest delay); r_e denotes the red time; vehicle B reaches the stop line when the signal turns green (this is the last vehicle to stop at the signal); vehicle C decelerates but the signal turns green before it stops so it accelerates from a non-zero speed v'. Note that the situation discussed here is only for undersaturated intersections. For the three different vehicles, the travel times are calculated respectively by*

$$\begin{aligned}
\text{Vehicle A:} \quad & TT_r^a = \frac{L_r}{v_f} + r_e + \frac{v_f}{2\gamma_d} + \frac{v_f}{2\gamma_a} \\
\text{Vehicle B:} \quad & TT_r^b = \frac{L_r}{v_f} + 0 + \frac{v_f}{2\gamma_d} + \frac{v_f}{2\gamma_a} \\
\text{Vehicle C:} \quad & TT_r^c = \frac{L_r}{v_f} + 0 + \frac{\left(v_f - v'\right)^2}{2\gamma_d v_f} + \frac{\left(v_f - v'\right)^2}{2\gamma_a v_f}
\end{aligned} \tag{2.13}$$

For a simple example, given $L_r = 200m$, $r_e = 36s$, $\gamma_d = 2m/s^2$, $\gamma_a = 1.5m/s^2$, $v' = 20km/h$ and $v_f = 60km/h$, the percentage differences in delays are shown in Figure 2.7. Note that in reality drivers do not slow down/speed up at constant decel-

eration/acceleration rates. Readers who are interested in modeling a dynamic deceleration/acceleration rate can refer to (Akcelik & Besley 2001).

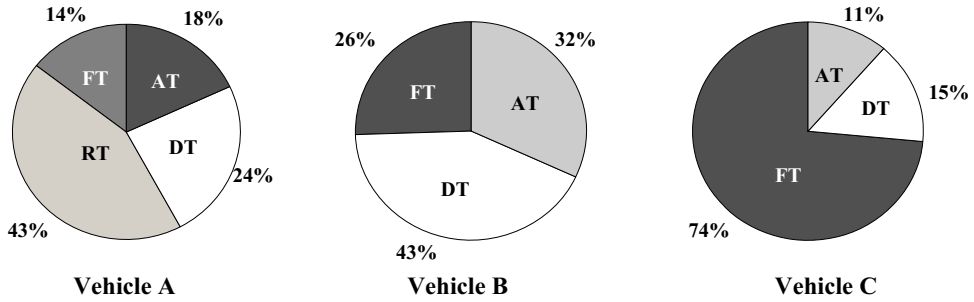


Figure 2.7: The percentages of components of travel time ($L_r = 200m$, $r_e = 36s$, $\gamma_d = 2m/s^2$, $\gamma_a = 1.5m/s^2$, $\upsilon' = 20km/h$ and $\upsilon_f = 60km/h$)

2.4.2 Delays of vehicles in platoon

During congested periods, additional delays can be caused when the arrival rate exceeds the service rate at the traffic signal. Figure 2.8 shows vehicle trajectories in a congested situation. In the first cycle, a queue has been set up. When the green time of the second cycle starts, vehicle A, B, and C start to pass the intersection. Vehicle D reaches in the end of this queue, and decelerates to keep a safe distance after vehicle C. When vehicle D reaches the stop line, the second red time is activated. Thus, vehicle D has to wait until next green phase. Note that vehicle D experiences two-time decelerated, accelerated, and stopped delays before it passes the intersection. As a result, vehicle D experiences a much longer travel time than vehicle C, although they only have a small departure time lag. This illustrates that the variability of delays (travel times) might be significant, depending on the arrival patterns.

2.4.3 Delays of signal control strategies

The modes of signal control strategies can be divided into three main classes (Zuylen 2002):

- *Fixed and pre-timed control*, where the structure and timing of the traffic control process are determined in advance and the whole control process is steered by a program;
- *Vehicle actuated control*, where individual vehicles are detected and the information from detectors is used to influence the structure and timing of the control program;

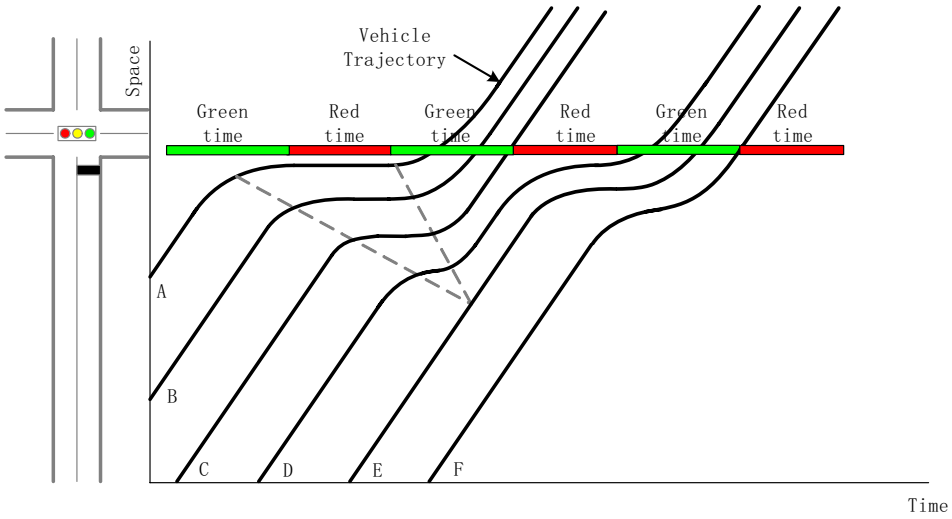


Figure 2.8: Schematic representation of delays because of overflow queues

- *Traffic dependent control*, where information about the whole traffic situation is used to make decisions about the progress of the control program.

Fixed time control is only suited for the traffic situations for which it has been designed. If the actual traffic situations are different from the designed ones, delays will increase even more than those of the conditions for which it was designed. Moreover, even if the actual volumes are equal to the designed average volumes, delays might be larger due to the random variations of arrivals. Thus, vehicle actuated and traffic dependent controls can reduce these delays due to the demand fluctuations, thereby adapting traffic control to the actual traffic situation.

Vehicle actuated control operates signals according to the detected arrivals of the vehicles at the intersection. The principal difference between the vehicle actuated control and the traffic dependent control is that the later attempts to optimize the traffic flow under the consideration of the total delay, number of stops, queue length, etc.

For isolated and smaller intersections the fixed time control normally gives higher delays than the vehicle actuated control, while traffic dependent control gives slightly better results than the vehicle actuated control. However, if the traffic volumes are larger than the capacity of the intersection, the vehicle actuated control behaves like a fixed time control (because the full green phases which run until their maximum times are realized).

In a network of closely space controlled intersections, the coordination between intersections has a large influence on the performance. The vehicle actuated control with coordination appears to give an inferior performance compared to the fixed time programs with pre-calculated green waves. There are traffic dependent control programs for network control that give a better performance than the fixed time or traffic actuated programs. Examples are the programs, e.g. SCOOT (Hunt et al. 1981), SCATS (Wilson et al. 2006), UTOPIA (www.peektraffic.nl), etc.

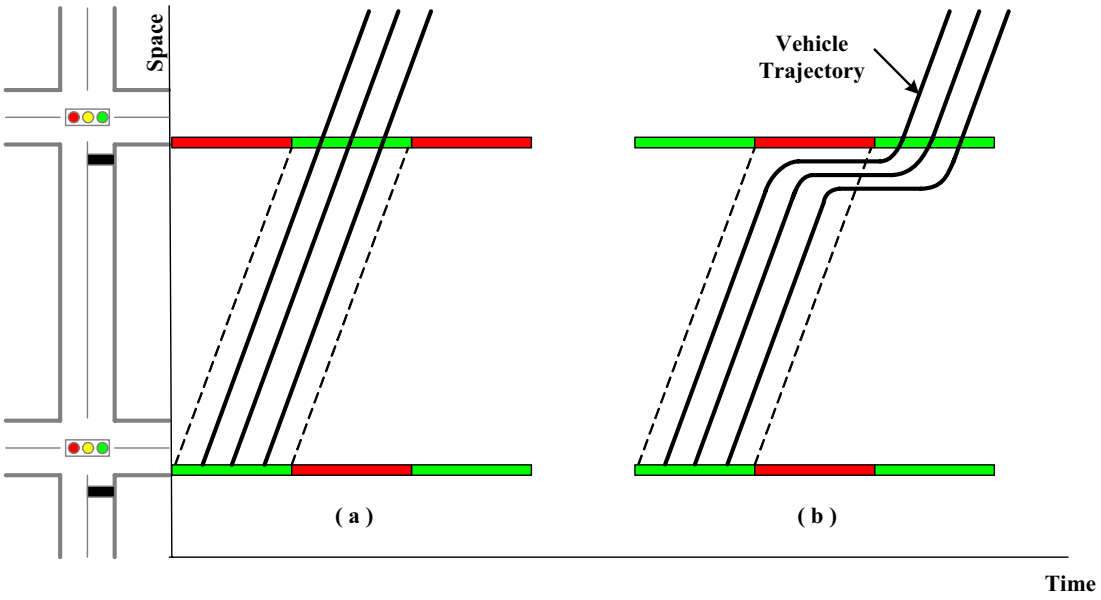


Figure 2.9: Delays caused by different offset settings: (a) green wave (b) bad offset

2.4.4 Delays caused by signal offsets

So far we have discussed the delays for vehicles approaching an isolated intersection. Along an urban street with two (or more) intersections, delays may also be influenced by offsets between adjacent signals. The offsets determine the arrival pattern of vehicle platoons at successive intersections (Geroliminis & Skabardonis 2005) and greatly affect the vehicle delays (Gartner & Wagner 2004, Skabardonis & Geroliminis 2005). Under favorable progression most of the vehicles travel without stops and delays between successive intersections. This is called a “green wave” (shown in Figure 2.9(a)). On the other hand, “bad” offsets cause high delays and may result in spillover, especially for short signal spacing (shown in Figure 2.9(b)). Gartner & Wagner (2004) used a Cellular Automata model to investigate the characteristics of traffic flow on signalized urban streets. They found that bad offset settings cause significant delays and reduce the throughput and capacity.

As mentioned above, the offset setting also influences the arrival pattern of traffic platoons at downstream intersections. The arrival pattern of traffic platoons determines the delays that those vehicles experience. The size of the platoon diminishes with the distance between signals due to the variability of vehicle behavior (Viti 2005). This phenomenon is usually referred to as platoon diffusion or dispersion. Hillier & Rothery (1967) showed the diffusion phenomenon using field data and the distance-dependency of this phenomenon. Geroliminis & Skabardonis (2005) propose an analytical methodology to predict the platoon arrival profiles and queue length along arterials with signalized intersections, based on a two-step Markov Decision Process (MDP) and the kinematic wave theory. Based on the experiments at two urban arterials, they concluded that the proposed approach can predict the arrival profiles of many signals downstream from a known starting flow.

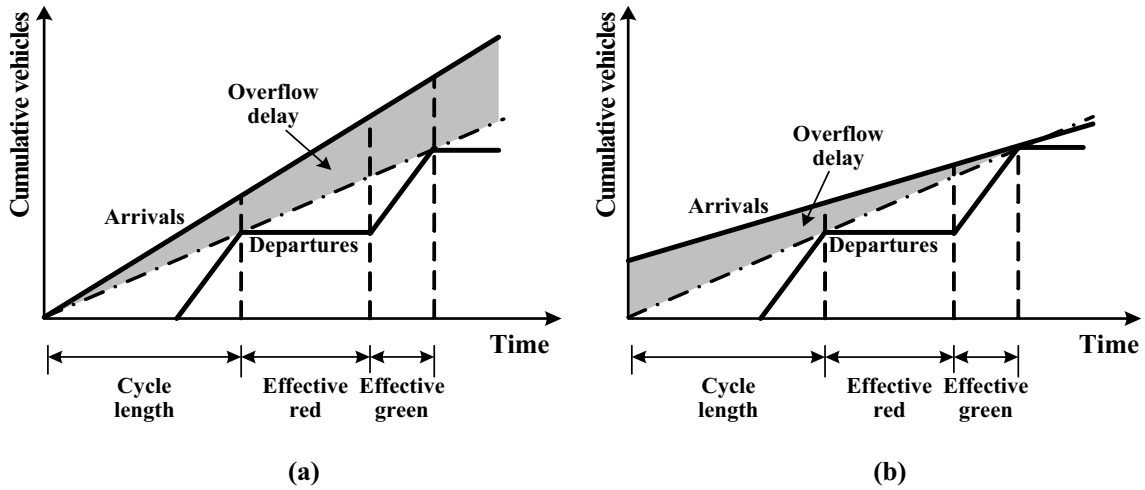


Figure 2.10: Evolution of arrivals and departures with overflow delay. (a) cumulative arrivals and departures in an oversaturated condition (b) cumulative arrivals and departures in an undersaturated condition with a non-zero initial queue (cite from Viti 2006)

2.4.5 Delays of overflow queues

The residual queue in the end of a green phase is usually called the *overflow queue*, and the corresponding delay is called the *overflow delay*. If the traffic demand is higher than the capacity of the intersection (oversaturated conditions), the assigned green phase is not sufficient to serve all vehicles arriving during the cycle time. Then, a residual queue will remain. The residual queue can be formed not only in oversaturated conditions but also in undersaturated conditions, with a non-zero initial queue caused by previous oversaturated flows (Viti 2006) or by stochastic variation in the arrival pattern.

Figure 2.10(a) illustrates the oversaturated queue with the assumption of zero initial queue and an average flow larger than the capacity, while Figure 2.10(b) shows an undersaturated case with a non-zero initial queue. It is clear that in the former case the overflow delay increases with each of the cycles. In the later case, the overflow delay decreases until that time when the two lines representing the cumulative arrivals and departure intersect each other.

2.5 The Variability of Travel Time

A common experience that everyone has is that the travel times along the same route may be very different for different days, even if the drivers depart at the same time of each day. This can be identified from the distribution of travel times. There tends to be a minimum travel time, but it is possible to have a very long travel time. In a real life, travel times vary with a number of factors e.g. fluctuations in traffic demand, vehicle composition, adverse weather, probabilistic distributions of traffic arrivals, signal timing and driver behavior, etc. The stochastic nature of these factors results in the variation of travel times (Van Lint 2004, Viti 2006). Those influencing factors interact with each other and result in the

variability of travel times. However, a thorough analysis of all influencing factors seems impossible so far.

Definition 12 *The variability of travel time refers to the degree of variation of travel times under certain conditions.*

To indicate the degree of the variation of travel times a single indicator, a so called statistical range method, is widely used as a measure for travel time variability (Tu et al. 2008). The indicator can be expressed as the difference between two percentile travel times.

$$TTV = TT90th - TT10th \tag{2.14}$$

where TTV denotes travel time variability, $TT90th$ and $TT10th$ denote 90th and 10th percentile travel times, respectively.

Due to the difficulties of collecting empirical data of influencing factors, practical investigations of travel time variability (for urban streets and freeways) are very limited (Robinson 2005). Only few studies have been conducted in terms of individual factors, such as traffic flows (Tu et al. 2007b), adverse weather (Tu et al. 2007a), incident condition (Li 2004). Apart from disaggregating travel time variability in terms of different influencing factors, time window has also been widely accepted to disaggregate travel time variability. In particular, four time windows (vehicle-to-vehicle, period-to-period, day-to-day, season-to-season) outlined are often explicitly considered (Robinson 2005).

This dissertation categorizes the variability of travel time into two groups as follows.

(1) *Vehicle-to-Vehicle travel time variability*: the variability of individual travel times within the same departure period. This type of variability is caused by driver differences, the stochasticity of traffic signals along the route, etc. Different types of vehicles (e.g. trucks or passenger cars) have specific vehicle performances, for example, acceleration/deceleration characteristics and maximum speeds. Each driver has specific driving behavior associated with his human factors (physical and mental conditions). The variability between vehicles and drivers increases the probability of breakdowns and affects the stability of the traffic conditions (Tampere 2004). The resulting variability of travel times can be identified from the individual travel time measurements. That is, two individual vehicles might experience different travel times even though they depart within a short time of each other. Figure 2.11, for example, shows individual travel time observations within a 5-minute departure period. The difference between maximal and minimal travel times equals 110 seconds.

(2) *Same-condition travel time variability*: the variability of individual travel times over more departure time periods under similar conditions. The similar conditions could be the same weather (rain, snow, foggy, etc.), time of day, day of week, inflow range, etc. Figure 2.12 demonstrates the travel time uncertainty as a function of inflows. Travel time variability before breakdown (TTV^f in Figure 2.12) does not change with the increasing inflows, while travel time variability after breakdown (TTV^j

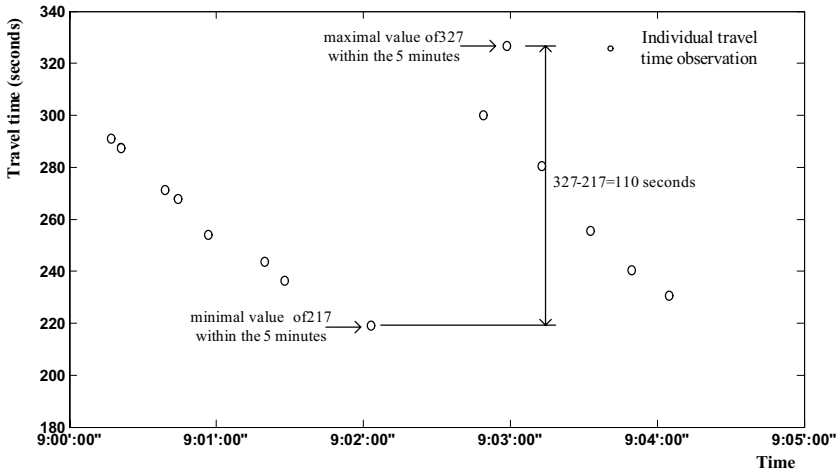


Figure 2.11: Individual travel time observations within a small departure period of 5 minutes (from 9:00AM to 9:05AM). The observations are collected from an urban arterial, Kruithuisweg, the Netherlands on Januray 20, 2004.

in Figure 2.12) increases with the increasing inflows. Figure 2.13 demonstrates the travel time variability as a function of inflow levels under both normal weather and rain conditions for six freeway corridors in the Netherlands. Figure 2.14, for example, shows aggregated travel time observations collected on 7 days in terms of the time of day. Clearly, different days have distinct shapes of travel times in the rush hour period (8:00AM to 10:00AM). The variability of travel time in the rush hour period is larger than those in the rest time of day.

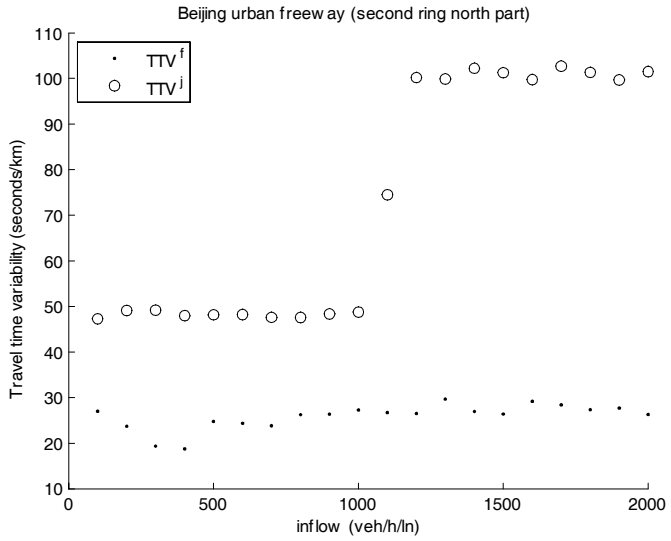


Figure 2.12: Travel time variability before and after breakdown as a function of inflow levels on Beijing second ring urban freeway (Tu et al. 2007b).

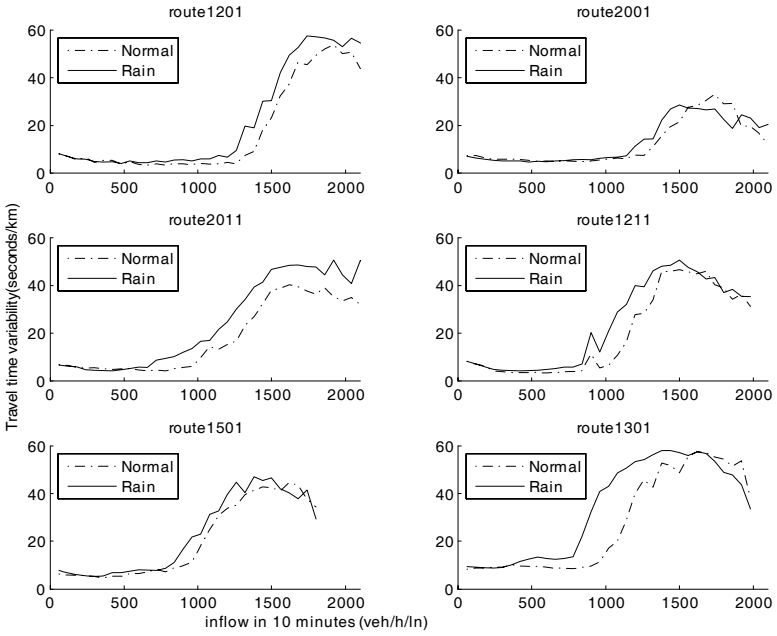


Figure 2.13: Travel time variability as a function of inflow level under both normal weather and rainy weather conditions. (Tu et al. 2007a)

2.6 Basic Relationships between Urban Travel Times and other Traffic Variables

In this section, some basic mathematics of *urban segment travel time* and traffic variables are presented. Those basic relationships can be used to estimate/predict urban segment travel times by converting traffic variables into travel times. The traffic variables are speed and volume, which are commonly collected by single/double loop detectors. Note that those basic mathematics need to be modified for *urban route travel time prediction*, taking 'platoon' and 'filtering' effects between upstream and downstream intersections into account. For more details about 'platoon' and 'filtering' effect we refer to (Rouphail et al. 2000). First, a discussion of how to derive the individual travel times from the volumes is presented. Then, the relationship between mean travel times and time/space mean speeds is described.

2.6.1 Individual travel time and volume

By means of a so-called cumulative curves approach (Daganzo 1997), travel times can be easily derived from volumes collected from inductive loop detectors. Suppose that two inductive loop detectors are installed on an urban link to collect volumes at locations A and B (see in Figure 2.15(a)). From the volumes, two cumulative curves, N_A and N_B, can be derived. If the traffic flows are conserved and satisfy a first-in-first-out (FIFO) property, then the flows that enter up to time t (i.e. $N_A(t)$) must all exit from the link by exactly time $t + TT(t)$. We can use this to compute $TT(t)$ from the cumulative curves

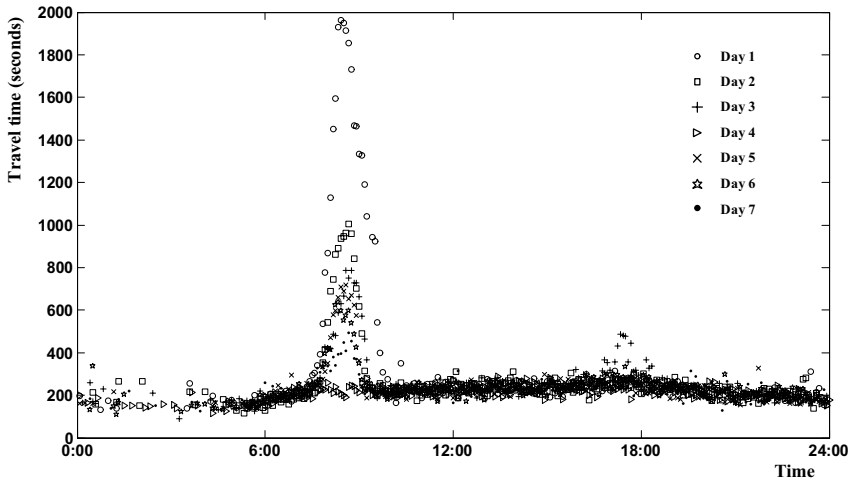


Figure 2.14: Travel time variability in terms of time-of-day. 7-day travel time measurements collected from an urban arterial, Kruithuisweg, the Netherlands in 2004.

N_A and N_B. Simply, we can find the time t^* at which the cumulative curve $N_B(t^*)$ is equal to the cumulative curve $N_A(t)$. Then t^* is the exit time and $TT(t) = t^* - t$. In other words, $TT(t)$ is the horizontal time shift between the two cumulative curves $N_A(t)$ and $N_B(t^*)$, as shown in Figure 2.15(b).

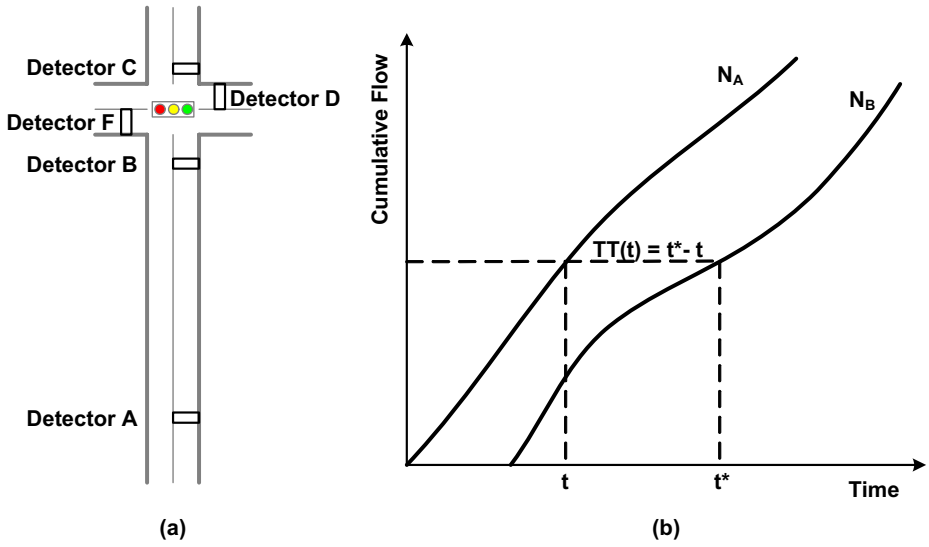


Figure 2.15: (a) the layout of inductive loop detectors installed along an urban link (b) the derivation of travel time from cumulative curves

Obviously, the travel time from locations A to B only covers the urban link. Now, we extend the spatial scope to an urban segment, from locations A to C. In order to derive travel times from volumes collected at locations A and C, more extra loop detectors are required. This is because loop detectors A and C cannot identify vehicles from other streams, such as left-turning volumes from location F and right-turning volumes from location D. With those extra loop detectors the exact throughput volumes from locations

A to C will be identified. Then, the derivation of travel times from locations A to C is similar to the method for the travel time from locations A to B.

The clear advantage of the cumulative curve approach is its simplicity. However, the underlying assumptions of this approach include *vehicle consistency* and *FIFO*. These requirements restrict its applications to only a small spatial scope (urban link) and a one-lane road (without overtaking behaviors). For a multi-lane urban route, the requirement of FIFO is hard to satisfy because of overtaking activities. Also, if an urban route covers more than one intersection, additional loop detectors should be installed for each traffic stream in order to guarantee vehicle consistency. The use of a numerous amount of inductive loop detectors is not feasible from the practical perspective. Because of these reasons, a real-time application of the cumulative curve approach only has been done on freeways (Nam & Drew 1996), and no literature, to the best of the author's knowledge, reports a good example of implementation on urban routes.

2.6.2 Mean travel time and mean speed

Before we explore the relationship between mean travel times and mean speeds for urban streets, the relationship for a link of a freeway will be presented. In a stationary and homogeneous state of a freeway link, the mean travel time simply is the inverse of the space mean speed times the length of the link:

$$TT = \frac{L}{v_s}$$

Since the space mean speed is a variable which is quite difficult to derive from that directly measured with loop detectors, the time mean speed is used to derive the space mean speed according to

$$v_t = \frac{\sigma_s^2}{v_s} + v_s \tag{2.15}$$

where σ_s^2 is the variance of the space mean speed. Readers who are interested in practical implementation can refer to (Van Lint 2004).

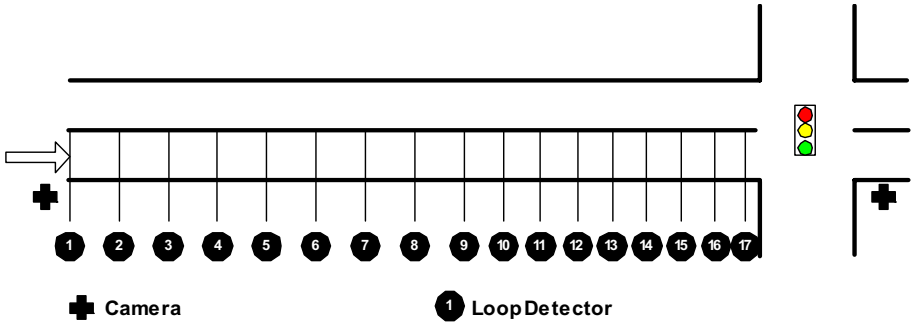


Figure 2.16: Schematic drawing of the layout of an urban segment

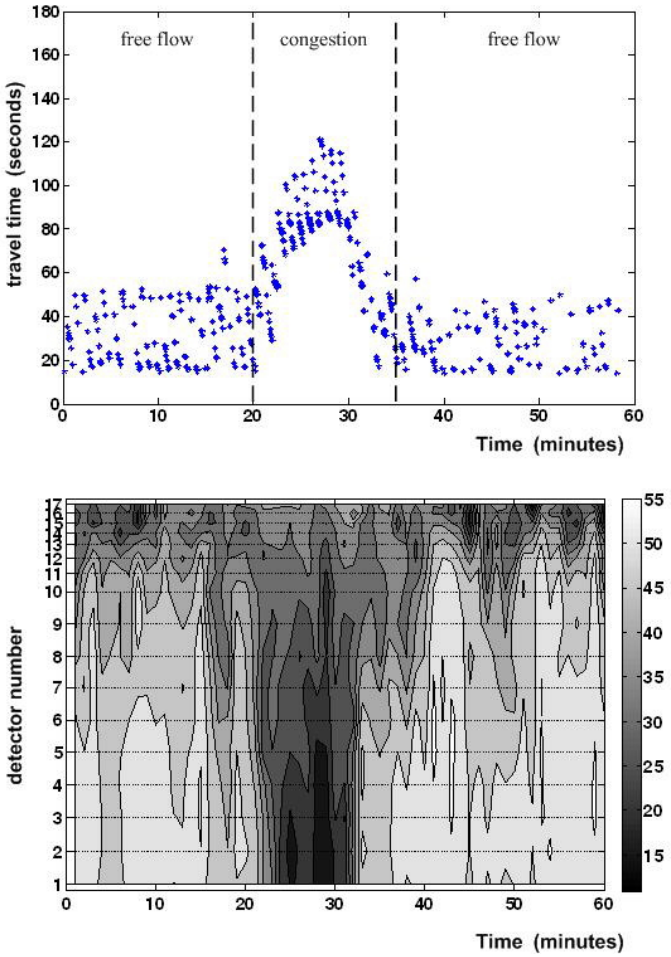


Figure 2.17: VISSIM simulation results on an urban segment. The figure shows both mean travel times (top graph) and a contour plot of mean speeds measured at inductive loop detectors (bottom graph).

The clear drawback of putting this theoretical relationship into practice is the assumption of stationarity and homogeneity. They will likely not hold on the road stretches of any practical length (a couple of hundred meters), especially not in congested and / or unstable traffic in which stop and go waves occur (Helbing 1997). For urban streets, traffic is interrupted by traffic signals so that stationarity and homogeneity are not satisfied even in free flow conditions. Also, the locations of loop detectors do have a significant influence on the measured mean speeds. Details of the analytical calculation of mean travel times and time/space mean speeds with respect to the locations can be found in Appendix C. The following will illustrate the measured speeds at different locations along an urban segment.

Due to the difficulty in finding a place where many detectors can be installed on the same link, VISSIM3.70 (PTV AG 2003) was used to simulate an urban segment. There have been 17 inductive loop detectors installed along a 250-meter urban segment. Travel time observations are collected automatically after the start and end locations are determined. Figure 2.16 gives a schematic drawing of the layout of the urban segment.

Time varying demand patterns were used, which simulate different traffic conditions (e.g.

roughly free flow and congestion). The top and bottom plots of Figure 2.17 show the mean travel times and time mean speeds, respectively. A short period of oversaturation occurs due to a large traffic demand. As a result, a congestion builds on after about 22 minutes and dissolves until 32 minutes. Under congested conditions, travel times increase up to 120 seconds, which is three times the mean free flow travel time (40 seconds).

Figure 2.18 shows the mean and standard deviation of measured time mean speeds at different locations along the urban segment. Under free flow conditions, the mean of the time mean speeds increases as the detector location is further from the traffic signal. Detector 1 (the farthest detector from the traffic signal) has the highest value for the mean of the time mean speeds, while it has the smallest standard deviation. Drivers are able to choose their desired speeds when they are far away from the traffic signal. But, they have to decelerate when they are close to the traffic signal, especially in a red phase. Depending on the signal phase at the arrival times, drivers might have zero speeds if it is red or maintain relatively high speeds as their desired speeds if it is green. This uncertainty explains why the closest detector has the highest standard deviation of time mean speeds.

Under a congested condition, the average speeds in the queue is low, and the speeds becomes large when cars pass the stop line. Consequently, the pattern of the mean and standard deviation of the time mean speeds is opposite to the pattern under the free flow condition.

In summary, the measured time mean speeds vary in terms of different locations, even within a small spatial scope. It is certainly inappropriate to assume stationary and homogeneous states for an urban segment under both free flow conditions and congested conditions. In other words, the time mean speed at any location cannot precisely reflect the state of the entire segment. Therefore, equation 2.15 is not applicable within the urban context.

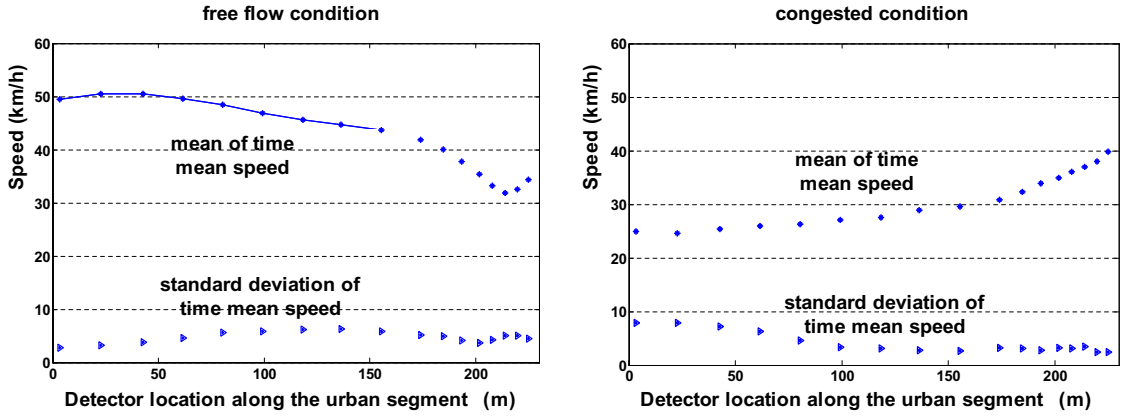


Figure 2.18: The mean and standard deviation of measured time mean speed at different locations along the urban segment under free flow condition (left graph) and congested condition (right graph).

2.7 Summary

This dissertation focuses on short term travel time prediction solely for urban networks. To distinguish this dissertation from those research studies on short term travel time prediction for freeways, this Chapter started with some definitions for the elements of urban networks. Some general definitions of travel times were given, such as individual and mean travel time, arrival and departure travel time, and travel time observation, estimation and prediction.

In this dissertation, only the traffic control strategy and traffic demand are considered as the influencing factors. Correspondingly, the traffic signal and traffic volume are used for modeling travel times. Clearly, we do not take into account other factors, like weather, transit priority, pedestrian disturbance, etc.

As stated above, delays at intersections play a dominant role in the travel times that drivers experience on urban networks. Many (not exclusively) factors which influence the travel times were given. Among those influencing factors, five types of delays were discussed: (1) the stop, deceleration and acceleration delays due to signal phases (e.g. red or green); (2) delays in terms of vehicles' position in the platoon; (3) delays caused by signal control strategies for an isolated intersection; (4) delays caused by the signal offset between two intersections; (5) delays caused by overflow queues. In accordance with those delays, a brief overview of the variability of urban travel times was given. Delays highly depend on the arrival times in the cycle (long delays for a driver arriving at the beginning of a red phase, short delays for a driver arriving in the end of a red phase. The consequence is that urban travel times have a large variability and exact predictions seem difficult.

Finally, the relationships between the urban travel times and traffic variables (volume and speed) were investigated. Estimating travel time with volumes requires that each stream of each urban link have one inductive loop detector. In a real operation, for a large network this requirement of a tremendous amount of loop detectors is hard to be met at present. This is why the cumulative curve approach is not applicable in practice. Given stationary and homogeneous traffic conditions, the time/space mean speeds and travel times can be analytically derived (see in Appendix C). A simulation shows that stationarity and homogeneity do not hold on an urban segment under both free flow conditions and congested conditions.

In the next Chapter, a literature review of existing models for urban travel time prediction will be presented.

Chapter 3

State-of-the-Art of Urban Travel Time Prediction

3.1 Introduction

In order to better present an overview of urban travel time prediction, two principles are used to limit the scope of this overview.

First, this overview only covers the prediction methods that have been used exclusively for *urban segments/streets* but not for *freeways*. Those who are interested in travel time prediction for *freeways* can refer to (Hinsbergen et al. 2007, You & Kim 2006, Van Lint 2004, Vanajakshi 2004, Liu 2004 and Paterson 2000).

Secondly, this overview only focuses on *short term* travel time prediction. Usually, the term of short term here refers to predicting the travel time of vehicles departing in the next 60 minutes, while the term of long term refers to predicting travel time of vehicles departing tomorrow, next week, month or year (Van Lint 2004). The main difference between the models for short term and long term is that the long term approaches usually use historical average values by classifying days into day types with similar profiles. For the topic of long term travel time prediction, we refer to (Vanajakshi 2004).

To categorize different travel time prediction models, two common classes are often used: *model based* and *data driven*. Model based approaches explain traffic processes based on a physical mechanism, while data driven approaches are based purely on data. In other words, the principal difference between the two classes is that the data driven approaches predict travel times without explicitly addressing the (physical) traffic processes. The parameters used in data driven models are not interpreted by any physical meaning. This implies that the data driven models are difficult to explain the physical mechanism relating travel times to other traffic variables (e.g. flow, speed and density). In practice, however, data driven models are implemented easily and show a good performance.

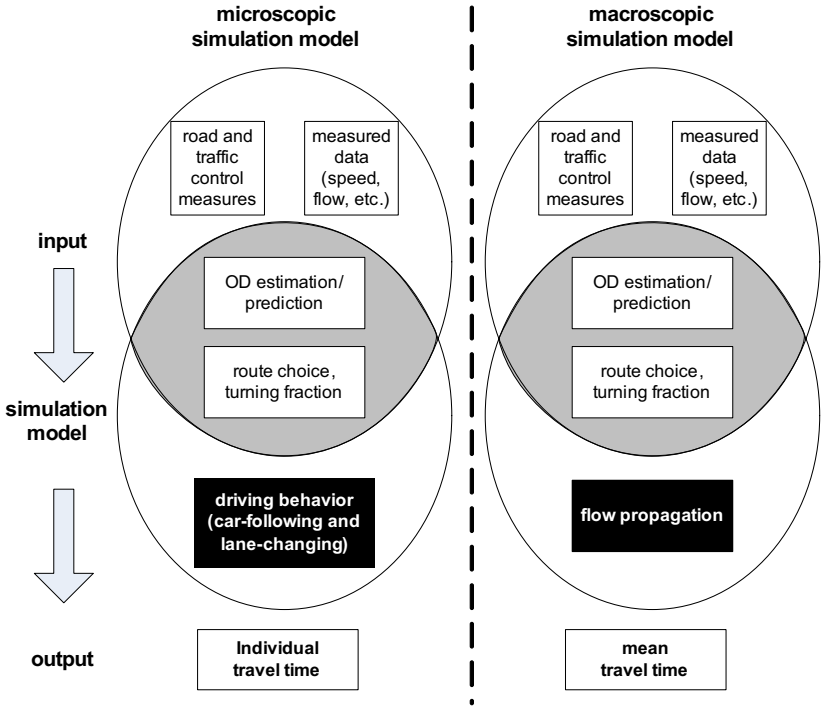


Figure 3.1: Structure of microscopic and macroscopic simulation model for travel time prediction.

3.2 Model-based Approaches

3.2.1 Simulation models

Simulation models imitate the movement of vehicles along road networks and are applicable for both freeways and urban streets. Based on the representation of traffic flow or vehicle movement, traffic simulation models can be classified into microscopic, mesoscopic and macroscopic. The main difference between microscopic and macroscopic models is : *microscopic traffic models* describe each individual vehicle by different types (e.g., passenger car, truck and bus), specific driver characteristics (e.g., aggressive or conservative), and driving behavior (e.g., lane-change maneuver, car following logic, gap-acceptance logic, and driver decision process). *Macroscopic traffic models,* on the other hand, simulate vehicle and driving behavior at an aggregated level, in which the traffic stream is represented by the aggregated traffic flow, density and speed (see in Figure 3.1). *Mesoscopic traffic models* simulate individual vehicles, but at an aggregate level, usually by speed-density relationships and queuing theory approaches.

Simulation models usually rely on an Origin-Destination (OD) matrix and a route choice model to assign vehicles onto road networks or to propagate vehicles with turning fraction. The OD matrix, route choice and turning fraction can be static (derived from historical data), or dynamic (estimated from online data). For a reference, see for example (Van der Zijpp 1996, Bierlaire & Crittin 2004).

The clear *advantage* of simulation models is the ability to explicitly interpret the physical traffic processes. They are able to isolate the factors influencing travel times and to investigate the cause and effect of travel times in different traffic conditions. Since they are

based on (physical) traffic flow theories, they can be transferred and applied to any route of interest by recalibration.

However, the major *disadvantages* of simulation models include highly computational complexity, the high degree of expertise required for design and maintenance, intensive model/parameter calibration, and the fact that they require the predictions of traffic demand and supply at the model boundaries as inputs (Van Lint 2004, Liu 2004). In particular, two online tasks, OD and turning fraction estimation, are crucial for the real-time application of simulation models. However, few successful practical applications of the two online tasks can be found, although some research efforts have shown promising concept designs in simulation environments (Bierlaire & Crittin 2004, Antoniou 2004).

3.2.2 Delay formulas

To precisely investigate the delays that vehicles experience at intersections, the pioneering work by (Webster 1958), which is expressed by equation 3.1, has been used widely probably because of its simplicity.

$$D = \frac{c_y(1-\frac{g_e}{c_y})^2}{2(1-\chi\frac{g_e}{c_y})} + \frac{\chi^2}{2q(1-\chi)} - 0.65(\frac{C_a}{q^2})^{1/3\chi^2+\frac{g_e}{c_y}} \tag{3.1}$$

where D is delay, s_a is saturated flow rate, g_e is effective green time, c_y is cycle time, q is vehicle arrival flow rate, $C_a = s_a g_e/c_y$ is the capacity of a intersection branch, $\chi = q/C_a$ is volume-to-capacity ratio.

Note that the equation 3.1 is only for average not for individual delays. Following Webster's work, other steady state stochastic models (Miller 1963, Newell 1960, McNeil 1968, Heidemann 1994) were proposed. A main consequence of these models is that the estimated delay rises to infinity as the traffic flow rate approaches saturation ($\chi = 1.0$). Therefore, these models are valid only under the condition that the average flow rate does not exceed the average capacity rate. This means that they are not able to deal with the congested traffic conditions. To overcome this weakness, a general time-dependent delay model was presented by (Kimber & Hollis 1979) and further enhanced by (Robertson 1979). The capacity guide delay models currently used in the United States, Australia and Canada are based on the general time-dependent delay model. A general form of these capacity guide delay models can be summarized as follows (Dion et al. 2004):

$$D = d_1 \times f_{PF} + d_2 + d_3 \times f_r \tag{3.2}$$

with:

$$d_1 = 0.5c_y \frac{\left(1-\frac{g_e}{c_y}\right)^2}{\left[1-\frac{c_y}{g_e}\cdot\min(\chi, 1.0)\right]} \tag{3.3}$$

$$d_2 = 900\chi^n T\left[(\chi-1)+\sqrt{(\chi-1)^2+\frac{mkI}{CaT}\cdot(\chi-\chi_o)}\right]$$

$$f_{PF} = \frac{(1-P)f_t}{1-\frac{g_e}{c_y}} \tag{3.4}$$

where d_1 is uniform delay (delay assuming uniform arrivals), d_2 is incremental delay (accounting for effect of random arrivals and oversaturation queues), d_3 is initial queue delay for over-saturation queues that may have existed before the analysis period, f_{PF} denotes adjustment factor accounting for the quality of progression in coordinated systems, f_r denotes adjustment factor for residual delay component, f_t denotes adjustment factor for situations in which the platoon arrives during the green interval, n and m denote capacity guide model parameters, P denotes proportion of vehicles arriving during effective green interval, k denotes incremental delay factor accounting for pre-timed or actuated signal controller settings, I denotes adjustment factor for upstream filtering/metering, T denotes evaluation period, χ_o denotes volume-to-capacity ratio below which the overflow delay is negligible. Table 3.1 indicates the specific values assigned to the parameters in each capacity guide model (Dion et al. 2004).

Table 3.1: Capacity guide delay model parameters (cite from Dion 2004)

Parameter	**Model**			
	Australian(1981)	Canadian(1995)	HCM(1994)	HCM(1997)
f_r	0	0	0	1
n	0	0	2	0
m	$6 or 12^a$	4	$4 or 16^c$	8
k	n/a	n/a	n/a	$0.04-0.5^e$
I	n/a	n/a	n/a	1.0^f
T	$variable$	$variable$	$15min$	$variable$
χ_o	$0.67+\frac{s_a g_e}{600}$	0	0	0
f_{PF}	1	$1 or Eq.3.4^b$	$1, 0.85, or Eq.3.4^d$	$Eq.3.4$

[a] 12 for random arrivals; 6 when platooning occurs.
[b] 1.0 for isolated intersections; equation 3.4 in other cases.
[c] Function of arrival type (16 for random arrivals, 12 for favorable or non-favorable progression, 8 for very poor or highly favorable progression, 4 for very unfavorable progression).
[d] 1.0 for pre-timed, non-coordinated signals; 0.85 for actuated, non-coordinated systems; equation 3.4 for coordinated systems.
[e] 0.5 for pre-timed signals; $0.04-0.5$ for actuated controllers.
[f] 1 for isolated intersection only.

For the real time application, several *limitations* of these delay formulas have been pointed out by (Dion et al. 2004). First, the assumption that the number of arrivals follows a known distribution, typically a Poisson distribution, does not change, but over time, is hard to maintain in a real situation for a longer period. Secondly, they all assume that the headways between vehicle departures from the stop line follow a known distribution

with a constant mean, or are identical. Thirdly, these formulas assume to run long enough to have a steady state condition. Fourthly, the purpose of equation 3.1 is to calculate the expectation value not a specific delay in one situation. Overall, the complex behavior of queues at traffic signals and the large variety of cases observed in real life limit the validity and applicability of such formulas. An extensive overview of queue and delay models at signalized intersections is given in (Viti 2006).

3.2.3 Queuing theory based models

The so-called sandglass travel time model is named for the image of sand flowing through an hourglass and is an analogy for a vehicle queue discharging at an intersection. Usami et al. (1986) were perhaps the first to propose a sandglass type of travel time model for an oversaturated link. In their formulation, the delay is calculated by

$$TT = \frac{N_q}{s_a} + \frac{L - L_q}{v_f} \tag{3.5}$$

where N_q denotes the number of vehicles in queue, L denotes the length of road segment, L_q denotes the length of queues, and v_f denotes free flow speed. The first term in equation 3.5 represents the time spent moving in the congested queues, and the second term represents the time spent travelling at free-flow speed in the uncongested part of the road segment.

Takaba et al. (1991) extended the sandglass model by further decomposing the travel time on a congested section into two parts (the first and second terms of equation 3.6). The link travel time is now expressed as

$$TT = L_q \frac{k_m}{N_q} - L_q \left(\frac{k_m}{s_a} - \frac{1}{v} \right) + \frac{L - L_q}{v_f} \tag{3.6}$$

where v is the travel speed, k_m is the jam density. To apply this model, jam density, saturation flow rate, and free-flow speed are estimated , while the traffic volume and queue length are collected.

The requirement of a measured queue length in the model may hinder its applicability because it is difficult to collect the queue length directly (Zhang & Kwon 1997). Furthermore, Anderson & Bell (1997) reported that the poor performance of the queuing models was due to their sensitivity to the saturation flow and jam density. These two variables tend to be fixed in queuing models, whereas the reality suggests they tend to vary (Robinson 2005). We tentatively argue that these static queuing models are not suitable for real time applications.

3.3 Data-driven Models

Data driven models relate travel times linearly and/or nonlinearly to the influencing factors and/or their combinations. These models can be expressed in a general form as:

$$TT = G(X, \psi) \tag{3.7}$$

where X denotes a vector of influencing factors (e.g. occupancy, offsets, and green/cycle ratio), and ψ denotes a vector of all the parameters in the regression model G.

3.3.1 Regression based models

Gault & Taylor (1981) proposed a simple model, taking into account the factors such as degree of saturation and signal offset. The model is in the form

$$TT = (1 - \delta)at^* + \delta c\phi^b + K \tag{3.8}$$

in which t^* denotes arrival time during the signal cycle, $\delta = \{^{\text{1 traffic signals are green}}_{\text{0 traffic signals are red}}$, ϕ is measured occupancy, and a, b, c, and K are parameters that are functions of signal offset.

The clear *limitation* is that this model requires the knowledge of arrival time, a variable that cannot be easily obtained from field operations. In recognizing the deficiency of this model based on arrival time, Gault & Taylor (1981) developed an occupancy-based travel time as follows:

$$\begin{aligned} TT &= A\phi + B \\ A &= a_0 + a_1 \frac{L}{v_f} + a_2\chi + a_3 (P_d/P_u) \\ B &= b_0 + b_1 \frac{L}{v_f} + b_2\chi + b_3 (P_d/P_u) \end{aligned} \tag{3.9}$$

where P_d and P_u are defined as the percentage of green time at the downstream and upstream signal, respectively, a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3 are parameters to be determined.

By taking into account the influence of detector locations, Sisiopiku et al. (1994) proposed a general travel time regression model:

$$TT = \frac{L}{v_f} + c_0 + c_1\phi + c_2 P_d + c_3 P_u + c_4 P_s \tag{3.10}$$

where P_s denotes the ratio of the detector setback distance to the link length, c_0, c_1, c_3, c_4,and c_5 are regression parameters.

In general, the above regression models have a simple model structure and are relatively easy to be calibrated if all the data required are available. The major *advantages* of these models is that they are easy to implement in practice. Various factors such as the signal offset and degree of saturation can be easily incorporated into regression models. However, the *drawbacks* of these models are that, first, they are all only for urban segment travel time prediction not for urban route travel time prediction. It is not indicated how to

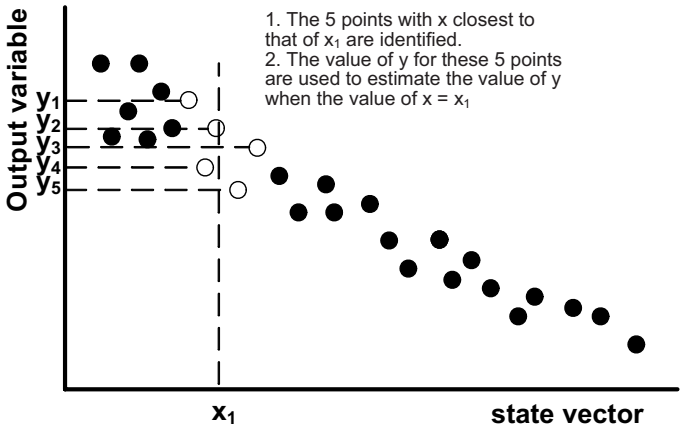


Figure 3.2: Example of the use of the k-NN method when k = 5 (cite from Steve 2005).

extend these regression models for a longer route. Secondly, in the sense that the parameters and coefficients remain fixed, these models are static regression models. We doubt whether those static regression models can produce a good performance in dynamic traffic processes.

3.3.2 K-Nearest Neighbor models

The basic idea behind the k-Nearest Neighbor (k-NN) approach is to match the current state vectors (input variables) with historical observations that have similar input variables. For simplicity, an example with 1-dimensional state vector is depicted in Figure 3.2. Suppose it is desired to use the k-NN method to estimate the value of y when measured state vector $x = x_1$. If k is chosen to be 5, then the 5 historical observations with a value of x closest, that is, the smallest distance (e.g. Euclidean distance), to x_1, are identified. These observations are shown by the shaded points in the diagram. The y values of these 5 closest observations are then used to estimate the value of y when $x = x_1$. It is common to use the arithmetic mean of these five y values.

You & Kim (2000) incorporated the K-NN method with GIS technologies to implement a hybrid travel time forecasting model. A highway and an urban area in Seoul were selected as the test beds. They reported that this hybrid model performed well with satisfying results. More recently, Bajwa et al. (2003a and 2003b) proposed to use genetic algorithm for determining the optimal number of nearest neighbors. For a comprehensive review of K-NN travel time model, the readers can refer to (Robinson 2005).

The clear *advantages* of K-NN are its straightforward concept and ease-of-implementation. However, some *disadvantages* have been identified as well. First, there is no standard rule to select variables which constitute the state vector. It is difficult to determine the spatial and temporal seize of the variables. For the temporal size of the variables, an insufficient size can cause an incomplete image of the traffic searched in historical databases. This results in a poor prediction. However, the large size will require a large computational effort. This can also result in a poor prediction as traffic patterns may consist of unnecessary details which are not affecting present travel times (Bajwa et al. 2003b). Similarly,

the spatial size of variables in the state vector also need to be carefully investigated. Secondly, the K-NN needs a large amount of historical data. Accordingly, increasing the size of the data increases the computation time to search out the K nearest neighbor in a large database. Finally, if the input state vector is located at the boundary of the state space of the existing observations, the results will be subjected to larger bias errors (Robinson 2005).

3.3.3 Markov Chain models

Markov Chain models for estimating delays

Zuylen (1985) firstly described a Markov model for queues at isolated intersections, assuming Poisson arrivals and normally distributed saturation flows. Olszewski independently developed the idea of applying the Markov Chain technique to signal control problems (Olszewski 1990, Olszewski 1994). He showed how the different behavior of the queues can change if a different initial value is assigned, together with a constant demand during the whole evaluation period. Different demand conditions were later analyzed by assuming a stepwise constant demand. Following the approaches adopted by Zuylen (1985) and Olszewski (1994), a similar Markov Chain model was developed by Viti (2006) to compute the evolution in time of the queue length. Viti (2006) proposed a new formula which is applicable for different types of traffic signal controls (i.e. fixed time control and vehicle actuated control), for multiple lanes, and for general traffic patterns. Moreover, the author derived new analytic formulas for the expectation value and the standard deviation of overflow queues in time.

First, the main criticism to those Markov Chain models is on the assumption of homogeneous traffic conditions for a single vehicle type, which is embedded in those models. But in reality the traffic composition is usually a very stochastic variable. This variability may affect the dynamic and stochastic behavior of overflow queues and delays as well. Little research has been carried out of the impact of traffic heterogeneity on the estimation of delays at signalized intersections (Viti 2006).

In addition, most analytical models are based on some assumptions, which are likely reasonable only in the idealized road traffic and signal controlled conditions. For instance, they assume that saturation flow rate is constant, vehicle arrivals follow a certain known probabilistic distribution, the average rate of vehicle arrivals during the evaluation time is constant. However, the saturation flow rate and traffic demand are stochastic and variable even in a short time period. The assumption of being constant is less applicable for the real-time delay prediction. These assumptions restrict the validity of these models for practical studies.

So far, most analytical models are only concerned with isolated intersections. Extending them to a route with more than two intersections is a challenge which needs to be explored.

Markov Chain for estimating travel times

Lin (2004) proposed a Markov Chain model to estimate arterial travel times. The proposed approach was developed by reducing the continuous delays experienced by drivers

at intersections into two discrete states, a state of zero-delay and a state of nominal delay, coupled with a one-step probability transition matrix. For a general case, the delay at intersection i can be expressed in a matrix form as follows:

$$D^{(i)} = [\vartheta\,(0)\,, 1-\vartheta\,(0)] \begin{vmatrix} p_{11}^{(1)} & 1-p_{11}^{(1)} \\ p_{10}^{(1)} & 1-p_{10}^{(1)} \end{vmatrix} \times \ldots \begin{vmatrix} p_{11}^{(i)} & 1-p_{11}^{(i)} \\ p_{10}^{(i)} & 1-p_{10}^{(i)} \end{vmatrix} \begin{bmatrix} E(d^{(i)}) \\ 0 \end{bmatrix} \quad (3.11)$$

where $\vartheta\,(i) = \{^{\text{1 vehicle is delayed at intersection } i}_{\text{0 vehicle is not delayed at intersection } i}$, $p^{(i)}_{\vartheta(i)\vartheta(i-1)}$ denotes the conditional probability of a vehicle being at state $\vartheta\,(i)$ at an intersection i, given that the vehicle was at state $\vartheta\,(i-1)$ at the intersection $i-1$. $E(d^{(i)})$ denotes nominal delay at intersection i, which is computed by a well-defined delay formula, e.g. the popular Webster delay formula (Webster 1958). Although this approach is quite simple, the calibration for this conditional probability is a big challenging in practice. The authors did not provide a procedure for model calibration. Because of the intensive requirement of detailed data (e.g. individual trajectory), we argue that this model is not suitable for practical applications.

3.4 Discussion

Based on the above overview, seven clear impressions can be identified: (1) compared with the amount of literatures on travel time prediction for freeways, very few researches into travel time prediction for urban routes can be found in the literature; (2) among the existing works on travel time prediction, most of them are designed for urban segments, not for urban routes/arterials; (3) none of those models presents how to predict future traffic conditions; (4) very few of them have been validated with empirical data; (5) delay formulas and Markov Chain are not suitable for real time applications of travel time prediction, because they give expectation values over an ensemble of states that may differ considerably from the actual situations; (6) only the simulation model and KNN take into account the traffic flows turning from other streams; (7) due to the difficulties of collecting empirical data of influencing factors, practical investigations of travel time variability for urban routes are very limited.

Table 3.2 summarizes the main features of existing models in literature.

Table 3.2: Overview of existing urban travel time (delay) prediction models

	Spatial Scope	Data for Validation	Real-time Application	Signals	Turning Traffic	Variance
Simulation	route	both	yes	yes	yes	yes
Delay formulas	segment	both	no	yes	no	no
Queuing models	segment	empirical	yes	yes	no	no
Regression models	segment	empirical	yes	yes	no	no
K-NN models	segment	empirical	yes	no	yes	yes
Markov Chain (Zuylen)	segment	simulation	no	yes	no	yes
Markov Chain (Lin)	route	simulation	no	no	no	no

An interesting result of this comparison is the required input data for each model (listed in Table 3.3). Taking a close look into the required input data will show whether the model is feasible for a practical application (also see the validation of the data in Table 3.2). It can be seen that only the K-NN models give a promising performance for real time travel time prediction with empirical data. The following will explore more details of each model.

Table 3.3: Overview of the required input for existing urban travel time (delay) prediction models

	Input Data
Micro. Simulation	individual driver behavior, OD and route choice (or demand and turning fraction)
Macro. Simulation	speed, OD and route choice (or demand and turning fraction)
Delay formulas	saturated flow rate, signal timing, arrival flow, free flow travel time
Regression models	occupancy, signal timing
K-NN models	flow, occupancy
Markov Chain (Zuylen)	initial state, arrival type
Markov Chain (Lin)	delay probability

The limitations of the simulation models include the degree of expertise required for design and maintenance, and the fact that they require the predictions of traffic demand (e.g. OD, route choice, or turning fraction) and supply (capacity) at the model boundaries as inputs (Van Lint 2004, Liu 2004). The major disadvantages lie in the stochasticity and too many degrees of freedom. Microscopic simulation models require many runs to simulate the stochasticity (e.g. different driver behaviors). This time-consuming task is not suitable for the real time application. In the simulation models, there are a lot of parameters that have to be (offline or online) calibrated, which produces too many degrees of freedom. Although microscopic *simulation models* provide us with valuable insight into the mechanisms of traffic flow and queue dynamics, those sophisticated models need to overcome their inherent limitations (discussed above) before they can be successfully implemented in practice.

These *delay formulas* are valid under the condition that the arrival distribution of vehicles is a known probability distribution and vehicles arrive at an intersection with a known initial queue. Then, the process of the queue evolution can be pre-determined. These delay formulas suffer from the inherent assumptions which are most likely not satisfied in real life circumstances. Thus, they serve mainly as scenario evaluation tools, but do not aim to deal with real-time travel time prediction. Similarly, the *Markov Chain models* are all based on hypothesized distributions which are likely inappropriate under real world circumstances. The *queuing theory based models* are too sensitive to two variables, the saturation flow and jam density, which are difficult to be determined in a real environment. It is tentatively argued that these static queuing models are not suitable for real time applications. The *regression models* are static and do not track dynamic traffic processes. Moreover, they are all for urban links, not applicable in a large spatial scope (e.g. urban routes).

The *K-Nearest Neighbor models* require a lot of efforts to design the state vectors. They need to determine the spatial and temporal size of the variables in the state vectors. Since no standard rule exists for how to select the variables which constitute the state vector, a

very difficult task for the model designer exists. Moreover, because the KNN models are location specific, a calibrated model setting that works well on one location may not work at all on other ones.

Finally, those models have not been

3.5 Summary

This Chapter gave an overview of existing urban short term travel time prediction models. Those models have been categorized into either model based or data driven. Both advantages and disadvantages of each model have been highlighted. From the discussion above, it is clear that among the limited number of models, few of them aim at short term travel time prediction for a large scale urban route. Since most of them only focus on urban segments and do not take into account turning traffic flows, they fail to properly work on urban routes in their present forms. Only the simulation and KNN can be used for online/real time travel time prediction. However, the evaluations of simulation models with empirical data are not available (Miska 2007). In addition, the predictions made by the simulation models are never compared to any other model, making their predictive quality of questionable value (Hinsbergen et al. 2007). Furthermore, the simulation models still have to be improved by tackling the problem of online parameter calibration and OD estimation (Van Lint 2004, Miska 2007).

In conclusion, we state that there is great potential for the development of new urban travel time prediction models, particularly using data driven approaches. Also, the new model should be investigated for its robustness under real world measurements. A data driven model will be introduced in the next Chapter.

Chapter 4

Model Development for Urban Travel Time Prediction

4.1 Introduction

In the previous Chapter an overview of the various approaches that have been developed to tackle the urban travel time prediction problem has been presented. It has been found that few research efforts have undertaken urban route travel time prediction. In this Chapter a novel method within the category of data driven approaches is proposed to address urban route travel time prediction. The selection of a data driven approach in this dissertation is based on the discussion in Chapter 3, but does not imply that this approach is necessarily better than model based approaches. As discussed in Chapter 3, both model based and data driven approaches have advantages and disadvantages. The selection of data driven or model based approaches depends on the application.

This Chapter first lists criteria for the model development. The issue of how to choose an appropriate approach for urban travel time prediction is presented. A single segment model based on the State Space Neural Network (SSNN) is used for modeling traffic flows on one single signalized segment. Then, modeling a longer urban route covering several signalized intersections is performed by assembling individual segment models. After deriving mathematically the proposed model, two training methods for model calibration are presented. The proposed model will be evaluated in both a simulation environment and a real time test site. Chapter 5 will show the results of the proposed model evaluated with synthetic data. Chapter 6 will put this model into practice. In the result sections of both Chapters 5 and 6, a comparison is carried out between the results obtained from the proposed model and the simple, but widely used in practice, baseline model.

4.2 Criteria for Model Development

The aim of this dissertation is the development of an accurate and robust short term urban route travel time prediction model which can be implemented in practice. Before the model is presented, several criteria should be clarified:

Criterion 1 *The model should be general and not location-specific, at least in terms of its mathematical structure and the overall input-output relationship. In other words, the model should be applicable for different urban routes.*

Criterion 2 *For practical implementation, the model should be based on available and feasible data collected from existing measurement equipments.*

Criterion 3 *The model should show improved predictive performance compared with a baseline model, which is widely used in practice (see details in Chapter 5).*

Criterion 4 *The model should be able to deal with different traffic conditions (free flow, congested, etc.).*

Criterion 5 *The model should be able to produce reasonable outcomes under the condition of missing and corrupted data.*

4.3 General Design Strategies

4.3.1 Problem Description

Subsequent to the criteria discussed above, some questions are presented to clarify the development of the new model. The answers to these questions will make the main task of this dissertation clear. Also, they will explicitly indicate the advantages and disadvantages of the new model.

Problem 1 *What is a typical urban street?*

- In this dissertation, we define a typical urban street as a road facility which consists of several urban segments (links and intersections) (for details see Chapter 2). There are no minor streets crossing within the urban segments. The traffic only can turn to other urban links at the intersections. This means that the traffic volume will be conserved between the upstream and downstream intersections of the urban segments.
- The traffic signal control strategy is a very important factor and should be taken into account. Different signal control strategies (three strategies are discussed in Chapter 2) produce different signal timings (e.g. red and green time). Note that in a congested condition an actuated control strategy is similar to a fixed time control strategy. More details will be presented in the following sections.

Problem 2 *Which data are available for modeling urban travel time?*

- The data that can be used depend on the available measurement equipments.

- Single loop detectors are usually installed at the upstream of an urban link to measure traffic *volumes* (number of vehicle passages per unit time). Cameras are installed at the start and end of a trip to measure vehicles' license number.
- *Individual travel times* can be calculated by comparing the elapsed time instant for a single vehicle to pass from the start to the end of this trip.
- The *signal timings* (e.g. green time, red time) of each stream are provided by traffic signal controllers.
- Volumes, individual travel times and signal timings are available data. Note that the availability of data is a common case in the Netherlands but probably not in other countries.

Problem 3 *How should the performance of the new model be evaluated?*

- As stated in Chapter 3, very limited research has been conducted on urban route travel time prediction. Therefore, this dissertation will only use a simple baseline model (details in Chapter 5), one which is widely used in practice, to compare the performance of the proposed model. Accuracy and robustness are two measures used to reflect the performance. Simply speaking, the smaller the errors, the better the accuracy. The robustness will test the model in the situation where data are corrupted or missing. This is a common problem in real time traffic data collection systems.

4.3.2 Research Approach

In general, the research roughly consists of four steps: literature review, model derivation, model validation and model application. Chapter 3 has presented a comprehensive literature review. This section will only focus on model derivation, the core of the research. Chapters 5 and 6 will show the results of model validation and application.

To start a model derivation, the primary task is to answer the question: "which approach is appropriate for our problem?"

The literature review of travel time prediction (including freeways and urban networks) showed that a variety of approaches exist. Most are developed considering specific problem requirements. Travel time prediction models can be categorized into either direct and indirect approaches or model based and data driven approaches.

Indirect travel time prediction starts with predicting traffic quantities (e.g. volume, speed, occupancy, etc.) and then translating the predicted traffic quantities into travel time. For example, the traffic flow theory model is an indirect approach. *Direct* travel time prediction is to predict travel time based on previous travel times and is not involved in predicting other traffic quantities. ARIMA is an example of a direct model. The principal difference between an indirect and a direct travel time prediction model is the input data. The indirect models use traffic quantities, except travel times, while the direct models use only travel times. The direct models implicitly assume that the time series of travel time

has an inherent mechanism. It can be used for predicting a future travel time based on previous travel time data. However, travel time has no causal relationship with itself. That is, a travel time is independent of the previous travel times. The dynamic change of travel times is due to external factors (e.g. increasing volumes, traffic signals, accidents, etc.) not in and of itself. Moreover, in real time situations, travel times can be obtained only when vehicles complete their trips. This means that the measured travel times contain 'past' information. For instance, the measured travel time of 10 minutes means that the travel time is 10 minute when vehicles departed ten minutes ago. In this dissertation, an indirect approach for travel time prediction will be used.

Data driven approaches predict travel times using statistical relations, which are derived from historical data (travel times, speeds, volumes, etc.). *Model based* approaches predict travel times by using traffic flow models (based on traffic flow theory). Although the model based approaches provide valuable insight into the mechanisms of traffic flows and queue dynamics, their inherent limitations hinder their applications for urban travel time prediction. The major limitations include computational complexity, the degree of expertise required for design and maintenance, and intensive model/parameter calibration. In addition, they require to predict traffic demand and supply at the model boundaries as inputs (Van Lint 2004, Liu 2004). More details about the distinct features of both model based and data driven approaches have been discussed in Chapter 3. On the contrary, data driven approaches do not require extensive expertise on traffic flow modeling, and they are fast and easy to implement (Dougherty 1995). In particular, a specific neural network approach, the so-called state space neural network (a type of recurrent neural network), has been proven to accurately predict freeway travel time (Van Lint 2004). This dissertation will adopt a hybrid method of combining the state space neural network (data driven) and model based approaches and apply it in an urban context.

In this dissertation, an *indirect* and *hybrid* approach is used in the following chapters.

4.3.3 Basic Concept for Model Development

An elaborate discussion of direct/indirect approaches and model based/data driven approaches has been given. In Chapter 3, a comprehensive overview of existing urban short term travel time prediction approaches was presented. Each approach has been explored in terms of advantages and disadvantages. With these investigations of existing approaches, a data driven approach was chosen due to its ease of implementation in practice and (reasonably) accurate performance. The positive aspect of the data driven approach is that the complex spatiotemporal urban traffic processes can be modeled by learning directly from data instead of building up sophisticated traffic flow models from prior assumptions. In addition, because they 'learn' from data, they can capture subtle functional relationships among the data even if the underlying relationship is unknown or hard to describe. This modeling approach, with the ability to learn from experience, is very useful for many practical problems because it is often easier to obtain data than to have a good theoretical understanding about the underlying laws governing the system from which the behavior is measured (Vanajakshi 2004).

As shown in Chapter 2, an urban signalized route can be schematized by several connected urban segments. Thus, the urban segments are treated as the basic elements of an

urban route. Modeling travel time along a signalized urban route can be conducted by assembling each basic segment model together.

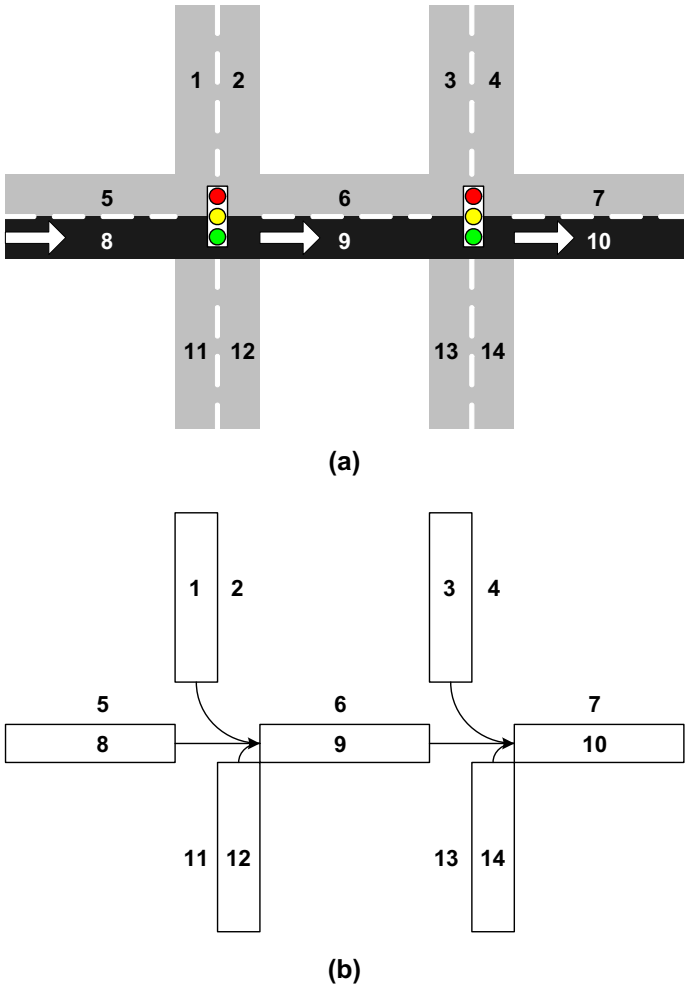


Figure 4.1: Modelling an urban signalized route by decomposing it into urban segments

Figure 4.1(a) illustrates this concept. For example, we are going to model an urban route which starts with urban link 8, then continues with link 9, and finally ends with link 10 (shown in dark in Figure 4.1(a)). The traffic turning left from link 1 and turning right from link 12 will merge with throughput traffic from link 8. The merged traffic traverse through link 9 and merge with left-turning traffic from link 3 and right-turning traffic from link 14, and so on. Thus, link 1, 3, 9, 12 and 14 are very important for modeling this urban route. The rest of the links will not be taken into account because they do not have any influence on the travel time of interest. The final model representation of this urban route is depicted in Figure 4.1(b). Modeling an urban route by decomposing it into urban segments will provide the following benefits:

- general model for urban segments

  Any typical urban route can be decomposed into several urban segments. Conversely, concatenating urban segments can comprise an urban route. The similarity of each segment encourages the development of a general model to describe traffic

dynamics at urban segment level. The general model may have different parameters for different urban segments. But, the mathematical structure of the general model should at least be generic. In other words, the model for one urban segment is suitable for all urban segments with the same properties (e.g. length, the number of lanes, speed limit, signal timing).

- ability to describe traffic propagation

  Figure 4.2 shows three trajectories which have the same travel time from segment 1 to N. This means the same travel time could be derived from different trajectories. The concept is to develop a general model for each segment that is able to track the trajectory. Modeling an urban signalized route can be decomposed into modeling each urban segment of the route. The traffic leaving from the upstream urban segment will enter into the connecting downstream urban segment. With inflow constraint (more details will be provided in the following section), the model is able to limit traffic flows to less than the capacity of downstream links. Travel time of each segment can be calculated based on flow rates. This provides a simple way to describe traffic propagation along the signalized urban street.

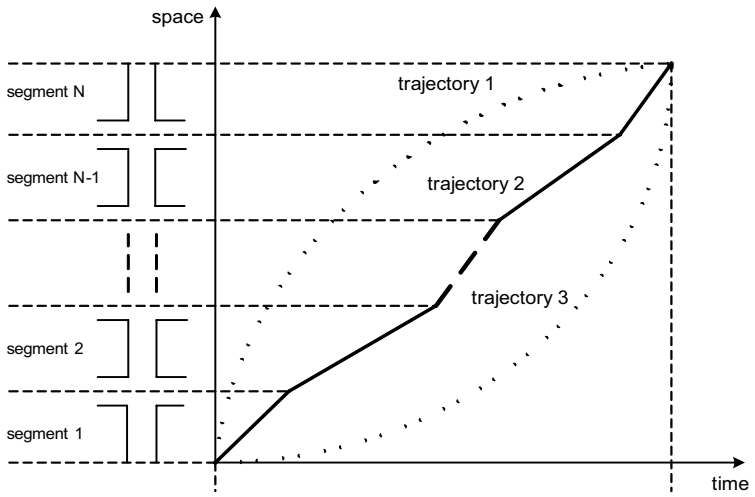


Figure 4.2: Travel time prediction for a route (from segment 1 to segment N) can be conducted by predicting travel time for each segment individually.

- less model parameters

  If more parameters are embedded in the Artificial Neural Networks (ANNs) than the essential ones, this will cause not only intensive calibration tasks but also make the ANNs prone to over fit the data and hence generalize poorly. Except modeling an urban signalized route by separate models for each urban segment (see Figure 4.3(b), we also can treat this route as a whole (Liu et al. 2006) (see Figure 4.3(a)). That is, all the spatially separated inputs (volume measurements from each urban segment) are augmented in a single input vector. This increases the number of parameters. For instance, let assume a route with 3 segments, each segment has 3 input variables, and each segment requires 5 hidden neurons. Thus, the total

number of weight parameters for an ANN model with one single input vector is 375 ($3 \times 3 \times 3 \times 5 + 3 \times 5 \times 3 \times 5 + 3 \times 5 \times 1$), whereas the number for a ANN model with 3 basic segment input vectors is 135 ($3 \times 3 \times 5 + 3 \times 5 \times 5 + 3 \times 5 \times 1$).

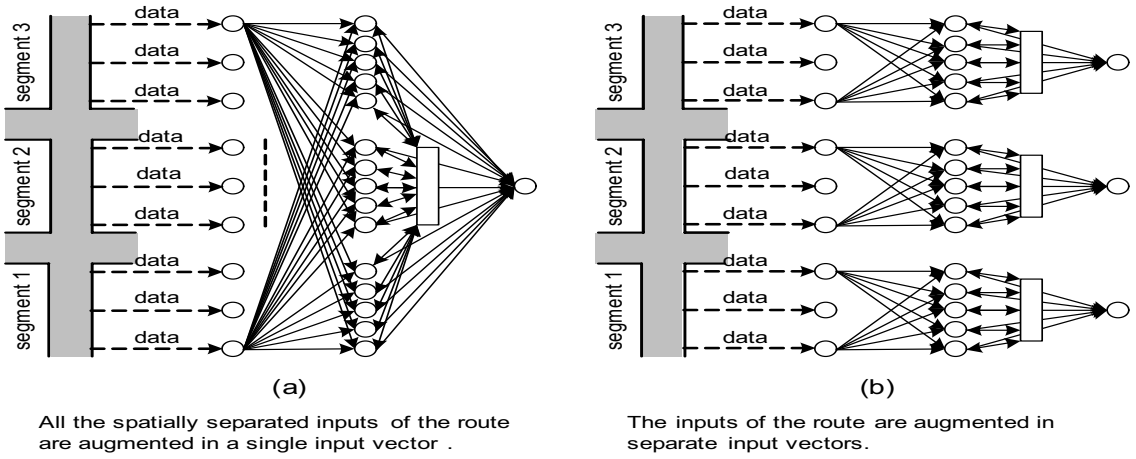


Figure 4.3: The spatially separated inputs can be augmented in two ways: (a) all in a single input vector; (b) in separated input vectors.

Based on this concept, the ensuing sections will first present a generic model for modeling urban segment travel time, and then show how to concatenate generic models to model urban route travel time. In the following sections, the urban segment and urban route travel time prediction models will be referred to as the **USEG** model and the **UROU** model, respectively.

4.4 Model Development for Urban Segment Travel Time Prediction (USEG)

The USEG model, like any other data driven model, can be formulated as follows

$$TT(p) = G(u(p), W) \tag{4.1}$$

where $u(p)$ and $TT(p)$ denote the inputs (the volume measurements and signal timings) and outputs (the travel time measurements) respectively, at time interval p. W denotes the vector of all the parameters in the data driven model $G(.)$. The main task of creating a data driven model is to find out the relationship between inputs and outputs, that is, to set up the model $G(.)$ and find out the appropriate parameters. In the following section, the USEG structure selection will deal with the problem of setting up the model $G(.)$. In the section about training the USEG two different approaches for finding the appropriate parameters will be presented.

4.4.1 USEG Structure Selection

First, we present some basic terms used for artificial neural network (ANN). ANN models often consist of a large number of simple neuron-like processing *neurons* (*nodes)*, organized in *layers*. Every node in a layer is connected with all or a selection of the nodes in the previous layer. Each connection may have a different strength, a so called *weight*. These weights are the adaptable parameters in an ANN.

Since the early 1990's ANNs have been widely used in transportation engineering. A comprehensive review of the uses of ANNs in the field of transportation engineering can be found in (Dougherty 1995). For a concise overview of modeling travel time prediction with ANNs we refer to (Liu et al. 2006). It was found that most of those ANN models were developed for freeway travel time prediction.

There are many different kinds of ANN models (Bishop 2005). From the perspective of network topology, ANN models can be categorized into feed-forward and feedback. In a *feed-forward* ANN (FNN), the connections between nodes do not form cycles. Data enter at the input layer and pass through the neural network, layer by layer, until they arrive at the output layer. In a *feedback* ANN, there are cycles in the connections. These cycles act as a short term memory, allowing them to dynamically deal with input and output patterns. Among feedback ANNs, the State Space Neural Network (SSNN) has the ability to learn temporal sequences of spatial patterns.

In this dissertation, we choose to use the SSNN instead of a FNN based on the following reasons:

- The state of a particular road section (average speed and vehicular density) is determined completely on its previous state and the inputs in the previous time period (Hoogendoorn & Bovy 2001). FNNs are static, that is, they do not take memory into account from the previous states. In contrast, the SSNN enable the previous states to be temporally memorized into the neural network (Mandic & Chambers 2001). Therefore, we argue that the SSNN is more suitable than FNNs for the nonlinear dynamic problem of urban travel time prediction.
- If FNNs would use historical input data, they require a prior choice of which inputs and at which time lag is needed to capture the dynamics of the problem at hand (Van Lint 2004). In contrast, the SSNN is able to avoid the selection of input settings. The feedback (memory) mechanism in the SSNN allows the inputs to be fed at consecutive time instants sequentially. A clear advantage of the SSNN is that the selection of an input time lag is not required.

4.4.2 Mathematical Description of State Space Neural Network

The state space neural network (SSNN) model is a First Order Context Memory Neural Network (Kremer 2001) consisting of four layers (shown in Figure 4.4). The *input layer* $X(p)$ receives the input data (incoming volume and green time), and distributes them to the *hidden layer*. The hidden layer vector $S(p)$ in period p is calculated from the input vector $X(p)$ and context layer vector $S(p-1)$ of the previous period $p-1$, which stores

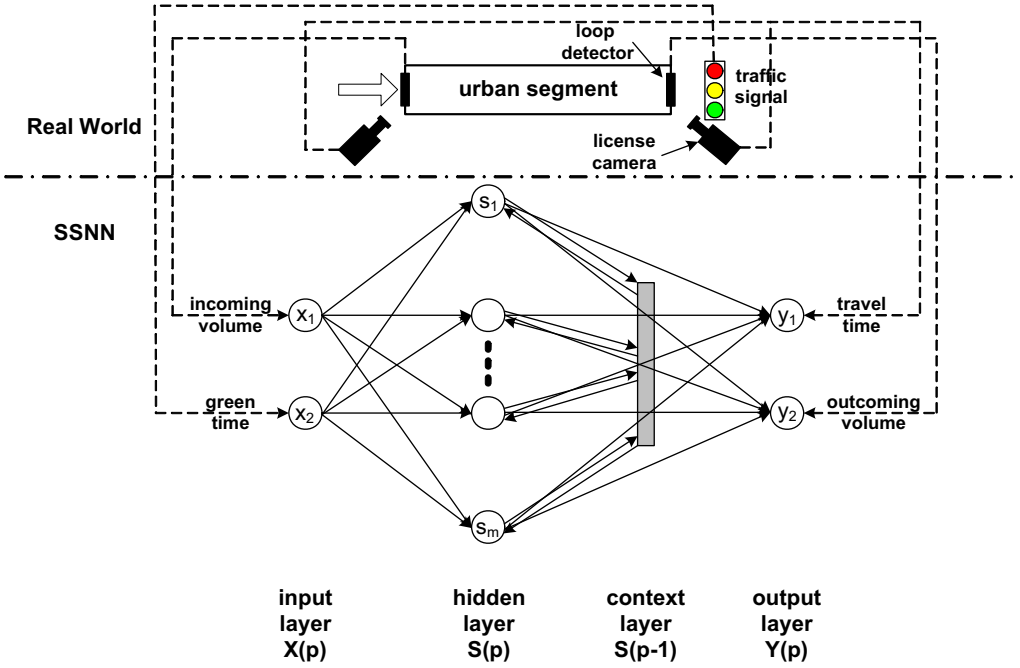


Figure 4.4: State Space Neural Network (SSNN) topology for short term urban travel time prediction.

the hidden layer states. The output layer finally processes the hidden layer outputs and produces the output data (travel time and outgoing volume).

Input layer:

$$X(p) = \begin{bmatrix} x_1(p) \\ x_2(p) \end{bmatrix} = \begin{bmatrix} q^{in}(p) \\ g(p) \end{bmatrix} \tag{4.2}$$

where $x_i(p)$ denotes the value of the ith input neuron, $q^{in}(p)$ denotes incoming volume during time period p, and $g(p)$ denotes green time during time period p.

Hidden layer:

$$S(p) = \begin{bmatrix} s_1(p) \\ \vdots \\ s_m(p) \end{bmatrix} = \begin{bmatrix} \Phi\left(\sum_{i=1}^{2} w_{i,1}^{ah} x_i + \sum_{j=1}^{m} w_{j,1}^{ch} s_j(p-1) + v_1^h b_1^h\right) \\ \vdots \\ \Phi\left(\sum_{i=1}^{2} w_{i,m}^{ah} x_i + \sum_{j=1}^{m} w_{j,m}^{ch} s_j(p-1) + v_m^h b_m^h\right) \end{bmatrix} \tag{4.3}$$

where s_m denotes the value of the mth hidden neuron, $w_{i,m}^{ah}$ denotes the weight connecting the ith input neuron and the mth hidden neuron, $w_{j,m}^{ch}$ denotes the weight connecting the

jth context neuron and the mth hidden neuron, v_m^h denotes the weight of bias associated with the mth hidden neuron, b_m^h denotes a bias with fixed value 1 for the mth hidden neuron, $\Phi(.)$ is the transfer function.

Context layer:

The context layer only stores the hidden layer neurons' one-step previous output. Thus, this layer is not involved in any calculation.

Output layer:

$$Y(p) = \begin{bmatrix} y_1(p) \\ y_2(p) \end{bmatrix} = \begin{bmatrix} \Phi\left(\sum_{k=1}^{m} w_{k,1}^{ho} s_k(p) + v_1^o b_1^o\right) \\ \Phi\left(\sum_{k=1}^{m} w_{k,2}^{ho} s_k(p) + v_2^o b_2^o\right) \end{bmatrix} \tag{4.4}$$

where $y_i(p)$ denotes the value of the ith output neuron, $y_1(p)$ denotes departure travel time, $y_2(p)$ denotes outgoing volume, $w_{k,i}^{ho}$ denotes the weight connecting the kth hidden neuron and the ith output neuron, v_i^o denotes the weight of bias associated with the ith output neuron, b_i^o denotes a bias with fixed value 1 for the ith output neuron, $\Phi(.)$ is the transfer function.

4.4.3 USEG Training

Training the SSNN refers to finding the appropriate parameters (weights and bias) which minimize an objective function. The SSNN can be trained in a supervised manner, given sufficient data pairs (inputs and outputs). There are many possible choices of the objective function which can be used, depending on the particular application. For urban route travel time prediction the primary objective of training the SSNN is to minimize the errors between predictions and observations:

$$E(W) = \frac{1}{2}\sum_{p=1}^{M}\left(Y(p) - \widetilde{Y}(p)\right)^2 \tag{4.5}$$

where W denotes the parameters in the SSNN model, $Y(p)$ denotes the output calculated from the SSNN, $\widetilde{Y}(p)$ denotes the desired output, and M denotes the total number of data pairs in the training data set.

Two main training algorithms have emerged: *batch training*, in which parameter optimization is carried out with respect to the entire training data set simultaneously, and *incremental training*, where model parameters are updated after the presentation of each training example. The batch training and incremental training usually are also called the offline and online training, respectively. The batch training and incremental training methods can be used in the different phases of travel time prediction procedure. Normally, batch training is applicable for offline parameter optimization. The parameters

which minimize the objective cost function are stored as the default optimal parameters, which are used as the basis for online application. Instead of treating parameters as fix values, incremental training methods use the available data to steer the model parameters to values closer to the realized ones at each time step.

Batch training with Regularization

More parameters and larger weights can cause an excessive variance of the outputs (Geman et al. 1992), and then lead to poorer generalization. A traditional way of dealing with the negative effect of the large weights is regularization. The idea of regularization is to make the neural network response smoother through modification in the objective function by adding a penalty term for more and larger parameters, for example, the sum of squared parameters. In this dissertation, we propose to use Levenberg-Marquardt and Bayesian Regulation (LM-BR) algorithm (Mackay 1992). We present a concise description of LM-BR in the following paragraphs. More details of the algorithm used can be found in Appendix D, which are largely based on (Mackay 1992, Foresee & Hagan 1997, Bishop 2005).

Let D represent the data set and W represent the vector of the neural network parameters. The objective function becomes to minimize a sum-of-squares error function while at the same time trying to minimize the sum of squares of weights

$$F(W) = \beta E_D + \alpha E_W = \beta \sum_{p=1}^{M} \frac{1}{2} \left(Y(p) - \widetilde{Y}(p) \right)^2 + \alpha \sum_{i=1}^{N} \frac{1}{2} W_i^2 \tag{4.6}$$

where N denotes the total number of weights in parameter vector W, and α, β are objective function parameters which dictate the emphasis of the training. The regularization task is to find the optimal values of α and β so that the trained model will have a good performance but is not over fitted. In other words, the central to the LM-BR algorithm is the objective to maximize the posterior probability of a particular weight vector given the training data D, and the regularization parameters α and β. Assuming the noise and the prior distribution of the weights are both Gaussian distributed according to $N(0, 1/\beta)$ and $N(0, 1/\beta)$ respectively, the posterior distribution of the weights can be written:

$$\begin{aligned} P(w|D, \alpha, \beta) &= \frac{P(D|w, \beta) P(w|\alpha)}{P(D|\alpha, \beta)} \\ &= \frac{1}{P(D|\alpha, \beta)} \left(\frac{1}{Z_D(\beta)} \exp(-\beta E_D) \frac{1}{Z_W(\alpha)} \exp(-\alpha E_D) \right) \\ &= \frac{1}{P(D|\alpha, \beta)} \left(\frac{1}{(\pi/\beta)^{P/2}} \exp(-\beta E_D) \frac{1}{(\pi/\alpha)^{P/2}} \exp(-\alpha E_D) \right) \\ &= \frac{1}{P(D|\alpha, \beta)} \left(\frac{1}{Z_F(\alpha, \beta)} \exp(-F(W)) \right) \end{aligned} \tag{4.7}$$

In (Foresee & Hagan 1997), it is shown that the maximum likelihood estimates for α and β can be calculated as follows

$$\alpha^{MP} = \frac{\gamma}{2E_W\left(W^{MP}\right)} \tag{4.8}$$

$$\beta^{MP} = \frac{M-\gamma}{2E_D\left(W^{MP}\right)} \tag{4.9}$$

where $\gamma = N - 2\alpha^{MP} \cdot trace\left(H^{-1}\right)$ is called the effective number of parameters. The Bayesian optimization of the regularization parameters requires the computation of the Hessian matrix at the minimum point W^{MP}. Foresee & Hagan (1997) proposed using the Gauss-Newton approximation to the Hessian matrix H as follows

$$\widehat{H} = \beta J^T J + \alpha I \tag{4.10}$$

where J denotes the Jacobian matrix of the objective function with respect to the SSNN weights, $J = \frac{\partial E(W)}{\partial W}$, and I denotes the identity matrix. In short, the training procedure can be summarized as

Step 1. Initialize weights W and the objective function parameters α and β.

Step 2. Use the LM algorithm to calculate new weights with fixed α and β based on the output error $e(W)$

$$W^{new} = W^{old} - \left(\widehat{H}(W) + \lambda I\right)^{-1} J^T(W)\, e(W) \tag{4.11}$$

Step 3. Optimize α and β given new weights.

Step 4. If stop criteria met (minimum performance goal, maximum number of epochs) then stop, otherwise continue with step 2.

More details on the algorithm used can be found in Appendix D.

Incremental training

Since the parameter set obtained from the batch training represents the average condition over the period represented in the data, it is not sensitive to the variability of prevailing traffic conditions. For example, the change of weather or surface conditions may result in variations in the parameters over time. Therefore, the objective of the incremental training is to introduce a systematic procedure that will use the available data to steer the model parameters to the values closer to the ones that are most applicable for the present situation. The parameters obtained from batch training can be used as the initial estimates for the incremental training.

The global Extended Kalman Filter (EKF) training algorithm was introduced for incremental training neural networks (Haykin 2001). A neural network's behavior can be formulated by the following nonlinear discrete-time system:

$$W_k = W_{k-1} + \delta_{k-1} \tag{4.12}$$

$$Y_k = G\left(X_k, W_k\right) + \zeta_k \tag{4.13}$$

where W is the parameter set of the SSNN which is assumed to correspond to a stationary process (random walk), δ and ζ are process noise and measurement noise respectively, X and Y are input and output respectively, and $G(.)$ denotes the SSNN model. The algorithm to solve this nonlinear discrete-time system is listed as follows

Algorithm 1 *Incremental Training Neural Networks*

1 Initialize the estimate parameter W and the error covariance P_0 with

$$\widehat{W}_0 = E\left(W\right) \tag{4.14}$$

$$P_0 = E\left[\left(W - \widehat{W}_0\right)\left(W - \widehat{W}_0\right)^T\right] \tag{4.15}$$

Algorithm 2 *1.*

2 Time update for time step $k = 1, 2, \ldots$

$$\widehat{W}_{k|k-1} = \widehat{W}_{k-1} \tag{4.16}$$

$$P_{k|k-1} = P_{k-1} + E\left[\delta_{k-1}\delta_{k-1}^T\right] \tag{4.17}$$

3 Measurement update for time step $k = 1, 2, \ldots$

$$\varepsilon_k = \widetilde{Y}_k - G\left(X_k, \widehat{W}_{k|k-1}\right) \tag{4.18}$$

$$K_k = \frac{P_{k|k-1}J_k^T}{P_{k|k-1}J_k^T P_{k|k-1} + E\left(\varepsilon_k \varepsilon_k^T\right)} \tag{4.19}$$

$$\widehat{W}_k = \widehat{W}_{k|k-1} + K_k\varepsilon_k \tag{4.20}$$

$$P_k = P_{k|k-1} - K_k J_k P_{k|k-1} \tag{4.21}$$

Roughly, such an online learning algorithm reads as follows, where $y_k = G\left(\psi, u_k\right)$ depicts a data-driven model.

1) Make a prediction $y_k = G\left(\psi_k, u_k\right)$.

2) Set $k = k+1$, and update model weights ψ_{k-1} with error $\varepsilon_{k-1} = d_{k-1} - y_{k-1}$ yielding the updated weights ψ_k.

3) Go to step 1.

Note that each prediction is based on the one-step updated weights. This implies that each time step has one observation. In a travel time prediction context, this one-step-ahead procedure is clearly not applicable since a realized (actually measured) travel time d_k is not available at time instant $k+1$ but, in fact, after $k + d_k$ time periods. Van Lint

presented a new extended Kalman filter (EKF) based online-learning approach, called the online-censored EKF method, which can be applied online and offers improvements over a delayed approach in which learning takes place only as realized travel times are available (Van Lint 2006).

Consider that at some time period p, the last realized travel time d_m is available from vehicles departing at period m, where $m = p - d_m$. Although, for those periods k, $m < k < p$, no realized travel times are available yet, a censored observation (in fact, a lower-bound value) is given by

$$d_k > d_k^*(p) = p - k \tag{4.22}$$

Although the true prediction error $\varepsilon_k = d_k - y_k$ [where $y_k = G(\psi_k, u_k)$] is not available, again, a censored observation of this error is given by

$$\varepsilon_k^*(p) = d_k^*(p) - y_k \tag{4.23}$$

Due to 4.22, 4.23 represents a monotonically increasing lower bound of the true error ε_k, i.e.,

$$\varepsilon_k^*(p) < \varepsilon_k, m < k < p \tag{4.24}$$

At each time period $p > k$ for which no realized travel time d_k of vehicles departing at k is available, the censored error 4.24 provides an incremental estimate of the model prediction error.

Letting

$$\zeta_k(p) = \varepsilon_k^*(p) - \varepsilon_k^*(p-1) > 0 \tag{4.25}$$

where

$$\sum_{p=k+1}^{m+dm+1} \zeta_k(p) = \varepsilon_k$$

implies that for a particular departure time k for which no realized travel time is available, the weights ψ_k can be updated stepwise at each $p > k$ by substituting 4.25 into 4.20. Such an update is retained if this update indeed improves the model performance, that is, if

$$d_k^*(p) - G(u_k, \psi_k) > d_k^*(p) - G(u_k, \psi_{k+1}) \tag{4.26}$$

which is the case if and only if

$$G(u_k, \psi_{k+1}) > G(u_k, \psi_k) \tag{4.27}$$

In all other cases, the update is discarded. Constraint 4.27 implies that if the parameter update results in a larger predicted travel time than before, it is retained; otherwise, it is discarded, in which case, $\varepsilon_k^*(p)$ must be reset to zero. Intuitively, this procedure makes sense. For example, in cases where travel times (of, for example, 10 mins) are an order-of-magnitude larger than the unit of discrete time k (of, for example, 1 min), the lower bound of equation 4.22 will initially (as p is only a few time steps away from k) be much smaller than free-flow travel times. Adapting the weights to these clearly underestimated travel times would not improve performance at all. In situations of congestion build-up, during which travel times tend to increase, it is clear that, according to equation 4.27, updates are retained only if these contribute to the increasing trend. In case of declining congestion, during which travel times tend to decrease, equation 4.27 has no effect since, in those cases, realized travel times will become available increasingly faster.

Last, note that at any particular time period p, there will be a number of past time periods k for which no realized travel times are available yet. This means that per time period p, possibly more than one weight can be applied with censored errors. In this paper, this is done sequentially, whereas at each update, equation 4.27 is evaluated with respect to the last weight update, which could also have been applied during p.

4.4.4 Some Important Issues for Implementation

SSNN Structure Optimization

As shown above, the number of input and output neurons is fixed by nature of our problem. Because most theoretical works show that a single hidden layer is sufficient for ANNs to approximate any complex non-linear function with any desired accuracy (Cybenko 1989, Hornik et al. 1989), one hidden layer is adopted in this dissertation. The context layer has the same number of neurons as the hidden layer. Thus, the entire structure of the SSNN is determined by the hidden layer. In other words, the selection of the appropriate number of hidden neurons is the major concern of designing a SSNN. To determine the number of hidden neurons is a fundamental trade-off between model complexity and model generality. A large number of hidden neurons imply more model parameters, which provide more descriptive and predictive power of the model. On the other hand, a complex model (more parameters) is prone to over fit the data and hence generalize poorly. But, a too simple model (less parameters) is simply inadequate to capture all the nonlinearity of the problem.

There are some algorithms, including pruning and growing algorithms, to determine an 'optimum' number of neurons. *Growing algorithms* initiate networks with relatively small number of neurons, and allow new neurons to be added during training (Bishop 2005). On the contrary, pruning algorithms start with a relatively large network and gradually remove neurons (Bishop 2005). This dissertation adopted the most common way of determining the optimal number of hidden neurons by a sensitivity analysis. A trial and error procedure using different number of neurons was used. Five different architectures of USEG were developed (with 2, 4, 6, 8, and 10 hidden neurons).

Initial Weights

Training neural networks with different initial weights might result in different weight settings. Although some authors (Mackay 1992, Drago & Ridella 1992, Wessels & Barnard 1992, Chen et al. 2000) have proposed approaches to initiate weights, the weight distribution however in a trained neural network is not well understood, which is due to a strong problem dependency for the weight distribution. There is no training algorithm that guarantees to find a global optimum. One way to alleviate this problem and to increase the likelihood of obtaining near-optimum local minima is to train several neural networks with a random set of initial weights, and choose the one with the lowest error. However, in practical application, we actually don't know which network performs best for future unseen data. There may be many parameter sets within a model structure that are equally acceptable as simulators of a dynamic process of interest. Consequently, instead of choosing one best single USEG model, we may make predictions based on an ensemble of neural networks trained for the same purpose (Sharkey 1996). In this dissertation, the idea of the ensemble prediction is adopted, and the simple average ensemble method is used. For each USEG model, we train it ten times so as to get 10 neural networks, and choose 5 best ones according to their training performances. With the selected neural networks, we get 5 outputs for each segment travel time prediction. Then we take a simple average of the 5 outputs to be the final output.

Transfer Function

Different transfer functions such as sigmoid, logistic, hyperbolic and linear etc. have been widely used (Demuth & beale 1998) for prediction applications (e.g. weather forecast, economy prediction, water flow forecast, etc.). Among them sigmoid function shows good convergence properties for the training algorithm through its differentiability and ensured stability of the method through its finite range of values (Vanajakshi 2004, Van Lint 2004, Palacharla & Nelson 1999). In this dissertation, the sigmoid function is selected.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.28}$$

This transfer function takes the input, which may have any value between plus and minus infinity, and maps the output into the range 0 to 1. Clearly, the choice of the sigmoid function is arbitrary. The selection of the sigmoid function is due to the property of its monotone increasing behavior. This property is to a certain extent in line with the obvious relation that travel times increases with volumes. That is, an increase of input values (volume) produce an increase of output values (travel time).

Initial Internal States

Internal states actually refer to the value of hidden neurons. As shown above, the output (travel time) is a product of hidden neurons. Thus, the initial internal states correspond to initial output (travel time). If we sort data pairs (inputs and outputs) starting with free flow condition (earlier in the morning), the initial internal states can be regarded as free flow

condition as well. In free flow condition the volume has (approximately) no influence on the travel time. Because of the monotone increasing property of sigmoid transfer function, it is reasonable to assign zero-value for those initial internal states.

Time Resolution

The time resolution of the SSNN is the unit of one time step. Generally, a too small time resolution requires a high performance of measurement equipments and a great demand of communication networks for data transmission. Also, the too small time resolution takes more time steps to calculate the same length of a time period than a large time resolution, that is, the small time resolution requires more computation time. In addition, for travel time observations, the too small time resolution will cause more time steps likely to have null values (because there are no matched vehicles). This will increase the work load of filling in the null data because the training of a neural network requires a pair of input and output. In contrast, a too large time resolution might smooth significant changes during this time resolution. The smoothing could neglect the details of the traffic dynamics during the time step.

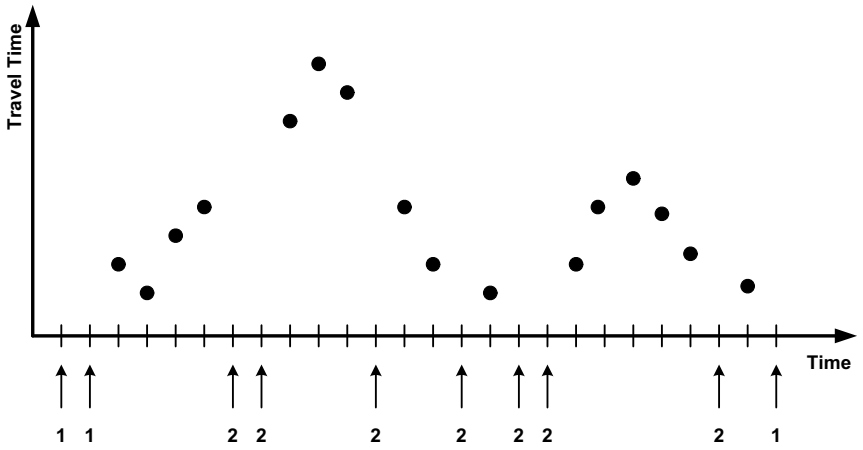
For urban signalized segment applications, we therefore choose the minimal cycle time as a suitable time resolution. The reasons are twofold: first, the cycle time is normally around 1 minute (not too small and too large), which is the common choice in practice as well. In addition, the variable of green time can be removed out of the USEG when the time resolution is equal to the cycle length (the green time is constant at each time step for fixed time controlled intersections). Neglecting the green time will not influence the USEG because the constant green time can be potentially treated as a bias in USEG.

Data Preparation

As shown above, there are four different variables involved in the USEG. Incoming and outgoing volume are measured by single loop detectors. Green times can be obtained from the signal controllers, especially for fixed time control the constant green time is known in advance. The travel times are provided by license plate cameras. All these variables are time series data. During the training phase, these variables are fed into the USEG at each time step. In this dissertation, we consider that signal timing data can be correctly measured or provided. The volumes and travel times need to be treated carefully (see details in Chapter 6).

Two crucial issues with respect to travel time measurements are taken into account in this dissertation. First, travel time measurements might include outliers, which will be discussed in detail in Chapter 6. Second, travel time is the only variable that might have no observation for some time steps due to no vehicle finishes the trip of interest. Thus, we need to fill in these empty gaps because training neural networks requires a pair of input and output. Here, we only focus on how to deal with the second issue because simulation data used in this Chapter can be considered 100% correct (no outliers). The detection of outliers will be explored in Chapter 6 for real time application.

Figure 4.5 shows a typical representation of travel time measurements. Two kinds of empty gaps occur frequently. The difference between type 1 and type 2 is that the later



Type 1 Time step without measurement need to be filled in, which only has one available neighbor observations before or after this time step. Normally, this occurs at the beginning and end of time series.

Type 2 Time step without measurements need to be filled in, which has two available neighbor observations before and after this time step. Normally, this occurs in the middle of time series.

Figure 4.5: Two different types of empty gaps in the time series of travel time observations.

has two 'neighbor' observations (before and after the time step of interest), while the former only has one 'neighbor'. Type 1 happens usually in the early of the morning or late in the night, when there are few vehicles running on road networks. Type 2 occurs often during the daytime. To fill in the empty gaps of type 2, linear interpolation method is used in this dissertation. Since type 1 occurs in the free flow conditions (early in the morning or late in the evening), it is appropriate to fill in with a default value (e.g. the median of free flow travel times). A simple replacement strategy of giving default value is applied for type 1.

Data Scaling

It is desirable to scale the data before training the neural network. There are two reasons for this. First, the reason for scaling the data is that the desired outputs of the neural network should fall into the range of the output transfer function. For example, if a standard sigmoid transfer function is used for the output neuron, then the desired neural network output should fall in the range between 0 and 1. Secondly, the scaled data cover the same range for all variables, and therefore errors in each variable contribute in the same proportion to the changes in the neural network weights. In other words, the weights are able to learn at the same 'speed' (Zijderveld 2003).

The simplest way of data scaling is the so called linear scaling. When the data is spread evenly over its range, the data set can be squashed into a desired range. For example, to linearly scale a datum ρ of data set Ω within the range $[a, b]$, the new scaled value ρ^* can be calculated as:

$$\rho^* = a + \frac{(b-a)}{\rho_{\max} - \rho_{\min}} (\rho - \rho_{\min}) \tag{4.29}$$

where $\rho_{\min}$ and $\rho_{\max}$ are the minimal and maximal value of data set Ω, respectively.

To rescale from ρ^* to ρ, the inverse of a linearly scaled value is calculated as

$$\rho = \rho_{\min} + \frac{\rho_{\max} - \rho_{\min}}{(b-a)} \left(\rho^* - a\right) \tag{4.30}$$

4.5 Model Development for Urban Route Travel Time Prediction (UROU)

The above section has elaborated the basic ingredient (USEG) for an urban route travel time prediction model (UROU). In this section we present how to concatenate USEG models for modeling urban route travel time.

4.5.1 Inputs of USEG

After we have built up a basic neural network model (USEG) that describes the traffic flows on a single signalized segment, an urban route can be modeled by assembling the basic models. Two types of segments can be classified as: the *boundary segment* which have no inflow traffic from upstream segments, and the *internal segments* which receive inflow traffic from upstream segments (shown in Figure 4.6(b)). For example, segment 1 and segment 6 of intersection k are a boundary segment and internal segment, respectively.

The internal segments receive the outflow from upstream segments, while the boundary segments are fed with boundary inputs directly. Each segment propagates the inflow to a downstream segment. For example, at intersection k, segment 6 receives throughput traffic flows from segment 5, left-turning traffic flows from segment 1 and right-turning traffic flows from segment 8 (shown in Figure 4.6(a)). Since the rest segments (2, 3, 4, and 7) do not propagate traffic flows to the segment 6, they are not used for modeling. As a result, the physical urban route (filled in black color shown in Figure 4.6(a)) can be modeled by concatenating several USEGs (shown in Figure 4.6(b)).

For each segment ki, the ith branch of intersection k, the USEG can be expressed as follow:

$$\begin{bmatrix} q_{ki}^{out}(p) \\ TT_{ki}(p) \end{bmatrix} = G\left(\begin{bmatrix} q_{ki}^{in}(p) \\ g_{ki}(p) \end{bmatrix}, W_k^i\right) \tag{4.31}$$

where W_k^i denotes the vector of all the parameters in the USEG model $G(.)$ of the segment of interest, $q^{in}(p)$ denotes incoming volume during time period p, $g(p)$ denotes green time, $q^{out}(p)$ denotes outgoing volume, and $TT(p)$ denotes travel time.

For any segment ki, $q^{out}(p)$ and $TT(p)$ can be calculated from $q^{in}(p)$ and $g(p)$. Thus, for each time period p, $q^{in}(p)$ and $g(p)$ are needed to be updated. For fixed time control signalized urban routes, $g(p)$ are constant and known in advance. Here, we only consider how to update (predict) $q^{in}(p)$.

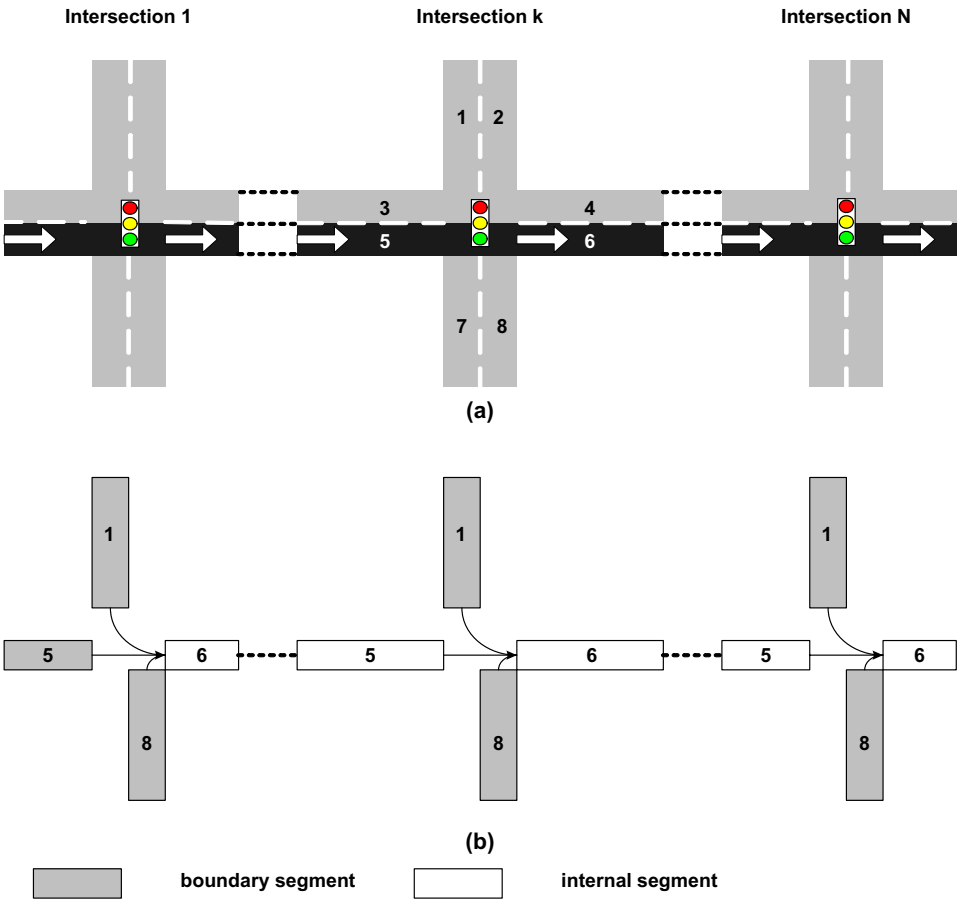


Figure 4.6: Modelling an urban route by concatenating UROU

- Boundary segments

  To predict the inflow volume of boundary segments is similar, to some extent, to (dynamic) OD/demand prediction. A detailed overview of OD/demand prediction can be found in (Lindveld 2003). Since OD matrices can hardly be measured (except for closed networks with vehicle identification at the entry and exit points) it is not easy to assess the existing methods with real world data (Miska 2007). In this dissertation, we do not intent to introduce a new way of predicting the inflow volume of boundary segments. A simple method is used in this dissertation, but can be replaced by other methods in the future. The concept of this simple method is that the real time volume is proportional to the historical profile. That means, the prediction is corrected with a factor β, deriving from the actual measurements $q_{ki}^{in}(p)$ and the historical (average) data $\widehat{q}_{ki}^{in}(p)$.

$$\begin{aligned} q_{ki}^{in}(p+1) &= \beta \times \widehat{q}_{ki}^{in}(p+1) \\ \beta &= \frac{q_{ki}^{in}(p)}{\widehat{q}_{ki}^{in}(p)} \end{aligned} \tag{4.32}$$

- Internal segments

  Let segment 6 be an example. The inflow volume for the segment 6 of intersection

k can be calculated as:

$$q_{k6}^{in}(p+1) = \alpha_l(p+1) \times q_{k1}^{out}(p) + \alpha_t(p+1) \times q_{k5}^{out}(p) + \alpha_r(p+1) \times q_{k8}^{out}(p) \tag{4.33}$$

where $\alpha_l(p)$, $\alpha_t(p)$ and $\alpha_r(p)$ are the left-turning, throughput and right-turning fractions. Obviously, the predicted inflow volume is based on the prediction of the three fractions. Due to the absence of OD prediction, we assume that the dynamic of the three fractions follows a random-walk process.

$$\begin{aligned} \alpha_l(p+1) &= \alpha_l(p) + \xi_l \\ \alpha_t(p+1) &= \alpha_t(p) + \xi_t \\ \alpha_r(p+1) &= \alpha_r(p) + \xi_r \end{aligned}$$

This is a common strategy applied in practice. It is reasonable to assume the traffic would not change significantly within a small time step. For instance, the next step fractions can be assumed to be equal to the previous ones plus noise.

- Inflow constraint

  Note that the above computation of the inflow is based on an implicit assumption that the downstream link have no restriction on its reserve capacity. However, in congested conditions the reserve capacity of the downstream link does have strong influence on the inflow of upstream links. The maximum inflow is the minimum of the computed value and the rest of the reserve capacity. A simple way to compute the rest of the reserve capacity is to divide the length of the downstream link by the average vehicle space (vehicle length plus distance between two consecutive vehicles), and minus the number of vehicles present on the link of interest. The maximum inflow is a constraint in order to avoid the over load of the downstream link.

4.5.2 Travel Time Prediction

The previous section has shown how to calculate the inputs (volumes) for each segment. With those calculated inputs, travel time on each segment can be computed. For instance, the $USEG_k^5$ of segment $k5$ produce travel time TT_{k5} and q_{k5}^{out} (see Figure 4.7). The q_{k5}^{out} merges with the left-turning flow from segment $k1$ and the right-turning flow from segment $k8$, and then the merged flows are fed into the $USEG_k^6$ of segment $k6$ to calculate travel time TT_{k6}. This procedure is thus repeated until the travel time of the final segment is calculated. Finally, the prediction of the route travel time can be conducted by summing of travel times on each segment

$$TT = \sum_{k=1}^{N} \sum_{i=5}^{6} TT_{ki} \tag{4.34}$$

where k denotes the intersection, i denotes the branch of intersection, N denotes the total number of intersections.

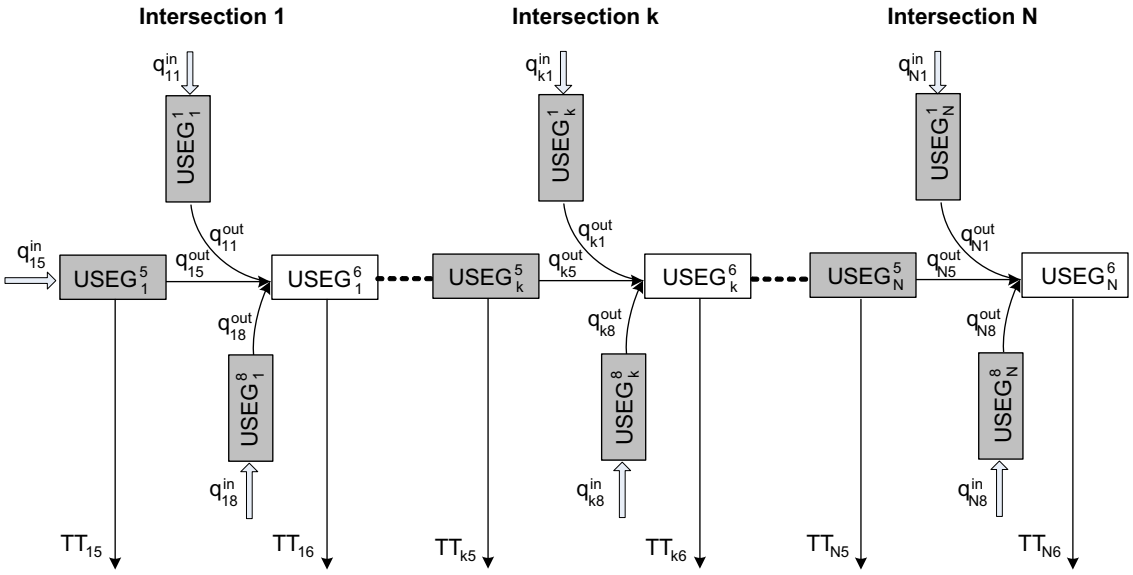


Figure 4.7: The predicted travel time from segment 15 to N6 is the sum of travel time on each segment.

4.6 Summary

This Chapter presents a neural network based traffic flow model to address the problem of travel time prediction on urban routes. A generic segment neural network model is proposed to predict travel time and outgoing flows on a urban segment. The outgoing flows will propagate from the segment into the connecting urban segments, and thus be fed to calculate travel times on the connecting urban segments. This procedure is repeated until the travel time of the final segment is calculated.

The remarkable features of this proposed model are: (a) using the state space neural network to model complex non-linear traffic processes at urban segment level; and (b) concatenating each urban segment model with traffic principles to predict travel times at urban route level. Correspondingly, two benefits can be obtained: (a) learning the mechanism of urban traffic processes directly from measured data, liberating from building sophisticated physical models; (b) fast and easy to implement in practice.

The proposed model will be evaluated in a simulated environment (Chapter 5) and a real world (Chapter 6).

Chapter 5

Model Testing on a Simulated Urban Route

In order to fully test the accuracy of the model formulated in Chapter 4 and techniques employed in this work, we first choose to use synthetic data obtained from a microscopic traffic simulation tool, VISSIM (PTV AG 2003), and then apply the model in a real-time environment (Chapter 6). The main advantages of using synthetic data are: (1) the flexibility of generating alternative scenarios (e.g. free-flow, intermediate, congested conditions) to be tested, which would otherwise be too expensive and time-consuming to be obtained from field test; (2) the provision of relatively clean and free-of-error data.

Three typical traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions) have been generated to test this proposed model. Based on the model developed in Chapter 4, this Chapter also addresses the issues of sensitivity and robustness. With the investigation on a simulated environment, it gives a clue for real time application (Chapter 6). The remainder of this Chapter will show the results of the test with synthetic data.

5.1 Simulation Scenario Description

Figure 5.1 shows an urban route in VISSIM that resembles a 2.05 kilometer urban arterial, Kruithuisweg, in the south part of Delft, the Netherlands. Since the synthetic data are only used to evaluate the performance of this proposed approach, we code this urban route with the same geometry as in the real world, ignoring the pedestrian lanes and tram lines. The simulated urban route has three signalized intersections (two four-leg intersections and one 'T' type intersection).

Single loop detectors are installed along this urban route and the segments crossing this urban route (shown in Figure 5.1), measuring flows. Two cameras are installed at the start and end of the urban route, measuring individual travel times.

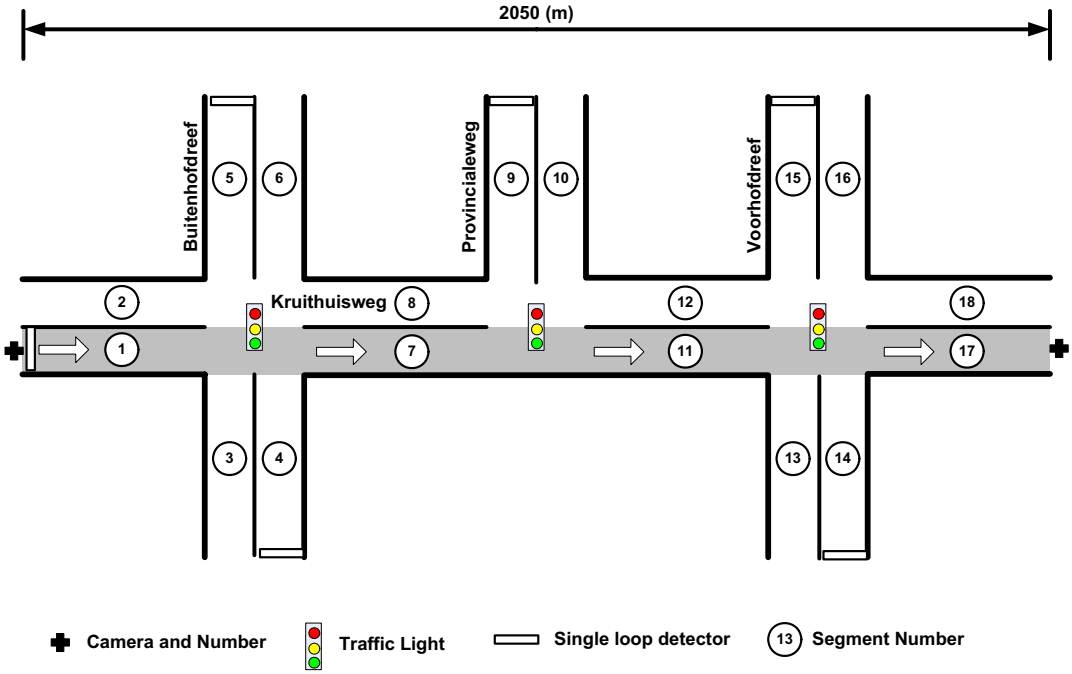


Figure 5.1: An urban street was simulated in microscopic traffic simulation tool VISSIM. This urban street resembles the Kruithuisweg provincial road in Delft, the Netherlands.

5.1.1 Signal Control Design

For an urban route, except the cycle length and signal timing of each phase at each intersection, the offset (the time difference between the start of the green phases on two adjacent intersections) is also very important for designing a control strategy.

Traffic control in the simulation is fixed time control. Two distinct designs of offsets were used in this study: green wave and red wave. The green wave pertains to the best design of the offset which makes a platoon that arrives at upstream intersections gets green right away at downstream intersections. The detailed calculation algorithm for green wave can be found in (Zuylen 2002). Red wave pertains to a contrary design which provides the platoon red phase when it arrives at downstream intersections.

Figure 5.2 shows the histograms of the ratio of red-wave travel times to green-wave travel times under free flow and congested conditions. The left plot shows that the histogram is slightly shifted to the right of the center point (ratio equals to 1). This indicates that a significant number of vehicles with red wave control experience longer travel times than those with green wave control. However, in congested conditions, the ratios distribute approximately symmetric around 1. This illustrates that in congested conditions vehicles experience long travel times no matter with green wave or red wave controls. Since congested conditions are more interesting for our study, we will use green wave for the following analysis.

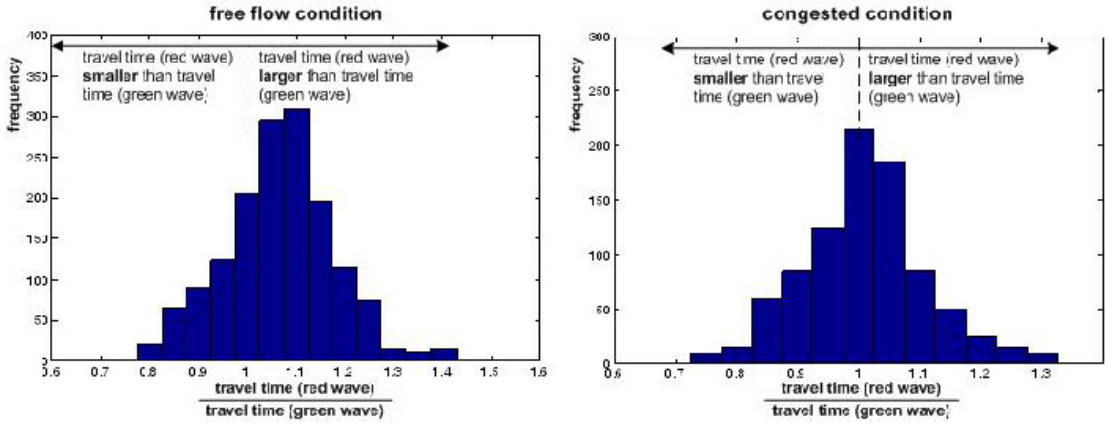


Figure 5.2: Histograms of the ratio of travel times (red wave) to travel times (green wave).

5.1.2 Input and Output Data

We have not calibrated the simulation with empirical data, because (1) with only single loop detector data, it is difficult and time-consuming to calibrate a microscopic simulation model, which involves in detailed driver behaviors (e.g. car following and lane changing); This is due to the microscopic simulation model has many more degrees of freedom than those can be detected by available macroscopic loop detectors. and (2) it is not necessary for the purpose of the feasibility test on the proposed model in the simulated environment.

To generate different scenarios for the model test, we set up several simulations by choosing different patterns of traffic demand. Since no sophisticated OD matrices were used, the turning fractions were used in the simulations to assign vehicles into the urban route. In this study, the turning fractions of each branch at each intersection were fixed as 10% left-turning, 10% right-turning and 80% throughput.

The outputs generated from simulations are individual travel times.

5.1.3 General Simulation Settings

The simulations were conducted over three hours, which pertain to a morning peak period from 7:00 to 10:00AM. Three traffic demand patterns were created to resemble three basic traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions). Fixed time control was used in these simulations. A green wave offset was implemented. Thus, we have three basic scenarios. For each scenario, we executed ten simulation runs with different random seeds. In total, there are 30 data sets of inputs (flows at detectors) and outputs (mean travel times for vehicles departing in each aggregated time period).

Note that training USEG was conducted at urban segment level, while testing the performance of travel time prediction was carried out at urban route level. Among those data sets, 18-segment sets were used for training and 12 sets of entire routes were used for testing.

Traffic Demand Pattern

Vehicles enter into the urban route through boundary segments (no vehicle generated from internal segments). For each boundary segment, a similar profile of traffic demand, starting with low flows (7:00-8:00), increasing to higher flows (8:00-9:00) and then decreasing back to lower flows (9:00-10:00). Three distinct traffic demand patterns have been set up: slightly saturated, moderately saturated and seriously oversaturated conditions. Traffic demand is assigned as a stepwise average pattern with time interval of 30 minutes. The actual flows released by VISSIM during the simulation, however, do vary from this average form due to the random probability functions used by VISSIM to release vehicles into the urban route.

Pattern 1: slightly saturated condition

Table 5.1 shows the assigned traffic flows for each boundary segment (segment 1, 4, 5, 9, 14 and 15 shown in Figure 5.1). Each boundary segment has a similar traffic demand pattern, which slightly exceeds the saturation flow during the period from 8:00 to 9:00. For the rest of two hours, the traffic flows are less than the saturation flow, which can be considered as free flow conditions. This slightly saturated flow causes queues to occur during a short time period. As a result, this yields a small peak with the travel times of approximately 200 seconds (shown in Figure 5.3).

Table 5.1: Time-varying traffic flow for all boundary segments (veh/h). Slightly high traffic flow occurs during period of 8:00 to 9:00, which yields slightly saturated conditions.

Time	segment number					
	1	4	5	9	14	15
7:00-7:30	200	50	50	50	50	50
7:30-8:00	400	300	400	400	250	300
8:00-8:30	800	700	750	750	550	650
8:30-9:00	800	700	750	750	650	650
9:00-9:30	400	600	500	400	250	300
9:30-10:00	200	200	200	200	50	50

Table 5.2: Time-varying traffic flow for all boundary segments (veh/h). Modestly high traffic flow occurs during period of 8:00 to 9:00, which yields modestly saturated conditions.

Time	segment number					
	1	4	5	9	14	15
7:00-7:30	200	50	50	50	50	50
7:30-8:00	400	200	350	400	250	300
8:00-8:30	800	800	650	800	450	500
8:30-9:00	1000	600	900	800	450	500
9:00-9:30	500	200	350	400	250	300
9:30-10:00	200	50	50	50	50	50

Table 5.3: Time-varying traffic flow for all boundary segments (veh/h). Extremely high traffic flow occurs during period of 8:00 to 9:00, which yields seriously oversaturated conditions.

Time	segment number					
	1	4	5	9	14	15
7:00-7:30	200	50	50	50	50	50
7:30-8:00	500	300	400	400	250	300
8:00-8:30	1100	950	950	850	650	600
8:30-9:00	1100	950	950	850	650	600
9:00-9:30	600	400	500	600	250	300
9:30-10:00	200	200	200	200	50	50

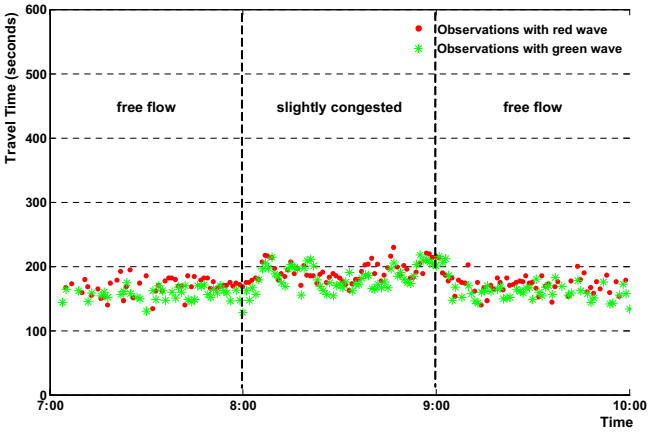


Figure 5.3: Travel time observations measured from VISSIM simulation on slightly saturated traffic demand. The figure shows one-minute aggregated travel time observations for different signal controls (green wave and red wave), respectively.

Pattern 2: moderately saturated condition

Compared with Patter 1, Patter 2 increases the traffic flows from 8:00 to 9:00 (see in Table 5.2. Under the moderately saturated condition, long queues occur along this route, which forces vehicles wait two or even more cycles before they can pass the intersections. Two peaks of travel times can be identified in Figure 5.4. This is due to the traffic flows still increase though the traffic flows of segment 4 decrease from 8:30 to 9:00. This illustrates that the traffic flows from crossing segments do have influence on the main urban route. One of the peaks has the highest travel time reaching 300 seconds (shown in Figure 5.4), which is two times the mean free flow travel time.

Pattern 3: seriously oversaturated condition

The third pattern has a relatively high traffic flow from 8:00 to 9:00. The queues along this route build up quickly and spill back to upstream intersections. Overflow queues appear at all the three intersections. Vehicles do not have opportunities to pass the intersections within one cycle time, which causes long delays for the whole trip. As a consequence, serious congestions on the main route occur during this period. Due to the serious block-

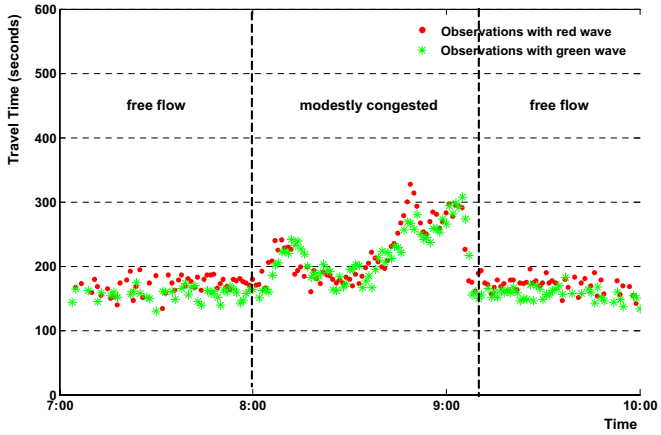


Figure 5.4: Travel time observations measured from VISSIM simulation on modestly saturated traffic demand. The figure shows one-minute aggregated travel time observations for different signal controls (green wave and red wave), respectively.

age, travel times increase to approximately 450 seconds, which is three times of the mean free flow travel time.

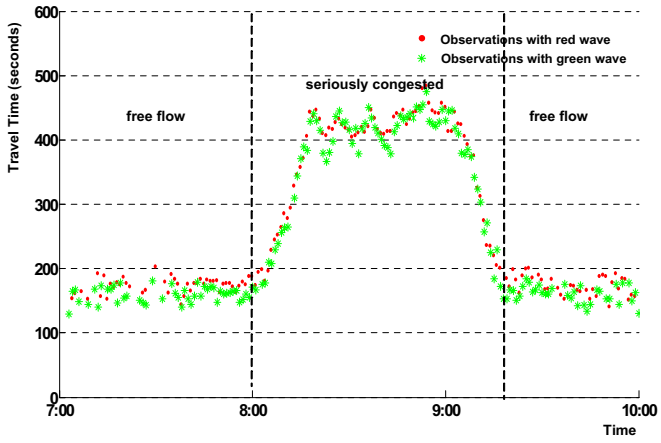


Figure 5.5: Travel time observations measured from VISSIM simulation on seriously saturated traffic demand. The figure shows one-minute aggregated travel time observations for different signal controls (green wave and red wave), respectively.

Output pattern

For each traffic scenario, we executed ten 3-hour simulation runs with different random seeds. Different random seeds yields different travel times per simulation run even if the simulation setting remains same. Thus, there are 30 data sets for three scenarios. Figure 5.6 gives an example of the results generated under the condition of the seriously saturated traffic demand pattern and green wave signal control strategy. It can be seen that during the peak period (from 8:40 to 9:10) the variability of travel times is larger than

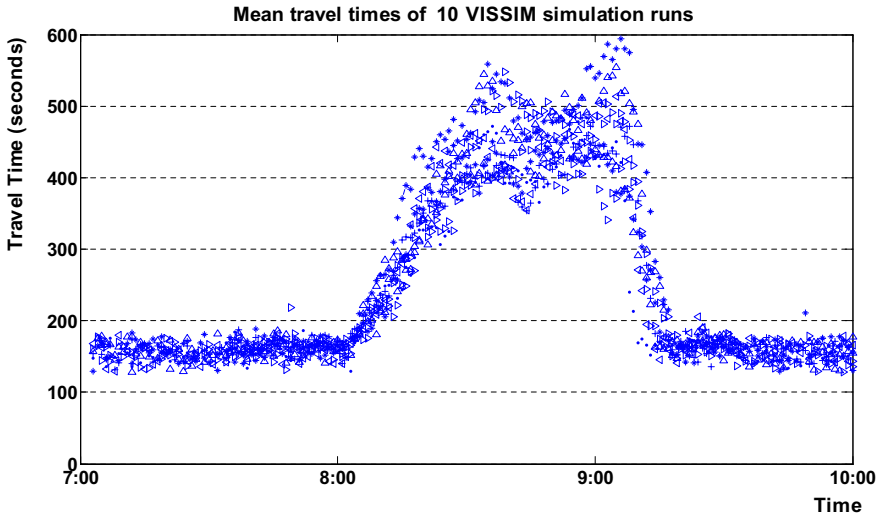


Figure 5.6: Mean travel times for 10 VISSIM simulation runs generated with seriously saturated traffic demand and green wave signal control. Each line represent one simulation result.

those during off peak periods. The worse case (the largest variability) is approximately 250 seconds at 9:06.

5.2 Results

To better present the results, we categorized them into three subsections. In the first subsection, we present the results of a baseline model, which is used for comparison. The ensuing two subsections correspond to the two phases of the model application: training the USEG and assessing the UROU. Note that the training process is done at segment level, while the assessment is done at route level.

5.2.1 Results of the Baseline Model for Comparison

Table 5.4: Predictive performance of the baseline model.

	MARE(%)	MRE(%)	SRE(%)
baseline model	20.4	4.7	17.9

As stated in Chapter 3, very limited amount of researches have been conducted on urban route travel time prediction. In the literature, few models have been proposed for this specific topic. Therefore, this dissertation only use a simple baseline model, which is widely used in practice (Beijing, Rotterdam), to compare the performance of the proposed model. The baseline model simply uses measured arrival travel times TT^a as the predicted departure travel time, which can be expressed as:

$$TT(p+1) = \frac{1}{n}\sum_{i=1}^{n} TT_i^a(p) \tag{5.1}$$

where n denotes the number of vehicles arriving at the end of the trip during time interval p.

Table 5.4 shows the predictive performance in terms of the mean relative error (MRE), mean absolute relative error (MARE), standard deviation of relative error (SRE) (formulas are given in Appendix A). As expected, the baseline model produces biased travel time predictions (MRE of 4.7%) with a SRE of 17.9%, and a large MARE of 20.4%. The inherent problem of the baseline model is already outlined in Chapter 2. Roughly, arrival travel times are the results of shifting departure travel times with the absolute value of corresponding travel times. Thus, simply using measured arrival travel times as the predicted departure travel time will: (1) underestimate the travel time in congestion onset conditions (prediction errors are negative), and (2) overestimate the travel time in the conditions of dissolving congestions (prediction errors are positive).

Table 5.5: MARE of training USEG in terms of different learning epochs and the number of hidden neurons.

Number of	Learning epochs						
hidden neurons	20	40	60	80	100	150	200
USEG1							
2	22.1%	18.3%	17.7%	13.5%	11.4%	11.4%	11.4%
4	26.3%	21.3%	16.7%	12.3%	8.6%	7.2%	7.2%
6	29.5%	22.4%	17.5%	13.7%	7.1%	6.8%	6.8%
8	30.6%	24.9%	19.6%	12.8%	8.9%	7.7%	7.7%
10	32.4%	26.9%	18.3%	13.4%	9.4%	6.2%	6.2%
USEG7							
2	30.6%	22.6%	18.5%	16.7%	14.3%	14.2%	14.2%
4	27.2%	22.1%	16.9%	12.3%	7.1%	7.1%	7.1%
6	28.6%	23.1%	19.2%	14.7%	6.3%	6.2%	6.2%
8	28.2%	21.3%	17.5%	12.2%	6.2%	6.2%	6.2%
10	30.6%	25.8%	20.6%	14.9%	8.7%	6.7%	6.7%
USEG11							
2	26.2%	23.4%	20.9%	16.4%	15.3%	15.3%	15.3%
4	28.3%	22.9%	16.2%	11.3%	8.6%	7.5%	7.5%
6	31.7%	26.4%	20.5%	15.1%	9.7%	6.3%	6.3%
8	30.6%	27.5%	22.4%	16.5%	10.1%	8.4%	8.4%
10	34.9%	29.8%	22.4%	17.2%	12.5%	7.4%	7.4%

5.2.2 Sensitivity Analysis of Training USEG

The training process is trying to find optimal parameter setting(s) of the USEG. There are several important factors that influence the structure of the USEG, and hence influence the performance of the UROU. These factors include the number of the hidden neurons,

the transfer function, the initial weights, and the initial internal states. The former two determine the structure of the USEG and the later two initiate the start point for training the USEG. Recall that the sigmoid function is selected as the transfer function and initial internal states all equal zero. Therefore, the following will only explore the sensitivity of the USEG to variations in the number of hidden neurons and initial weights.

The number of hidden neurons

To investigate the sensitivity of the training procedure on the number of hidden neurons, six different structures of the USEGs were developed (with 2, 3, 4, 6, 8, and 10 hidden neurons). For each USEG, the weight initialization is optimized based on (Nguyen & Widrow 1990). Table 5.5 shows the training results in terms of different numbers of hidden neurons for segment 1, 7 and 11 (see Figure 5.1). As expected, errors decrease with the increase of learning epochs, and after 100 epochs errors decrease slowly or even keep constant.

After being trained, those USEGs with different structures have been evaluated based on test data. Those test data are different from the data used above for training.

Table 5.6: MARE of assessing USEGs with different number of hidden neurons on independent data.

	Number of hidden neurons					
	2	3	4	6	8	10
USEG1	13.4%	13.1%	10.2%	9.3%	15.2%	12.3%
USEG7	15.7%	11.3%	8.4%	7.5%	11.5%	12.9%
USEG11	16.2%	15.4%	9.5%	13.8%	8.2%	17.3%

In the training process, the USEGs with more hidden neurons give smaller errors than those with less hidden neurons. This illustrates that complex (more neurons) models have more powerful ability to fit data than simple ones. Table 5.6, however, shows that complex models also intend to over fit data, and then produce a poor generalization. For the three segments, the USEGs with 4 and 6 hidden neurons outperform others. Based on those considerations, we decide to choose the USEG with 4 hidden neurons in the following analysis. This is due to (1) the USEGs of segment 1, 7 and 11 with 2 hidden neurons result in 13.4%, 15.7% and 16.2% MARE respectively, which are worse than those with more hidden neurons; (2) there is no significant difference between the USEG with hidden neurons of 4 and 6; (3) the USEG with 4 hidden neurons outperforms those with 3, 8 and 10 hidden neurons.

Initial weights

Note that the above training process is based on one initial weight parameter setting. As stated before, training neural networks with different initial weights might result in different weight settings, though the outputs of the neural network are similar. In addition, the use of different training algorithms could also produce different weight settings. In

reality, it is time-consuming ,even impossible, to draw an entire distribution of weight solutions in the multi-dimension weight space.

Here, we tentatively investigate the possible weight solution, initiating weights randomly from a zero mean normal distribution with different variances. The variances in our study are chosen as 0.1, 0.5, 1, 5, 10, 100. For each distribution, we randomly choose 10 scenarios of the weight parameter settings. In total, 60 initial weight parameter settings were used to training the USEG of segment 1. We stopped training after a maximum of 200 epochs.

From Table 5.7, we observe that the initial weight settings derived from the small distribution variances (e.g. 0.1 and 0.5) all produce error indicator (MARE) smaller than 9%. Some initial weight settings generated from the large distribution variances (e.g. 10, 100) also result in a good performance. This indicates that optimal weight solutions are scattered in the multi-dimension weight space. We take a close look at the weight values. 48 weight settings, which generate MARE smaller than 9%, are selected from Table 5.7. Figure 5.7 shows the histogram of the individual weight values of all the initial 48 weights settings. Although some weights are out the scope of [-50 50], most weights approximately concentrate within [-10 10]. In particular, the highest frequency of the weights is around 0. We hypothesize this due to two reasons. First, the inputs and outputs have all been scaled to [0.1 0.9]. The experimental data related to some nonlinear processes can vary across a wide range. Hence, the smallest values approach zero, and the differences between training points become smaller after being scaled. We admit that those 60 initial weight parameter settings might not be sufficient to represent the entire distribution of weight solutions. But, this investigation at least gives a clue for the real time application (Chapter 6).

Table 5.7: training results (MARE) with different initial weight parameter settings.

variance	**ten scenarios of initial weights setting**									
	1	2	3	4	5	6	7	8	9	10
0.1	6.8	7.3	7.9	7.4	7.3	7.5	7.4	7.3	7.3	7.5
0.5	8.2	7.6	7.4	7.2	7.3	7.5	7.4	7.3	7.6	7.3
1	8.7	11.3	7.4	9.2	7.3	7.6	10.6	12.7	8.6	7.4
5	7.9	10.2	7.3	7.5	7.3	8.3	7.5	7.2	8.9	8.5
10	7.5	10.3	11.8	7.3	8.4	8.4	7.2	8.3	9.2	6.1
100	10.3	10.2	17.4	8.4	7.3	7.5	7.4	10.4	8.2	8.5

5.2.3 Predictive Performance of UROU

After being trained, the USEG can be regarded as a model, which can describe the traffic processes at segment level (assuming that the presented data represent the traffic processes correctly). The following subsection will present the overall performance of the UROU based on the well-trained USEGs. First, the performance of the UROU trained by batch training algorithm will be given. After batch training, the parameters of the proposed model are fixed. The results of different prediction time aheads will be presented. Then, the performance of an incremental training algorithm will be given to compare with those by the batch training algorithm.

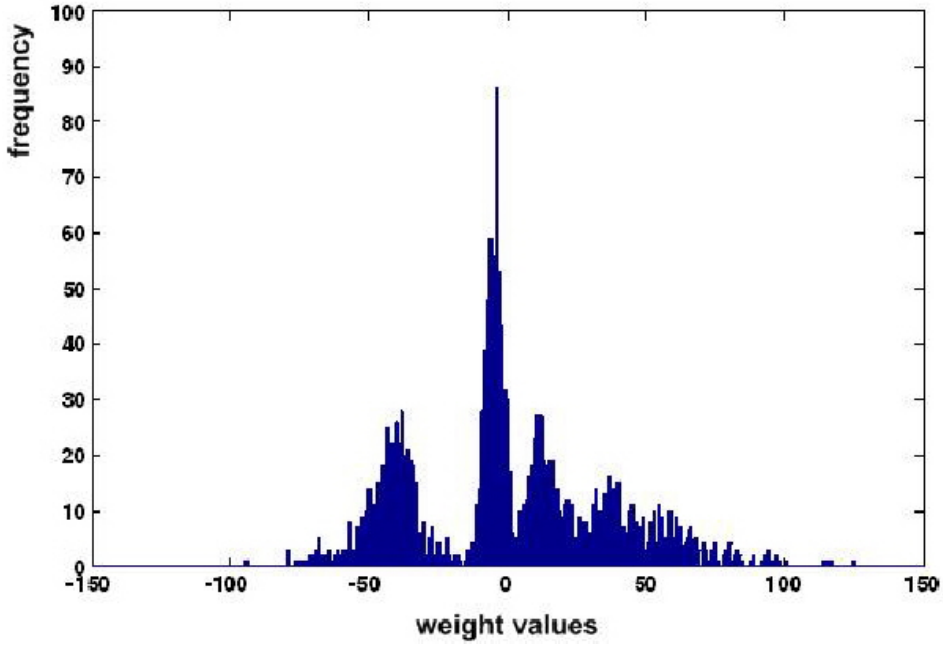


Figure 5.7: Histogram of weight values

Results of Batch Training

Table 5.8 shows that the proposed model performs better than the baseline model up to 30 minutes of prediction time ahead. The worse case is with MARE of 14.8%, MRE of 4.2% and SRE of 12.9, which are acceptable. In this simulation, we conclude that this proposed model is able to predict accurate prediction of travel times up to 30 minutes of prediction time ahead. The proposed model proves an accurate prediction of travel time in urban networks. However, in real world it is hard to priori predict the boundary traffic flows and turning fractions.

Table 5.8: Predictive performance of the proposed model with different prediction time ahead.

	Prediction Time Ahead (minutes)						
	1	5	10	15	20	25	30
MARE(%)	6.9	6.3	7.6	7.9	12.3	13.4	14.8
MRE(%)	2.7	1.9	3.2	3.6	3.7	3.8	4.2
SRE(%)	7.7	7.2	8.1	9.9	10.4	11.8	12.9

Results of Incremental Training

Table 5.9 shows the results of the proposed model with an incremental training algorithm. Recall that the principal difference between the incremental training algorithm and the batch training algorithm is to update weight parameters at each time step. Clearly, the proposed model with the incremental training algorithm performs significantly worse than the batch trained models. All performance indicators (MARE, MRE and SRE) are

approximately double in comparison to those resulted from the batch training. In the incremental training cases, the proposed model performs even worse than the baseline model when the prediction time ahead is equal or larger than 20 minutes. This illustrates that the weight parameters obtained from the incremental training algorithm are not appropriate for online travel time prediction. This is due to the fact that travel time observations are available only after vehicles finish their trips. Clearly, in the incremental training algorithm the proposed model is able to follow the arrival travel time curve not the departure travel time curve. Especially, it lags behind significantly during the congestion build-up and dissolving conditions. Moreover, it seems that the proposed model with the incremental training algorithm over fit the new observations and lead to a model which generalize poorly.

Table 5.9: Predictive performance of the proposed model with incremental training.

	Prediction Time Ahead (minutes)						
	1	5	10	15	20	25	30
MARE(%)	13.4	14.6	17.9	19.1	21.2	25.7	29.2
MRE(%)	4.5	4.8	6.1	6.7	8.2	8.1	10.3
SRE(%)	14.3	13.2	15.8	15.9	18.2	22.3	26.4

5.2.4 Robustness Analysis of UROU

In the previous sections we have tested the proposed model with 100% accurate simulated data. However, the input data in a real world situation, collected by a real time traffic monitoring system, will often have corrupted or missing values. Corrupted data refer to those systematically inaccurate data because of equipment measurement errors (e.g. miss count or over count). Recall that the definition of robustness of the proposed travel time prediction model is the ability to deal with the cases of being fed with corrupted and missing data (Chapter 1). More precisely, this dissertation specifies the robustness as the performance based on corrupted and missing data. The objective is that the proposed model is still able to produce reasonably accurate predictions fed with certain amount of corrupted and missing data.

Obviously, those incorrect data affect the offline training and online operation of the proposed model. In the training procedure, incorrect data steer weight parameters into a 'wrong' region in the weight space, and thus produce a 'wrong' model. Usually, the training procedure is conducted in an offline situation. In the offline case, we have enough time to replace missing data and to correct corrupted data. However, the online operation requires the calculation of travel times in a very short time. Thus, missing and corrupted input data are likely fed into the proposed model in the case of the online operation. The following sections are to investigate the online performance in case of missing and corrupted data given a well-trained model.

Missing data

Based on previous 12 testing data sets, we generated 28 missing data sets and 7 corrupted data sets. Considering all possible combinations of missing and corrupted data collected

by all the loop detectors leads to a very large amount of test data. Here we only consider the cases of detectors on segment 1, 4 and 9 (see Figure 5.1). We generated missing data according to 4 severity levels (5%, 10%, 15% and 20%). For example, the first level of 5% means that the testing data sets contain 5% missing data. The procedure of generating 5% missing data, for example, is conducted as follows:

step1 at each time step, produce a value from a uniform distribution on [0,1]

step2 if the value ≤ 0.05, then the data of this time step is labeled as missing data

step3 replace the correct data with a default value (in this dissertation 0.2, 0.5 and 0.8 are used for comparison)

In total, we have 84 synthetic missing data sets. Table 5.10 presents the MARE performance of the UROU on all the synthetic missing data sets. The third to fifth rows show that the performance of the UROU given that one detector provides missing data. No matter replacing the missing data with any values (0.2, 0.5 or 0.8), the performance deteriorates steadily as the percentage of missing data in the test sets increases. Remarkably, replacing the missing data with 0.5 yields the most encouraging performance. The results show that this simple strategy is able to ensure that the UROU performs quite well up to 10% missing data. We hypothesize this due to one reason. 0.5 represents an average scaled inputs. Choosing 0.5 allows replacing data not deviating far from correct ones, yielding a graceful deterioration.

The sixth to eighth rows show the results with two detectors providing missing data, and the last row shows the performance on three detectors providing missing data. As expected, more detectors encounter missing data, much worse performance was yielded. In the extreme worst condition when all three detectors providing missing data, more than 40% of MARE were yielded in the case of the missing percentage of 5%.

In a tentative conclusion, the UROU is able to produce robust prediction under the condition that one detector providing up to 10% missing data. Other serious cases of missing data should be considered carefully.

Table 5.10: MARE performance on missing data. The rows depict the location of detectors, providing missing data. The columns depict the combination of different severity levels and replaced values.

seg-	replaced with 0.2				replaced with 0.5				replaced with 0.8			
ment	5%	10%	15%	20%	5%	10%	15%	20%	5%	10%	15%	20%
1	26	38	54	62	11	20	34	48	22	36	52	78
4	24	42	58	71	10	18	36	51	21	37	48	86
9	28	37	49	65	10	19	34	46	26	41	56	81
1,4	37	53	72	84	25	44	57	73	31	48	68	85
1,9	34	47	65	86	27	49	62	81	29	52	72	86
4,9	36	52	69	78	31	43	59	78	32	55	74	91
1,4,9	48	62	81	97	42	68	86	98	41	58	72	96

Corrupted data

The procedure of generating corrupted data is similar to creating missing data except the third step. After determining the labeled time steps, we 'damage' correct data by adding corrupt ratios (-10%, -5%, 5%, and 10%). The negative and positive values are analogue to the cases of miss count or over count, respectively. Tables 5.11, 5.12 and 5.13 show the MARE indicator of the UROU for 10-minute time ahead travel time prediction on the corrupted data sets which represent one, two and three detectors encounter corrupted data, respectively. Only when the percentage of the corrupted data is equal or smaller than 10% and the corrupt ratio is -5% or 5%, the MAREs are less than 20%. As the percentage of the corrupted data increases or the corrupt ratio increases, the performance of the UROU deteriorates rapidly.

Table 5.11: Predictive performance of the proposed model on data sets containing corrupted data of segment 1.

percentage of	corrupt ratio			
corrupted data	-10%	-5%	5%	10%
5%	23	11	12	22
10%	28	19	18	32
15%	42	33	29	48
20%	68	46	56	74

Table 5.12: MARE performance of the proposed model on data sets containing corrupted data of segment 1 and 4.

percentage of	corrupt ratio			
corrupted data	-10%	-5%	5%	10%
5%	34	21	26	41
10%	58	38	45	63
15%	73	52	61	79
20%	84	68	72	89

Table 5.13: MARE performance of the proposed model on data sets containing corrupted data of segment 1,4 and 9.

percentage of	corrupt ratio			
corrupted data	-10%	-5%	5%	10%
5%	49	36	34	46
10%	62	48	45	75
15%	78	64	66	82
20%	96	73	82	98

5.3 Summary

This Chapter assessed the proposed model on 100% accurate simulated data. We choose to use synthetic data before the proposed model is tested in a real world environment because the simulation allows us to control the types of traffic situations and to test all kinds of designed scenarios. It provides a comprehensive test site to assess the concept design of the proposed model.

Three typical traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions) have been generated to test this proposed model. It is shown that the proposed model is able to provide accurate travel time predictions, and outperforms the baseline model.

In this Chapter, we have explored the sensitivity of the USEG to variations in the number of hidden neurons and initial weights. Based on simulation results, there is no significant difference between the USEG with 4 hidden neurons and those with hidden neurons of larger than 4. The small distribution variances of initial weights produce better training results. The two findings will be used in the real time application (Chapter 6).

Finally, we have tested the performance of the proposed model on synthetic missing and corrupted data sets. The results show that the proposed model still performs quite well under the condition of less than 10% missing and corrupted data. This proves that the proposed model satisfies our aim of providing robust and accurate travel time predictions. The next Chapter, we will apply the proposed model in a real world situation.

Chapter 6

Real-time Application

6.1 Introduction

In Chapter 4 and 5 a proposed model, the UROU, has been presented and evaluated in a simulated environment. The results supported the validation and applicability of this proposed model with 100% accurate simulated data. In this Chapter, we will put this model into practice and see whether it functions as well. Note that the major problem of the real time application is getting the right data. Thus, the data acquisition is one of the focuses of this Chapter.

First, two methods are presented to deal with the data pre-processing for loop detector data and travel time measurements. Then, the ensuing sections will address the variability of travel times with empirical data besides average travel time predictions.

The Regiolab Delft project provides an ideal test site for the real-time application. Regiolab Delft collects both travel times and volumes from various traffic data collection systems installed on a wide range of different roads (both urban routes and freeways) within the region of Delft (Zuylen & Muller 2002). The selection of appropriate routes from the urban road of Delft (shown in Figure 6.1) was based on two criteria: camera data and loop detector data were available for this route, and recurrent congestion occurs. After extensive analysis, the route with start point (256) and end point (257) was selected. This route was also used as a blueprint for the experiments based on synthetic data described in Chapter 5.

6.2 Description of the Test Site

The route, named “Kruithuisweg”, is a provincial road with characteristics of an urban arterial connecting two freeways, A4 and A13. Three license plate cameras with sequence numbers 256 (one-lane) and 257(left lane and right lane) are installed at two locations of the 2.05 km route. Those cameras are used to measure volumes and travel times (see Figure 6.1). Through GPRS (General Packet Radio Service), the first 5 license plate characters of all passing vehicles are sent to Regiolab Delft with a time stamp every 2 minutes. From the different time stamps of the same set of license plate characters

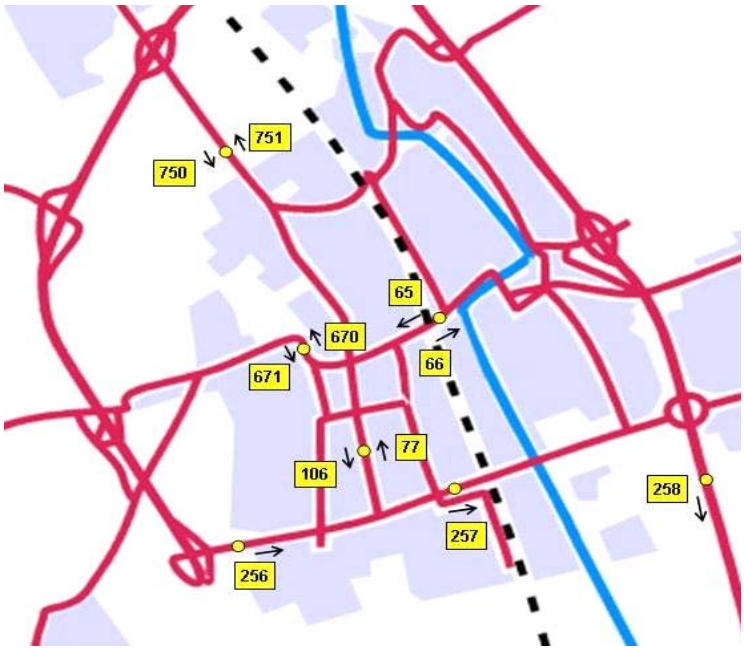


Figure 6.1: The camera locations covered by the Regiolab Delft. The little circle on the figure depict the camera location.

collected at the two locations, individual travel times from 256 to 257 are calculated. This route produces travel times of approximate 180 up to 1100 seconds in free flow and seriously congested conditions respectively.

Loop detectors provide one minute aggregated volumes. All controlled intersections along this route have vehicle actuated control programs. Every intersection is controlled separately without coordination. Traffic signal controllers provide signal timings (e.g. green time).

For our purpose of evaluating this proposed model under different traffic conditions, especially in the congested condition, morning peaks (between 7:00 and 10:00) data were used to train and evaluate this proposed model.

6.3 Model Description

According to the configuration of the single loop detectors installed along Kruithuisweg (shown in Figure 6.1), six segments were used to model this urban route. Figure 6.3 shows a schematic configuration. Since segment 4, 5 and 7 only have downstream loop detectors, the volumes collected by them are directly used as the boundary inputs into the downstream segments which are connected to them. For instance, the left-turning volumes from segment 4 and the right-turning volumes from segment 7 are fed as inputs into segment 2, as well as the left-turning volumes from segment 5 are fed into segment 3. Therefore, only three USEGs which represent segment 1, 2 and 3 are created and trained. For each segment, we choose the same topology of the USEG model. Based on the sensitivity analysis of the USEG model with simulation data (see Chapter 5), the

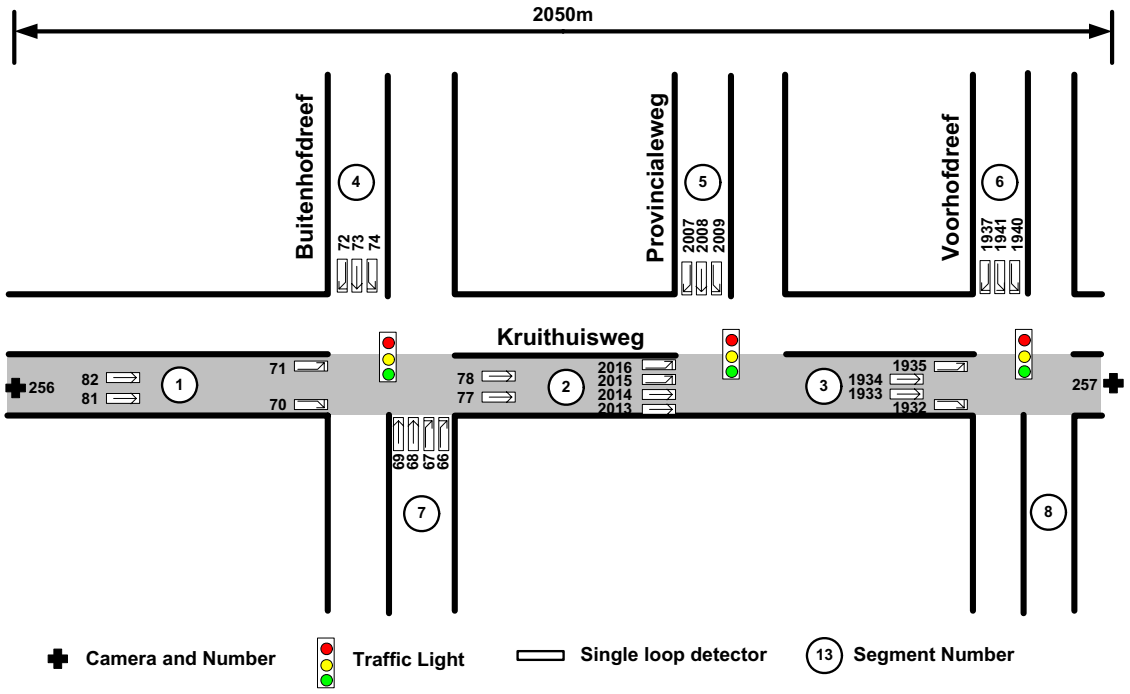


Figure 6.2: The layout of the loop detectors and license camera installed along the Kruithuisweg.

topology of the USEG model is defined with four hidden neurons. And, the sigmoid transfer function is used. A batch training algorithm has been selected for training those USEG models due to its better performance than the incremental training algorithm as shown in Chapter 5.

6.4 Data Preparation

In reality, the measured data often consist of missing or corrupted values, which are not like those in a simulation (100% accurate data). Thus, those raw data need to be checked and processed before being used. As stated before, there are three types of data used in this dissertation: volume, signal timing and travel time. We consider the signal timings (green time and cycle time) as accurate measurements. The following sections will address the processing of volume and travel time measurements.

6.4.1 Volumes Measured by Single Loop Detectors

In Regiolab Delft, volume measurements of each single loop detector are either with *null* values (no measurements available might because of malfunctioning) or *non-negative* integers. Data with null values are regarded as missing data. For this research, a data validation procedure is designed based on (Weijermars & Berkum 2006, Muller et al. 2005, Turner 2004).

First step: Preliminary Data Selection

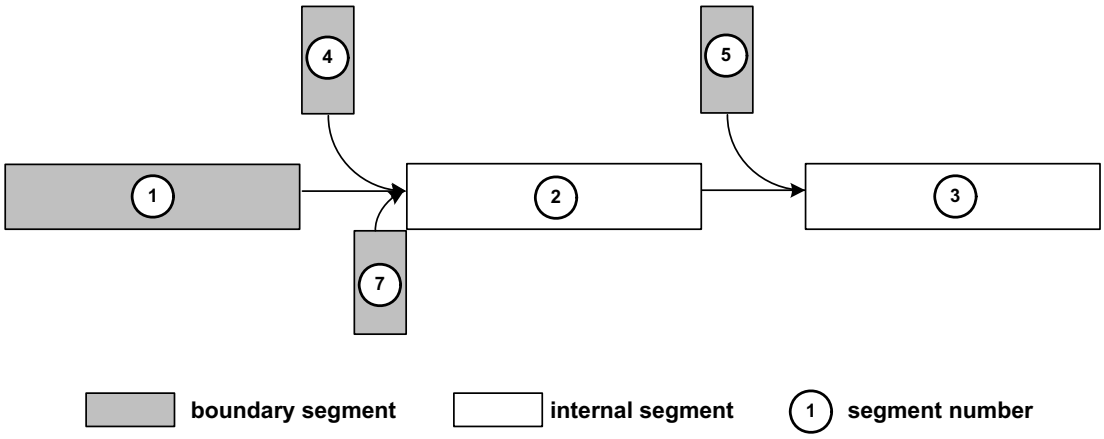


Figure 6.3: A schematic configuration of USTR with concatenating USEG for modelling travel time on Kruithuisweg.

- In the raw database, we found that malfunctioning occurred quite frequently not only for short periods (e.g. 1 minutes) but also for long time periods (e.g. an entire day in worse cases). If the consecutive time intervals of null values are larger than 20 minutes, these data were not selected. In addition, for preliminary selection, only those days that all the single loop detectors work well were chosen. After the preliminary data selection, only 82 days of 2004 are available.

Second step: Preliminary Data Completion

- In those 82-day data, we used the simple linear interpolation to fill in those time intervals of null values. Those calculated values will be verified whether they are reasonably correct with the following steps.

Third step: Individual Data Checking

- Traffic flows are bounded by the capacity of the measurement location and by the capacity of the upstream locations. For reasons of simplicity, a fixed upper bound was used that is the same under all circumstances (without considering other factors, like weather). However, the location of detectors has to be taken into account. For the single loop detectors installed upstream far from intersections, an appropriate upper limit of 30 veh/lane/minute was used. For the single loop detectors equipped close to intersections, the upper limit is determined by the saturation flow and green time. In this study, we choose an approximate estimation of 24 veh/lane/minute. Although the upper limits are chosen after an explorative analysis of the data and are realistic from a traffic theory point of view, they are somewhat arbitrary. Therefore, a sensitivity analysis is executed.

  Except the highest value of volumes, we also need to pay attention to the value of zero. Data of morning peak hour (7:00-10:00AM) are considered in this research, the volume of zero for multiple consecutive time intervals during the peak hours are suspicious and stamped with a flag for further verification.

Fourth step: Cross Data Checking

- The principle of flow conservation is widely accepted for checking volume data of one single loop detector with other detectors within the same intersection. For two locations between which traffic cannot 'disappear' and new traffic cannot be generated, the principle applies. This principle can be used to find out missing and over count of the traffic. The test site satisfies the requirement of a complete detector configuration. For instance, $I_{77} + I_{78} = I_{81} + I_{82} - I_{70} - I_{71} + I_{74} + I_{66} + I_{67}$ (I_k denotes the cumulative flow at detector k). In general, the principle however is difficult to apply for the urban networks because of the lack of dense detectors (discussed in Chapter 2).

Table 6.1: Flow conservation at the intersection at Buitenhofdreef and Kruithuisweg

	Flow conservation for hourly flows		
Flow conservation for daily flows	Met	Not Met	Total
Met	177 cases	33 cases	210(85%)
Not Met	24 cases	12 cases	36(15%)
Total	201(81.7%)	45(18.3%)	246 cases

Table 6.2: Flow conservation at the intersection at Provincialeweg and Kruithuisweg

	Flow conservation for hourly flows		
Flow conservation for daily flows	Met	Not Met	Total
Met	162 cases	14 cases	176(71.5%)
Not Met	36 cases	34 cases	70(28.5%)
Total	198(80.5%)	48(19.5%)	246 cases

Table 6.3: Flow conservation at the intersection at Voorhofdreef and Kruithuisweg

	Flow conservation for hourly flows		
Flow conservation for daily flows	Met	Not Met	Total
Met	156	9	165(67.1%)
Not Met	48	33	81(32.9%)
Total	204(82.9%)	42(17.1%)	246 cases

From Tables 6.1, 6.2 and 6.3, it can be seen that the principle of flow conservation is not met in almost 20% of the cases. Note that we have not distinguished the causes of one missing vehicle or hundreds vehicles. Of course, a small amount of missing vehicles might influence less than a large amount of missing vehicles. In the following sections, all those cases which do not meet the principle of flow conservation will not be used.

6.4.2 Travel Times Collected with License Plate Matching

Travel times can be measured with different technologies, such as distance measuring instrument, license plate matching, automatic vehicle identification, global positioning system, platoon matching, cellular phone tracking, etc (see in Appendix B). For a comprehensive overview of travel time data collection systems we refer to (Turner et al. 1998). In the Regiolab Delft project, automatic license number plate recognition (ANPR) systems were used. Three cameras record a picture of the license plate numbers of all passing vehicles (one land or multi-lanes) and then use a specific software to identify the license numbers. By matching the license numbers recorded at two locations, the travel times of vehicles between those two locations were calculated.

There are many causes (not exhaustive) which result in 'erroneous' travel times. Here, 'erroneous' travel times refers to the records that cannot represent the real traffic conditions. In the ensuing sections, we use the term of 'outlier' referring to the 'erroneous' travel time from statistical perspective.

Causes for Erroneous Travel Time Observations

- Misrecognition of License Plate Numbers

  The misrecognition of license plate numbers refers to yielding wrong license numbers due to misrecognizing the characters. For example, a license plates with the syntax of "46ATQS" might be recognized as "46ATOS" due to the letter "Q" is similar to the letter "O". Since image recognition, which is widely used technology for license plate matching, is sensitive to ambient conditions. For instance, adverse weather will lead to a high possibility of misreading license numbers.

- Mismatch with Partial License Numbers

  Under the consideration of privacy issues, only partial license plate numbers have been recorded in Regiolab Delft. The last four letters of the license plate numbers will be used for matching. This results in a possibility that two (unique) vehicles have a same recognized partial license number. For example, "17HRMG" and "24HRMG" could be regarded as one vehicle with the partial license number of "HRMG".

- Alternative Routes

  In urban networks, vehicles are able to change their directions at intersections, yielding several alternative routes between one pair of origin A and destination B. If only two cameras are installed at A and B, the travel times derived from license plate matching are hard to be classified into one specific route.

- Particular Vehicles

  Some particular vehicles are not restricted to normal traffic regulations so that their travel times should be filtered out. For example, emergency vehicles are able to travel at speeds higher than the speed-limit, sometime even travel in the 'wrong' direction, and travel through red-lights at intersections as well. Another example,

buses may pass intersections without delays due to priority signal settings, and experience no delays in congestion situations because they drive on reserved lanes. These particular vehicles realize 'erroneous' travel times which cannot represent the average traffic conditions.

- Non-driving Activities

  When vehicles have more trip purposes other than solely traversing from A to B, their travel times are considered as 'erroneous' travel times. For example, a parent passes by A, stops at school between A and B to pick up his/her children, then reaches B. Other activities, like going to a shop, filling gasoline at a gas station, also cause the travel times to include extra delays.

Outlier Detection of Travel Time Data

In real world, the consequence of ignoring outliers might be more serious than miscrecognizing valid data. Obviously, ignoring outliers results in training the model with incorrect data, which will deteriorate the performance of this model on new data. Misrecognizing valid data will filter out valid data, reducing the total number of correct data. Detecting outliers is the basis of producing a clean data set for training the model. The key objectives of an outlier detection algorithm are twofold: (1) the rate of ignored outliers (records identified as valid data while in fact they are outliers) is low, and (2) the misrecognized valid data rate (records identified as outliers while they are valid data) is low as well. In the literature, Percentile Test (PT) (Clark et al. 2002) and Deviation Test (DT) (Fowkes 1983, Clark et al. 2002) have been used for detecting the outliers of travel time observations (details see in Appendix E). To improve the accuracy of the detection of outliers, a generic procedure of outlier detection is proposed (Appendix E).

Table 6.4: Performance of each method to identify outliers in the matched ANPR data, on a provincial road Kruithuisweg, the Netherlands

	Proposed Method	**PT(Clark)**	**DT(Fowkes)**	**DT(Clark)**
2004-Nov-17				
No. of Observations	893	893	893	893
No. of Misrecognized	10	80	6	8
No. of Ignored	4	57	32	31
2004-Nov-18				
No. of Observations	939	939	939	939
No. of Misrecognized	7	86	11	10
No. of Ignored	6	55	33	32
2004-Nov-19				
No. of Observations	964	964	964	964
No. of Misrecognized	5	87	9	7
No. of Ignored	12	58	34	34

PT(Clark): Percentile test method proposed by Clark
DT(Fowkes): Deviation test method proposed by Fowkes
DT(Clark): Deviation test metnod proposed by Clark

Comparison of different outlier detection approaches

On average, the percentages of the outliers of each day vary from 8% to 15%. Most outliers occurred during the day time, while a few outliers were identified during the evening. Three-day results are shown in Table 6.4. Obviously, the Percentile Test is not able to correctly identify real outliers because the percentage of the total errors (ignored outliers and misrecognized valid data) to total observations are 15.34%, 15.02% and 15.04% for these three days, respectively. The two Deviation Test approaches have better performance with the percentages of approximately 4.5% of the total errors to total observations. There is no significant difference between the DT(Fowkes) and DT(Clark). The new proposed method outperforms other methods with a very low error of 1.57%, 1.36% and 1.78% for these three days, respectively.

The Relation between ignored outliers and misrecognized valid data

Figure **??** shows the two kinds of errors, misrecognized valid data N_s and ignored real outliers N_g, with respect to different parameter settings. As described in Appendix E, the three parameters, T_w, σ_{thr} and N_{thr}, influence the performance of the outlier detection. When the time window increases from 5 to10 minutes, the number of misrecognized valid data decreases slowly while the number of ignored outliers maintain constant. As the time window continues increase, N_g decreases very slowly and N_s increases sharply. This evidence supports the statement that a large time window probably increases the possibility of ignored real outliers. This is due to the large time window 'hide' the likely significant changes in itself. Under this condition, the real travel time outliers will be easily ignored. The number of misrecognized valid data changes very slowly. For this case, the proper time window is 10 minutes.

We can also observe from Figure 6.4 (middle and bottom) that obviously N_g increases and N_s decreases as both the critical standard deviation and critical count increase. Qualitatively, a larger critical standard deviation yields more ignored cases and less misrecognized cases. Similarly, a larger critical count generates less misrecognized cases, while N_g increases very slowly as critical count increases. The crossing points of the curve N_g and N_s, the lowest value of the total errors, are the optimal values for the parameter settings. For this case, $\sigma_{thr} = 108$ and $N_{thr} = 3$ are the proper parameter settings.

Influence of outlier detection on training USEG

As shown above, the simple outlier detection algorithms (PT and DT) are easy to be implemented in practice but the accuracy is poor (depending on the selection of the percentile and the critical distance). The proposed method gives a good performance of outlier detection.

In real world, the influence of the ignored outliers on training models might be more serious than the miscrecognized valid data. Tentatively, the ignored outliers cause the models are steered to 'wrong' parameter settings due to incorrect data. Misrecognized valid data only reduces the number of valid data. We further assess those outlier detection algorithms by answering the follow question:

Problem 4 *What is the influence of outlier detection on the training of USEG?*

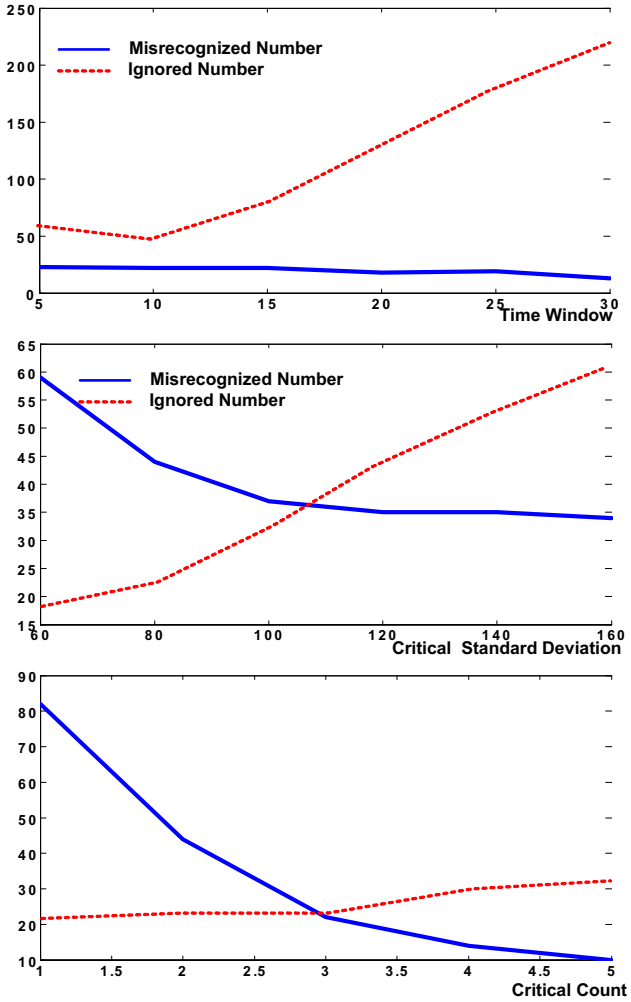


Figure 6.4: The performance of the new proposed algorithm with respect to different parameter settings (time window, critical standard deviation and critical count). The number of misrecognized valid data and ignored outliers are used as measures.

We used the four outlier detection algorithms separately again in order to filter out all the outliers. That is, the number of ignored outliers is zero. To achieve this goal, we just simply tuned the parameters. Table 6.5 shows the results of the number of misrecognized valid data when all the outliers are filtered out. Obviously, the number of misrecognized valid data increases with filtering out more outliers.

To fill in the empty gaps where the outliers are filtered out and probably the valid data are also filtered out by mistake, we used a simple method as described in Appendix E. Those four new data sets were used separately to train a USEG model. For the comparison purpose, the raw data without filtering out the outliers was also used to train a USEG model. We started to train each USEG with the same number of hidden neurons of 4, the same initial weights and internal states. We stopped the training after a maximum of 200 epochs.

An interesting result is that the training runs with the data sets, which are generated from the proposed method, DT(Fowkes) and DT(Clark), produce approximately the same sum

squared error (SSE) (2.699482, 2.701544, 2.702486), and moreover, produce approximately the same weight parameters. The data set processed with PT(Clark) produced much higher SSE (2.975624). As we expected, the training with raw data yields the highest SSE of 3.819282. Note that the training data are scaled into the interval $0.1, 0.9$. Obviously, the training produces the largest SSE when a lot of outliers exist in the training data. When all the outliers are filtered out, the USEG produces a lower SSE. A close look at the SSE without outliers indicates that the influence of the misrecognized valid data on training is not significant when a few valid data (the proposed method compared to DT(Fowkes) and DT(Clark)) are filtered out. In other words, there is no apparent improvement of the SSE with the proposed method compared with the simple algorithm DT(Fowkes) and DT(Clark). However, the SSE increases much more when the number of misrecognized valid data increases (PT(Clark)). A close look at the training data shows that more valid data are filtered out, and thus more empty gaps have to be filled in. The PT(Clark) produces long gaps, which are more than three or even five time intervals. The simple method we used to fill in the gap apparently cannot correctly deal with this case. But, the simple method works quite well when only short gaps in the training data have to be filled.

6.4.3 Subdivision of The Data Sets

After data checking and completion as shown above, we further divided these data into three data sets, data set A , data set B, and data set C. Data set A contains 50 morning peaks' (from 7:00 to 10:00AM) single loop detector data, signal timing data and segment travel times. Those data in data set A are raw data without data pre-processing. Data set B was extracted from data set A with data pre-processing, which contains 36 morning peaks' data. For each segment, each record reflects a departure travel time interval and contains input/output traffic flows, green time and segment travel time. Data set C, containing 26 morning peaks' data, is similar to data set B. The difference is that no segment travel times but only measured route travel times are available (for the purpose of testing). Note that only data set A contains missing/corrupted data. The use of data set A is to demonstrate the influence of missing/corrupted data on the performance of the proposed model.

6.4.4 Training

An inherent problem of training neural networks is that they might be sensitive to initial weights and may get stuck in a local minima of the error surface. One way to alleviate this problem and to increase the likelihood of obtaining near-optimum local minima is to train several neural networks, having a same structure, with a random set of initial weights, and choose the one with the lowest error. However, in a practical application, we actually don't know which neural network performs best for future unseen data. There may be many parameter sets within a model structure that are equally acceptable. Consequently, instead of choosing one best single USEG model, we may make predictions based on an ensemble of neural networks trained for the same purpose. In this dissertation, the idea of ensemble prediction is adopted, and the simple average ensemble method is used. For each USEG model, we trained it ten times so as to get 10 neural networks, and choose the 5 best ones according to their training performances. With the selected neural networks,

we got 5 outputs for each segment travel time prediction. Then we took a simple average of the 5 outputs to be the final output.

Table 6.5: Performance of each method to identify outliers in the matched ANPR data, on a provincial road Kruithuisweg, the Netherlands

	Proposed Method	PT(Clark)	DT(Fowkes)	DT(Clark)
2004-Nov-17				
No. of Misrecognized	24	168	47	52
2004-Nov-18				
No. of Misrecognized	28	159	54	54
2004-Nov-19				
No. of Misrecognized	25	166	48	52

PT(Clark): Percentile test method proposed by Clark
DT(Fowkes):Deviation test method proposed by Fowkes
DT(Clark): Deviation test method proposed by Clark

6.5 Results

To better show the performance of the proposed model, we first present the results with the baseline model. In the following sections, we will present not only the average travel time prediction, but also the travel time variability.

6.5.1 Performance of the baseline model

Table 6.6 shows the mean, mean absolute and standard deviation of the relative error for the baseline model. As expected, the baseline model produces biased travel time predictions (MRE of 4.5%) with a SRE of 25.8%, and a large MARE of 18.6%. The inherent problem of the baseline model is already outlined in Chapter 2. In general, arrival travel times are the results of shifting departure travel times with the absolute values of corresponding travel times. Thus, simply using the measured arrival travel time as the predicted departure travel time will: (1) underestimate travel times in congestion onset conditions (prediction errors are negative), and (2) overestimate travel times in congestion dissolve conditions (prediction errors are positive). Figure 6.5 illustrates that this simple model produces negative errors when congestions start and positive errors when congestions dissolve. In free flow conditions, the prediction errors fluctuate around zero within the range of approximate [-100 100].

Table 6.6: Predictive performance of the baseline model on training data set B and test data set C in 2004.

	MARE(%)	MRE(%)	SRE(%)
baseline model	18.6	4.5	25.8

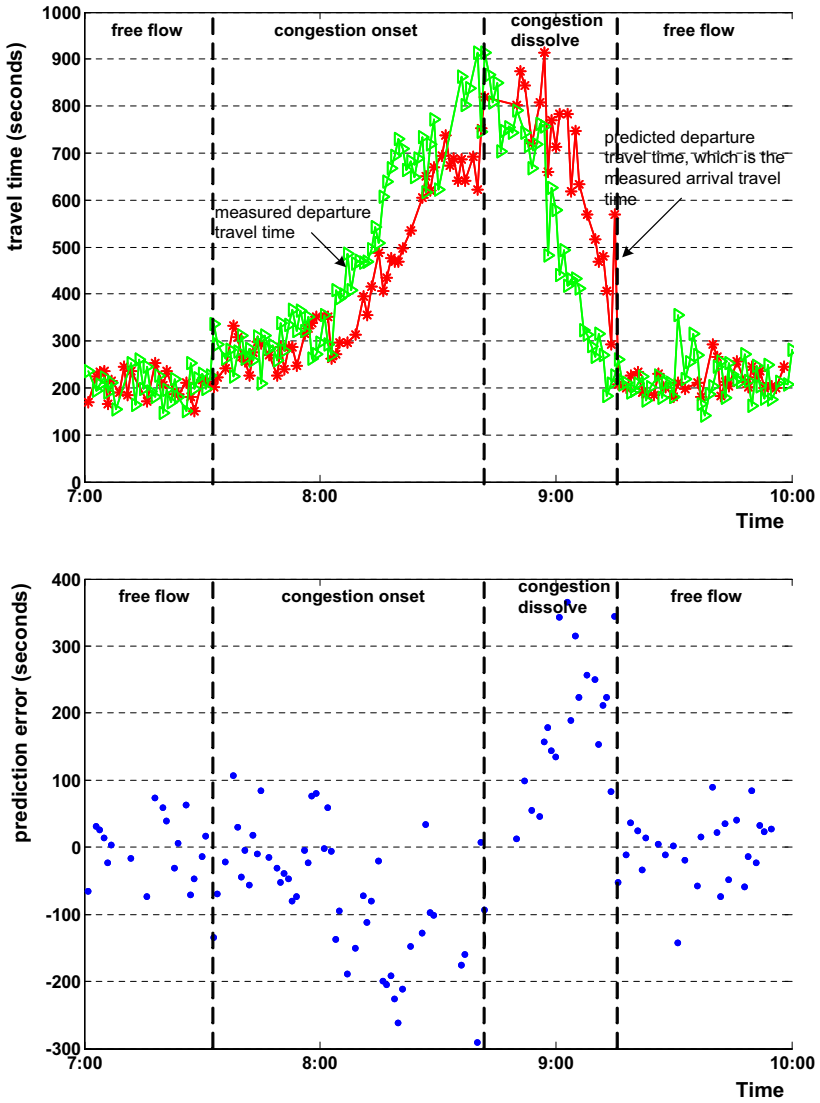


Figure 6.5: Predictive performance of the baseline model on the morning peak of 27 Octorber, 2004.

6.5.2 Performance of the proposed model

For the real-time application, three strategies could influence the performance of the proposed model: (1) the integration of traffic flow prediction, (2) the use of pre-processed data, and (3) the use of different training algorithms.

As shown in Chapter 4, the proposed model is based on the prediction of traffic flows on the boundary of the study area, which will propagate from an upstream segment to a downstream segment, and then result in travel times. Since the profile of the traffic flow is pre-determined in the simulation environment, the assessment of this strategy has not been done in Chapter 5. In the following section, the influence of the traffic flow prediction will be presented.

It is no doubt that the model trained with good data will give a good performance. Here,

we will show how the influence of different training data sets on different prediction time aheads. The prediction time aheads used in the following section are 1, 5, 10, 15, 20 and 25 minutes.

In Chapter 5, the simulation results show that the model trained with the batch training algorithm outperforms the model trained with the incremental training algorithm. The following section will show the results of the two training algorithms with empirical data.

The influence of traffic flow prediction

Tables 6.7 and 6.8 shows the performance of the model with and without traffic flow prediction, respectively. In general, the MARE, MRE and SRE decrease as the length of prediction time ahead increase no matter with or without traffic flow prediction. As expected, this proves that it is much harder for travel time prediction with a large prediction time ahead than a small prediction time ahead. The MARE of the model with traffic flow prediction are less than 18.6% (baseline model) up to the prediction time ahead of 10 minutes. Nonetheless, the MARE of the cases without traffic flow prediction are larger than 18.6% after the prediction time ahead of 5 minutes. Similarly, the MRE of the model with traffic flow prediction are less than 4.5% (baseline model) up to prediction time ahead of 10 minutes, while the MRE of prediction without traffic flow prediction are larger than 4.5% in all cases. With respect to the SRE, the results of both with and without traffic flow prediction are similar to the cases of MARE. Roughly speaking, the model with traffic flow prediction outperforms the baseline model up to a prediction time ahead of 10 minutes, and the model without traffic flow prediction outperforms the baseline model up to prediction time ahead of 5 minutes.

Table 6.7: Predictive performance of the proposed model with traffic flow prediction on training data set B and test data set C in 2004.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	11.6	9.8	13.4	21.6	32.5	38.3
MRE(%)	3.4	2.8	3.8	8.3	10.6	14.8
SRE(%)	13.2	11.4	14.5	25.4	33.6	36.8

Table 6.8: Predictive performance of the proposed model without traffic flow prediction on training data set B and test data set C in 2004.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	16.5	18.4	29.5	34.8	46.6	49.7
MRE(%)	8.9	14.2	19.3	27.5	33.9	35.3
SRE(%)	19.7	23.5	28.6	31.6	37.2	39.4

Table 6.9 shows the results of traffic flow prediction (with the algorithm described in Chapter 4). Obviously, the performance of traffic flow prediction decreases as the prediction time ahead increases. The minimal MARE, MRE and SRE are 18.5%, 5.6% and 11.2% for the case of 1-minute prediction ahead.

Table 6.9: Predictive performance of traffic flow prediction on test data set C in 2004.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	18.5	19.2	21.9	25.3	36.7	41.4
MRE(%)	5.6	7.4	7.9	19.1	23.3	37.9
SRE(%)	11.2	13.7	19.2	21.3	26.7	29.3

The influence of different training data sets

Tables 6.10 and 6.11 list the performance evaluation results of the 1 to 25 minutes ahead predictions with the training data sets A and B respectively. From the two tables, we can observe that the performance deteriorates with increasing the prediction time aheads. Compared with Table 6.6, both Tables 6.10 and 6.11 show that the proposed model performs better than the baseline model up to the 10-minute prediction time ahead. When the prediction time ahead increases to 15, 20 and 25 minutes, the performances of the proposed model are even worse than the baseline model. The model performs best for the 5-minute prediction time ahead. The appropriate prediction time ahead might be different at different application locations. In our case, we conclude that this proposed model is able to produce accurate predictions of travel times up to 10 minutes ahead. The main reason is due to the inherent drawback of boundary traffic flow prediction and the simple assumption of turning fraction. In urban networks, traffic flow probably vary in short time period (e.g. 10 minutes), causing the input of the model to be unpredictable. It is clear that when the boundary traffic flow and turning fraction change significantly within the prediction time ahead, e.g. congestion onset and congestion dissolve, the simple assumption can not be hold. Therefore, the performance of this proposed model is largely influenced by the performance of the prediction of boundary traffic flows and turning fractions.

Table 6.10: Predictive performance of the proposed model on training data set A and test data set C in 2004.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	19.6	25.4	31.3	28.7	37.6	42.5
MRE(%)	6.4	7.2	16.7	19.2	22.1	28.6
SRE(%)	24.7	27.3	26.6	28.3	30.4	38.4

Table 6.11: Predictive performance of the proposed model on training data set B and test data set C in 2004.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	11.6	9.8	13.4	21.6	32.5	38.3
MRE(%)	3.4	2.8	3.8	8.3	10.6	14.8
SRE(%)	13.2	11.4	14.5	25.4	33.6	36.8

The results of different training algorithms

Tables 6.12 and 6.13 show the performance of the proposed model trained by the batch training algorithm and incremental training algorithm, respectively. The overall performance is similar to the performance in the simulation environment (Chapter 5). Obviously, in the batch training algorithm the proposed model performs better than a model trained with the incremental training algorithm. From the two tables, it can be seen that no matter whether it is trained with the batch or incremental training algorithm, the performance of the proposed model in real time application is worse than it in the simulation environment. Even the pre-processed data sets are used, the minimal MARE, MRE and SRE are 9.8%, 2.8% and 11.4%, which are larger than those (6.9%, 2.7% and 7.7%) in the simulation environment. Different from the situation in the simulation environment (where the proposed model performs better than the baseline model up to 25 minutes of prediction time ahead), the proposed model trained with the batch training algorithm outperforms than the baseline model up to 10 minutes of prediction time ahead.

Table 6.12: Predictive performance of the proposed model trained with batch training algorithm.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	11.6	9.8	13.4	21.6	32.5	38.3
MRE(%)	3.4	2.8	3.8	8.3	10.6	14.8
SRE(%)	13.2	11.4	14.5	25.4	33.6	36.8

Table 6.13: Predictive performance of the proposed model trained with incremental training.

	Prediction Time Ahead (minutes)					
	1	5	10	15	20	25
MARE(%)	19.2	21.5	25.7	33.2	38.5	47.3
MRE(%)	6.9	8.5	12.7	17.5	20.8	29.4
SRE(%)	20.4	26.9	30.1	38.7	42.8	48.1

6.5.3 Travel Time Prediction with Variability Estimation

As described in Chapter 2, empirical travel times vary due to many influencing factors. That is, the variability of travel times is the result of the combination of the stochastic variation of those influencing factors (e.g. temporal effects, composition of vehicles, population characteristics, weather, road works, traffic control and management) (Van Lint 2004, Viti 2006). Thus, the travel times have a certain distribution, such that there tends to be a minimum travel time, but it is possible to have a very large travel time. From travelers' perspective, a decrease in travel time variability reduces the uncertainty in decision-making about departure time and route choice as well as the anxiety and stress caused by such uncertainty (Sun et al. 2003).

Very little empirical research has been undertaken into this field. Although a lot of studies have been conducted into travel time variability, the lack of sufficient data has prohibited researchers from properly investigating the different temporal scales of travel time

variability. Even today only roads fitted with ANPR cameras are likely to have sufficient travel time records for a thorough study of TTV to be undertaken. As we will show below, the travel time variability is obtained with the analysis of historical travel time measurements.

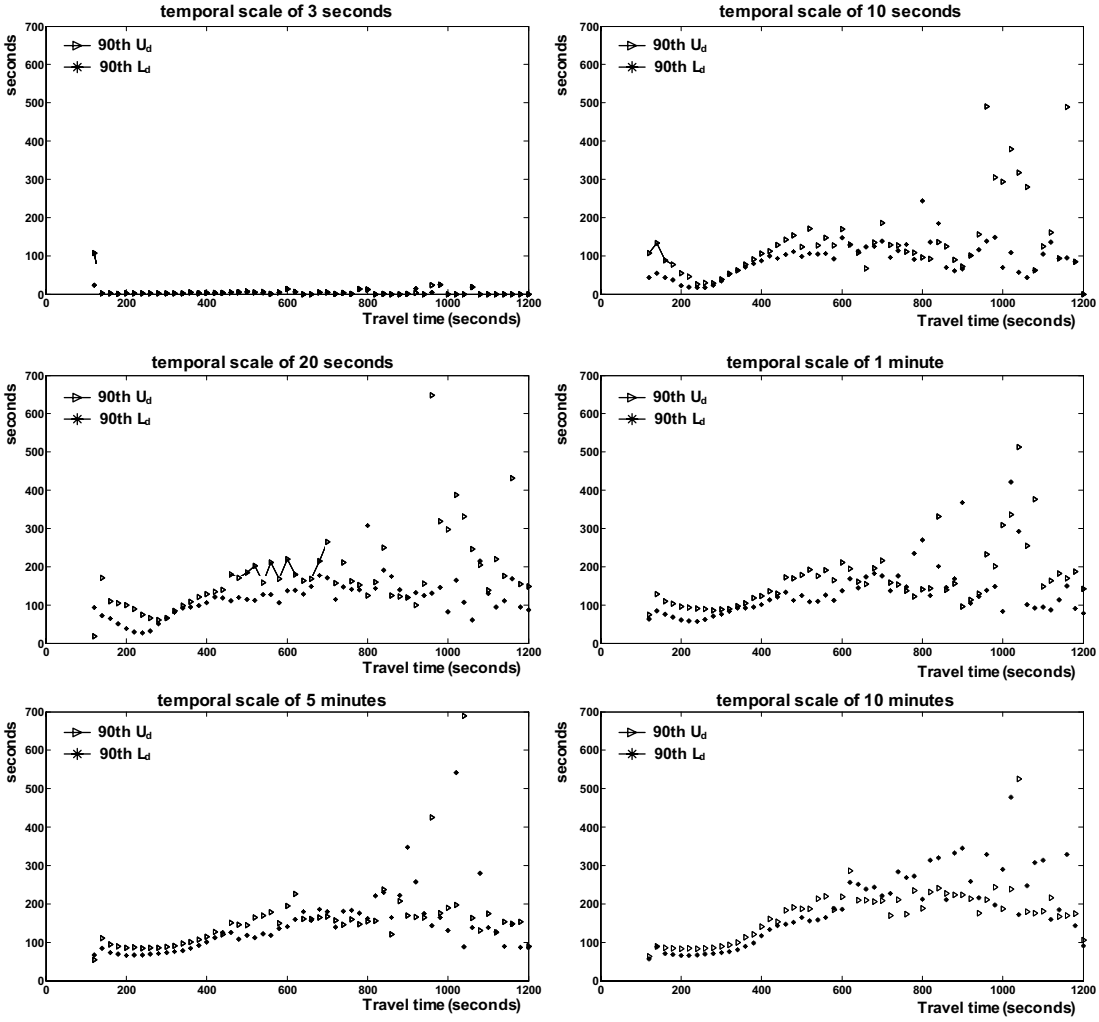


Figure 6.6: The variability of travel time in terms of different temporal scales.

In this dissertation, we restrict our definition of travel time variability in a statistical sense by means of prediction intervals.

Definition 13 *Prediction intervals are the minimal and maximal possible travel time around the prediction, that is ,the most plausible range of travel time a vehicle is likely to experience, given the prediction value.*

The prediction intervals account for vehicles starting within a couple of minutes of one another experiencing different travel times. The minimal and maximal possible travel times, TTV_x and TTV_d, are obtained as follows:

(1) Prepare departure travel time measurements in a time series, [departure time t_i, travel time tt_i].

(2) Determine the temporal scale TS.

(3) Start with $i = 1$, find out the 10th and 90th percentile travel time measurements departing within $t_i - TS$, $t_i + TS$. The upper and lower deviation of this travel time measurement are $U_d = 50th - 10th$ and $L_d = 90th - 50th$.

(4) Continue step 3 with $i = i + 1$ until the end of travel time series.

(5) Group travel time measurements in a stepwise pattern. Let stepwise step be $ST = 20$ sec. For example, start from the minimal travel time measurement TT_x, find out all travel time measurements in TT_x, $TT_x + ST$. Thus, find out the 90th percentile of U_d and L_d obtained from step 3.

(6) Finally, the 90th percentile values of U_d and L_d obtained from step 5 are the TTV_x and TTV_d, respectively.

For example, the data set A was used. Six different temporal scale were used: 3 second, 10 seconds, 20 second, 1 minute, 5 minutes and 10 minutes.

Figure 6.6 illustrates the percentile values of the variability of travel times in terms of different temporal scales. Note that the horizontal axis is travel time and the vertical axis is the variability of travel time. The magnitudes of the 90th percentiles increase with the increase of the length of the temporal scales. In addition, the 90th percentiles of upper deviation fluctuate more strongly than the lower deviation. This implies that travel times are instable in congested conditions (here high travel time value) than in free-flow congestions. Moreover, the upper deviation curve is always above the lower deviation curve. This demonstrates that the distribution of travel times is certainly not symmetric. Instead, the distributions for different temporal scales are skewed to the right.

To have a close look of the upper and lower deviations separately, we plot them in Figure 6.6. It can be seen that the 90th percentile of the upper and lower deviations of travel times remain very low and constant as the temporal scale is 3 seconds. For the cases of the temporal scale larger than 10 seconds, the 90th percentile of the variability of travel times increase significantly with the value of travel times. This shows that urban travel times are very variable. For this 2 km urban route, vehicles departed even within time difference of 10 seconds, they still had a large possibility of experiencing a large variability of travel times. The patterns of 90th percentiles of the upper and lower deviations for 1, 5 and 10 minutes are quite similar. Especially, the curves of travel times from 200 to 400 approximately overlap each other.

With the statistical upper and lower deviation, the prediction intervals can be placed around the mean prediction. The Prediction Interval Coverage Percentage (PI_CP) denotes the number of observations N_{in} that fall in the interval:

$$PI\_CP = \frac{N_{in}}{N_{tot}} \tag{6.1}$$

where N_{tot} denotes the total number of observations.

In practice, it is necessary to determine how the sample size affects the derivation of the upper and lower deviations. First, the data set B was used to estimate the prediction intervals. Six different sample sizes were used: 1, 5, 10, 15, 20, 25 days. For each sample size,

we randomly select data ten times from data set B. Then, the obtained prediction intervals were tested on data set C. Table 6.14 shows the performance of prediction intervals in terms of different sample sizes. The average PI_CP increases from 33.43% to 91.28% as sample size increases from 1 to 25 days.

Table 6.14: Prediction Interval Coverage Percentage index in terms of different sample sizes.

Sample	**Test Runs**									
Size	1	2	3	4	5	6	7	8	9	10
1 day	27.4	19.3	21.5	43.2	28.6	56.7	35.2	25.9	42.7	33.8
5 days	56.7	38.5	49.3	29.6	48.5	39.3	55.2	36.3	41.6	22.9
10 days	67.4	72.5	58.3	49.2	56.1	78.2	52.1	58.7	72.9	77.4
15 days	69.4	63.1	68.5	74.6	69.3	81.5	73.6	77.4	79.3	78.2
20 days	89.3	81.5	79.4	86.4	93.5	95.6	92.1	85.3	87.6	92.3
25 days	93.5	91.2	82.3	86.8	96.2	94.3	83.6	97.2	96.4	91.3

6.6 Comparison of Simulation and Real-time Results

The most prominent differences between the simulated travel time and real travel time are: (1) serious congested travel times in the simulated environments are smaller than in the real applications, although the free-flow travel times are similar; (2) both in simulated and real time environments, the travel time variability increases as traffic conditions change from free flow to congestion. For free flow conditions, the magnitude of the travel time variability in simulated enviroments is smaller than in real time applications. For congested conditions, the magnitude of the travel time variability in the real time environments is significantly larger than in the simulations.

Overall, the performance of the proposed model is, although still acceptable, significantly worse on real data than on simulated data. Recall that in the simulation the performance of the proposed model with the prediction ahead of 30 minutes is better than the baseline model. However, in the real application, when the prediction ahead is larger than 10 minutes, the performance of the proposed model becomes worse than the baseline model. It shows that applying the proposed model in a real-time environment is more complex than in a simulated situation. This is due to the fact that the simulation is controlled by the users while too many unknown influencing factors involved in the real-time application are undetected and out of control. In addition, the detection equipments cannot provide 100% accurate measurements, as in simulation environments.

Moreover, the arrival pattern and the turning fractions in reality are less predictable than in the simulation.

6.7 Summary

This Chapter presents an application of the proposed model in a real time environment. The practical application, not like simulation (100% correct data), requires an extensive

strategy to deal with the quality of actual observations. There are two different types of actual observations in our case study: volumes and travel times. We defined four steps to tackle the problem of missing and corrupted volume data.

Since license plate matching is widely implemented in urban areas, more details have been discussed, especially about the causes that yield erroneous travel time observations. The main problem of measured travel times is the outliers caused by mismatching and misrecognizing license plate numbers. We compared the performance of two simple outlier detection algorithms and one new derived algorithm. One of the simple algorithms is able to filter out outliers at the cost of simultaneously discarding valid data. The new derived algorithm shows best performance.

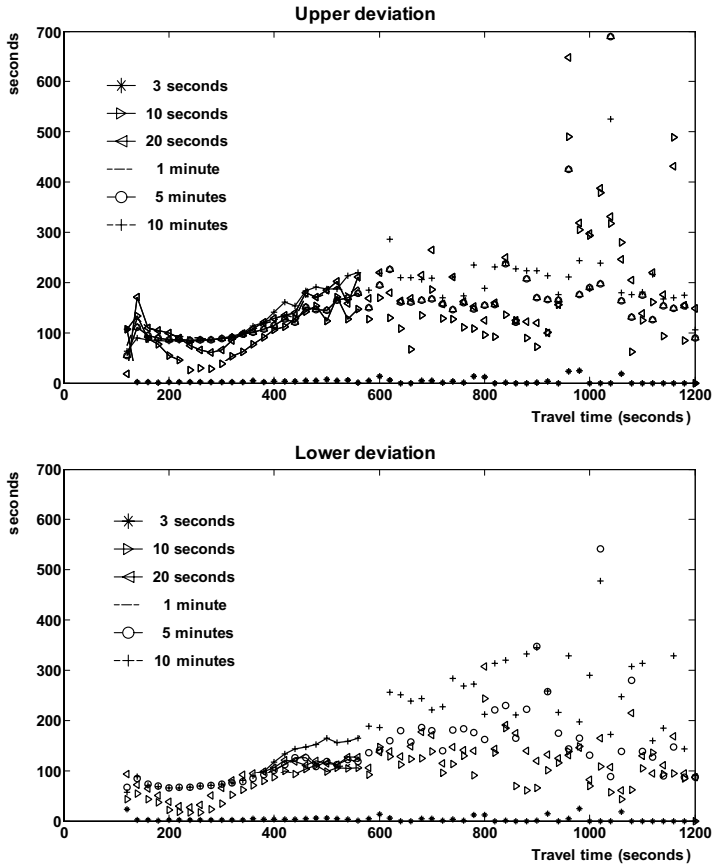


Figure 6.7: The upper and lower deviations with respect to different time scales.

For the real-time application, three strategies could influence the performance of the proposed model: (1) the integration of traffic flow prediction, (2) the use of pre-processed data, and (3) the application of different training algorithms. Those strategies were evaluated with respect to predictive performance. The results show that the predictive performance outperforms the baseline model when the prediction time ahead increases up to 10 minutes. By considering travel time variability, we introduce the application of prediction intervals into travel time prediction.

Chapter 7

Conclusions and Recommendations for Further Research

This closing Chapter summarizes the main conclusions of the work done in this dissertation. Thereafter, some further research directions are highlighted.

7.1 Conclusions

In this dissertation we clarified the components of urban travel times and presented a model for urban travel time prediction. Recall that the objective of this dissertation is to develop a methodology that can provide robust and accurate travel time predictions for urban networks. Correspondingly, the conclusions of this dissertation are summarized into the following sections: problem description, model development and model evaluation (a simulated environment and a real-time application).

7.1.1 Problem Analysis

People driving on urban networks will be influenced by many factors, e.g. other drivers, traffic signals, pedestrians, transit priority, parking, et. Due to the interaction among those factors, urban travel time prediction is a highly complex non-linear spatiotemporal problem. From our literature review it appears there are very few models available for urban travel time prediction. We argue that this is due to the shortage of data for calibrating and validating models. Presently, more and more cities, like Beijing (China), Stockholm (Sweden) and Delft (The Netherlands), however, have installed license plate cameras for monitoring large-scale urban networks. In addition, GPS data collected by many taxies are easily obtained. Those direct travel time measurements can be supplemented for calibrating and validating urban travel time prediction models.

7.1.2 Model Development

The proposed model in this dissertation can be seen as a sophisticated travel time or delay formula with adjustable parameters (which have no direct physical meaning). The

approach is a hybrid of data driven and model-based approaches. The benefit is that it is dynamic and (at least conceptually) simple. A generic segment-based model has been presented, and it also can be used for urban routes.

1. A generic travel time prediction model based on the State Space Neural Network (SSNN), so called USEG, has been developed for modeling traffic flows on a urban segment. Fed with inflow volumes and traffic signal timings, the USEG produces both outflow volumes and travel time predictions for the urban segment.

    - The SSNN model enables the previous states to be temporally memorized in itself. The feedback (memory) mechanism in the SSNN allows the inputs to be fed at consecutive time instants sequentially. We argue that dynamic neural networks are better suited for the travel time prediction task since they induce the time dynamics directly from the data, in contrast to static neural networks in which the time dynamics are constraint by a fixed input time window.
    - Compared with augmenting all the spatially separated inputs in a single input layer, modeling an urban signalized route with separate models for each urban segment significantly reduces the number of weight parameters which need to be calibrated.
    - With the availability of real time measured travel times by license plate cameras and GPS equipments, it is possible to use incremental training algorithm. The model parameters are updated after the presentation of each measured travel time.

2. The outgoing traffic flows leaving from the upstream urban segment, calculated with the USEG, can be used as the inputs for the connecting downstream urban segment USEG. By concatenating all the USEGs which are comprised of an urban route of interest, the UROU is developed to propagate traffic flows through the route of interest. The ability of propagating traffic flows enable the UROU to predict travel times for any long route of interest. Three key issues of concatenating the USEG are:

    - Boundary segments require the prediction of incoming traffic flows. We argue that the real time incoming traffic flows are proportional to the historical profile.
    - Unlike the boundary segments, the internal segments obtain the incoming traffic flows from the outgoing traffic flows of the upstream (connecting) segments. We argue that the dynamics of the turning fractions are assumed as random-walk processes. Taking into account the turning movements, the total incoming traffic flows of a internal segment are the sum of the left-turning, throughput and right turning traffic flows.

7.1.3 Model Evaluation in Simulation Environments

In order to fully test the model formulated in Chapter 4 and techniques employed in this work, we first choose to use synthetic data obtained from a microscopic traffic simulation

tool, VISSIM (PTV AG 2003). With the investigation on a simulated environment, it gives a clue for the real time application (Chapter 6). Three typical traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions) have been generated to test this proposed model.

1. There are several important factors that influence the architecture of the USEG, and hence influence the performance of the UROU. These factors include the number of the hidden neurons, transfer function, initial weights and initial internal states. The former two determine the structure of the USEG and the later two initiate the start point of the USEG training. In this dissertation, we have explored the sensitivity of USEG to variations in the number of hidden neurons and initial weights.

    - By considering accuracy and generalization, we found that the USEG with 4 hidden neurons was appropriate.
    - We tentatively investigate the possible weight solution, initiating weights randomly from a zero mean normal distribution with different variances. The results show that the highest frequency of weight is around 0.

2. The trained USEG was regarded as a generic model, which can describe the traffic processes at segment level. Based on the well-trained USEG, the UROU was tested in terms of the batch training and incremental training algorithms.

    - With the batch training algorithm, the UROU is able to produce accurate travel time predictions up to 30 minutes of the prediction time ahead. The performance of the UROU is better than the baseline model.
    - In the incremental training fashion the UROU is able to follow the arrival travel time curve, but it lags behind the departure travel time curve, particularly when the congestion builds up and dissolves. It illustrates that the incremental training algorithm over fitts the new observations and leads to a model which generalizes poorly. The UROU performs even worse than the baseline model when the prediction time ahead is equal or larger than 20 minutes.
    - In a conclusion, the availability of real time measured travel times is not a recipe to improve the performance of urban travel time prediction.

3. This dissertation specifies the robustness as the performance of the UROU fed with corrupted and missing data. We found that the UROU is still able to produce reasonable predictions with a certain amount of corrupted and missing data.

    - Replacing missing data with 0.5 yields more encouraging performance than replacing with 0.2 and 0.8. This simple strategy is able to ensure the UROU to perform quite well under the condition that one detector provides up to 10% missing data. In the extremely worst condition when all three detectors provide missing data, more than 40% of MARE was yielded in the case that even the missing percentage is equal to 5%.
    - As the percentage of the corrupted data increases or the corruption ratio increases, the performance of the UROU rapidly deteriorates. Only when the percentage of corrupted data is equal or smaller than 10% and the corruption ratio is -5% or 5%, the MAREs are less than 20%.

7.1.4 Real-time Applications

After the UROU was evaluated in a simulated environment, we put this model into practice. The major difference between simulations and real-time applications is that the former have 100% correct data. Therefore, the real-time application requires additional efforts to deal with the quality of the actual measurements.

1. We have presented different methods of dealing with corrupted and missing data collected by single loop detectors and license plate cameras, respectively.

   - According to the principle of flow conservation almost 20% of the single loop detector data are missing. In this dissertation, a data cleaning procedure for the single loop detector data has been designed based on (Weijermars & Berkum 2006, Muller et al. 2005, Turner 2004).
   - A procedure to detect outliers of travel time measurements has been proposed. The new procedure outperforms than three existing methods (e.g. Percentile Test (Clark et al. 2002) and Deviation Test (Fowkes 1983, Clark et al. 2002)).

2. For the real-time application, three strategies influence the performance of the proposed model: (1) the integration of traffic flow prediction, (2) the use of preprocessed data, and (3) the use of different training algorithms.

   - The model with traffic flow prediction outperforms the baseline model up to the prediction time ahead of 10 minutes, while the model without traffic flow prediction outperforms the baseline model up to the prediction time ahead of 5 minutes.
   - The performance of this proposed model is largely influenced by the performance of the prediction of the boundary traffic flows and turning fractions. In urban networks, the traffic flow varies in a short time period (e.g. 10 minutes), causing the input of the model to be unpredictable.
   - The proposed model with the batch training method outperforms it with the incremental training method.

3. For this test bed of a 2km urban street, vehicles that depart even within time difference of 10 seconds, they still have a large possibility of experiencing a large variability of travel times.

4. Overall, the performance of the proposed model is, although still acceptable, significantly worse on real data than on simulated data. In the simulation the performance of the proposed model with the prediction ahead of 30 minutes is better than the baseline model. However, in real application, when the prediction ahead is larger than 10 minutes, the performance of the proposed model becomes even worse than the baseline model.

7.2 Recommendations for further research

In this final section we present research directions that are triggered naturally by the research done in this dissertation. Those research directions are grouped into urban travel time prediction, the model improvements and others for further successful urban travel time prediction.

7.2.1 Urban Travel Time Prediction

1. The current formulation of the proposed model predicts travel times based on single loop detector data, travel time measurements calculated with license plate matching, and traffic signal timings. However, new technologies such as mobile phones and GPS, and vehicle re-identification (that use existing inductive loops) are providing new avenues for collecting real-time traffic information. As a result a possible direction for future research would be the incorporation of these data sources into the proposed model.

2. In this dissertation, we only consider two influencing factors of urban travel times, that is, traffic flow and traffic signal. Clearly, other factors (e.g. vehicle composition, pedestrians, weather, public transport, etc.) also influence urban travel times. An interesting future research direction could be the extension of the proposed model to account for the effect of those factors. As long as those variables are measured, they can be easily integrated in the proposed model. The selection of a suitable training method will be conducted in terms of different factors. For instance, the incremental training method might be better for the factor of weather.

3. As stated above, many influencing factors affect the urban travel time variability. An empirical research into the effects of those factors on the urban travel time variability would be important. More research is required to understand and quantify the effects. Moreover, the relative change in the urban travel time variability between two factors also should be investigated.

7.2.2 Model Improvements

1. In this dissertation, the UROU has been integrated with the prediction of boundary traffic flows and turning fractions. This means that the performance of the proposed model depends on the prediction of the boundary traffic flows and turning fractions. We only use simple methods to predict those two important factors. However, an effort should be put on this subject for further research.

2. We have tested the proposed model in simulation and real-world environments. To establish the generality of the conclusions of the UROU, more test beds with different geometry designs should be selected. For instance, the availability of data from Beijing city will provide another evaluation scenario for further research.

7.2.3 Other Research Directions

1. Travel time prediction is a core subject of ATIS. The predicted travel times will be provided to drivers and traffic managers via on-board systems, variable message signs, mobile phones, etc. Some theoretical and empirical knowledge and practical guidelines should be undertaken by the further research directions, including

    - what are the potential effects and traveler responses in terms of individual and collective traffic operations?
    - how to distribute unbalanced travel times for different travelers? For instance, providing different travel times for travelers even having same origin and destination in order to avoiding to attract all travelers to one route which results in congestion.

Bibliography

Adler,J.L. and V.J. Blue (1998), 'Towards the design of intelligent traveller information systems', Transaction Research Part C 6(3),157-172.

Akcelik, R. and M. Besley (2001), Acceleration and deceleration models, in 'The 23rd Conference of Austalian Institutes of Transport Research'.

Anderson, J. and M.G.H. Bell(1997), Travel time estimation in urban road networks, in 'IEEE Conference on Intelligent Transaction System', ITSC.

Antoniou, C. (2004), On-Line Calibration for Dynamic Traffic Assignment, Ph.D. thesis, Massachusetts Institute Technology.

Bajwa, S.U.L., E. Chung and M. Kumahara (2003(a)), A travel time prediction method based on pattern matching technique, in 21st ARRB and 11th REAAA Conference', Cairns, Australia.

Bajwa, S.U.L., E. Chung and M. Kuwahara (2003(b)), Sensitivity analysis of short term travel time prediction model's parameters, in '10th World Congress on Intelligent Transport Systems', Madrid, Spain.

Ben-Akiva, M., A. De Palma and I. Kaysi (1991), 'Dynamic network models and driver information systems', Transaction Research Part A 25(5),251-266.

Bierlaire, M. and F. Crittin (2004) 'An efficient algorithm for real-time estimation and prediction of dynamic OD tables', Operations Research Archive (52(1)),116-127.

Bishop, C.M. (2005), Neural Networks for Pattern Recognition, Oxford University Press.

Bovy, P.H.L and E. Stern (1990), Route Choice Wayfinding in Transport Networks, Kluwer academic publishers, Boston.

Bovy, P.H.L and R. Thijs (2000), Estimators of Travel time for Road Networks: New Developments, Evaluation Results, and Applications, Delft University Press.

Chen, Z., T. Feng and Z. Houkes (2000), Incorporating a priori knowledge into initialized weights for neural classifier, in 'Proceedings of International Joint Conference of Neural Networks', pp.291-296.

Chen, C., Skabardonis, A., and Varaiya, P. (2003), Travel time reliability as a measure of service, in '82nd Transportation Research Board', Washing D.C., USA.

Clark, S., S. Grant-Muller and H. Chen (2002) , 'Cleaning of matched license plate data', Transportation Research Record 1804, 1-7.

Cybenko, G, (1989), 'Approximation by superimposition of a sigmoidal function', Mathematical control Signals (2), 303-314.

Daganzo, C.F.(1997), 'Fundamentals of Transaction and Traffic Operations, Elsevier Science Ltd, Oxford, UK.

Demuth, H. and M. Beale (1998), Neural Network Toolbox for Use with Matlab, The MathWorks Inc.

Van der Zijpp, N.J. (1996), Dynamic Origin Destination Matrix Estimation on Motorway Networks, Ph.D. thesis, Delft University of Technology.

Dion, F., H. Rakha and Y.S. Kang (2004), 'Comparison of delay estimates at undersaturated and over-saturated pre-timed signalized intersections', Transaction Research Part B (38), 99-122.

Dougherty, M. (1995), 'A review of neural networks applied to transport', Transaction Research, Part C 3(4),147-260.

Drago, G.P. and S. Ridella (1992), 'Statistically controlled activation weight initialization', IEEE Transaction of Neural Networks, 3(4), 627-631.

Foresee, F.D. and M.T. Hagan (1997), Gauss-newton approximation to bayesian learning, in 'International Joint Conference on Neural Networks', Vol. 3, pp. 1930-1935.

Fowkes, A.S. (1983), The use of number plate matching for vehicle travel time estimation, in 'PTRC Proceedings of the 11th Annual Conference', University of Sussex, pp. 141-148.

Fu, L. and B. Hellinga (2000), 'Delay variability at signalized intersections', Transaction Research Record 1710, 215-221.

Gartner, N.H. and P. Wagner (2004), 'Analysis of traffic How characteristics on signalized arterials', Transaction Research Record 1883, 94-100.

Gault, H.E. and I.G. Taylor (1981), The use of the output from vehicle detectors to assess delay in computer-controlled area traffic control systems, Technical Report Research Report No.37,University of Newcastle, Newcastle, England.

Geman, S., E. Bienenstock and R. Doursat (1992), 'Neural networks and the Bias/Variance dilemma', Neural Computation 4, 1-58.

Geroliminis, N. and A. Skabardonis (2005), 'Prediction of arrival profiles and queue lengths along arterials by using a markov decision process', Transaction Research Record 1934, 116-124.

Haykin, S.(2001) , Kalman Filtering and Neural Networks, John Wiley and Sons Inc.

Heidemann, D. (1994), 'Queue length and delay distributions at traffic signals', Transportation Research Part B 28B, 377-389.

Helbing, D. (1997), Verkehrsdnamik Neue Physikalischen Modellierungskonzepte, Springer-Verlag, Berlin Heidelberg, Germany.

Hillier, J.A.and R. Rothery (1967), 'The synchronization of traffic signals for minimum delays', Transaction Science 1(2), 81-94.

Hinsbergen, C.P.I.J. Van, J.W.C. Van Lint and B.M. Sanders (2007), Short term traffic prediction models, in 'The 14th World Congress on Intelligent Transport Systems'.

Hoeschen, B., D. Bullock and Mark Schlappi (2005), 'A systematic procedure for estimating intersection control delay from large GPS travel time data sets', Transportation Research Record .

Hoogendoorn, S.P. and P.H.L. Bovy (2001), State-of-the-art of vehicular traffic How modeling, in 'Institute of Mechanical Engineers Part 1 Journal of Systems and Control Engineering', Vol. 215(1), Professional Engineering Publishing, pp.283-303.

Hornik, K., M. Stinchcombe and H. White (1989), 'Multilayer feedforward networks are universal approximators', Neural Networks 2(5), 359-366.

Hunt, P.B, Robertson, D.I, Bretherton, R.D and Winton, R.I. (1981). "SCOOT - a traffic responsive method of co-ordinating signals." TRRL Laboratory Report 1014.

Kimber, R.M. and E.M. Hollis (1997), Traffic queues and delays at road Junctions, Technical Report Laboratory Report 909,Transport and Road Research laboratory, Crowthorne, UK.

Kremer, S.C.(2001), 'Spatiotemporal connectionist networks a taxonomy and review', Neural Computation 13, 249-236.

Levenberg, K. (1994), 'A method forthe solution of certain non-linear problems in least squares', Quarterly Journal of Applied Mathematics II (2) , 164-168.

Li, R. (2004), Examining travel time variability using AVI data, in 'Proceedings of CAITR 2004'.

Lin, W.H., A. Kulkarni and P. Mirchandani (2004), 'Short-term arterail travel time prediction for advanced traveler information systems', Journal of Intelligent Transportation Systems 8(3), 143-154.

Lindveld, Ch.D.R. (2003), Dynamic OD Matrix Estimation a Behavioural Approach, Ph.D. thesis, TRAIL research school, The Netherlands.

Lint, H. Van (2006), Incremental and online learning through extended kalman filtering with constraint weights for freeway travel time prediction, in 'Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference'.

Liu, H., H.J. Van Duylen, J.W.C. Van Lint and M. Salamons (2006), 'Urban arterial travel time prediction with state-space neural networks and kalman filters', Transportation Research Record 1968, 99-108.

Liu, X. (2004), Development of Dynamic Recursive Models for Freeway Travel Time Prediction, Ph.D. thesis, New Jersey Institute of Technology.

MacKay, D.J.C. (1992), 'A practical bayesian framework for backprop networks', Neural Computation 4(3), 448-472.

Mandic, D.P.and J.A. Chambers (2001), Recurrent Neural Networks for Prediction, John Wiley and Sons Ltd.

Marquardt,D.W.(1963), 'An algorithm for least squares estimation of non-linear parameters', Journal of the Society of the Society of industrial and Applied Mathematics 11(2), 431-441.

McNeil, D.R. (1968), 'A solution to the fixed-cycle traffic light problem for compound poisson arrivals', Journal of Applied Probability 5, 626-635.

Miller, A.J.(1963), 'Settings for fixed-cycle traffic signals', Operations Research Quarterly 14, 373-386.

Miska, M.P. (2007), Microscopic Online Simulation for Real Time Traffic Management, Ph.D. thesis, Delft University of Technology.

Muller, T.H.J., M.P. Miska and H.J. Van Zuylen (2005), Monitoring traffic under congestion base for dynamic assignment in online prediction models, in 'The 84th Annual Meeting of Transportation Research Board'.

Nam, D.H. and D.R. Drew (1996), 'Traffc dynamics: Method for estimating freeway travel times in real time from How measurements', Journal of Transaction Engineering 122(3), 186-191.

Newell, G.F. (1960), 'Queues for a fixed-cycle traffic light', Annals of Mathematical Statistics 31(3), 589-597.

Nguyen, D. and B. Widrow (1990), Improving the learning speed of 2 layer neural networks by choosing initial values of the adaptive weights, in 'Proceedings of the International Joint Conference on Neural Networks', Vol.3, pp. 21-26.

Noland, R. and J. Polak (2002), 'Travel time variability: A review of theoretical and empirical issues', Transport Reviews 22(1), 39-54.

Olszewski, P.(1990), 'Traffc signal delay for non-uniform arrivals', Transaction Research Record 1287, 42-53.

Olszewski, P.(1994), 'Modeling probability distribution of delay at signalized intersections', Journal of Advanced Transaction 28(3), 253-274.

Pacey, G.M. (1956), The progress of a bunch of vehicles released from a traffic signal, Technical Report Rn/2665/GMP, Road Research Laboratory, London.

Palacharla, P.V. and P.C. Nelson (1999), 'Application of fuzzy logic and neural networks for dynamic travel time estimation', International Transportions in Operational Research 6,145-160.

Paterson, D.W. (2000), The Real Time Prediction of Freeway Travel Times, Ph.D. thesis, Monash University.

PTV AG (2003), Vissim User Manual, V3.70, Planung Transport Verkehr AG.

Pueboobpaphan, R. and F. Yamamoto (2005), 'Relationship between uninterrupted and interrupted speed at isolated signalized intersection', Journal of the Eastern Asia Society for Transaction Studies 6, 2194-2209 .

Robertson, D.I. (1979), Traffc models and optimum strategies of control, in 'Proceedings of the International Symposium on Traffc Control Systems',pp. 262-288.

Robinson, S. (2005), The Development and Application of an Urban Link Travel Time Model Using Data Derived from Inductive Loop Detectors, Ph.D. thesis, Imperial College London.

Rouphail, N., A. Tarko and J. Li (2000), Traffc How at signalized intersections, Technical report, Lieu H. Revised Monograph of Traffic Flow Theory, update and expansion of the Transportation Research Board (TRB) special report 16, "Traffc Flow Theory", published in 1975.

Sharkey, A.J.C. (1996), 'On combining artificial neural nets', Connection 8, 299-313.

Sisiopiku, V., N. Rouphail and A. Santiago (1994), 'Analysis of correlation between arterial travel time and detector data from simulation and field studies', Transaction Research Record 1457, 166-173.

Skabardonis, A. and N. Geroliminis (2005), Real-time estimation of travel times on signalized arterials, in The '16th International Symposium on Transportation and Traffic Theory', Elsevier.

Sun, C., G. Arr and R.P. Ramachandran (2003), 'Vehicle reidentification as method for deriving travel time and travel time distribution', Transaction Research Record 1826,25-31.

Takaba, S., T. Morita, T. Hada, T. Usami and M. Yamaguchi (1991), Estimation and measurement of travel time by vehicle detectors and license plate readers, in 'Vehicle Navigation and Information Systems', Society of Automotive Engineers, pp. 257-267.

Tampere, C.M.J. (2004), Human-Kinetic Multiclass Traffic Flow Theory and Modelling with Appliation to Advanced Driver Assistance Systems in Congestion, Ph.D. thesis, Delft University of Technology.

Transportion Research Board (2000), Highway Capacity Manual 2000, National Academies Press.

Tu, H., J.W.C. Van Lint and H.J. Van Zuylen (2007a), The impact of adverse weather on travel time variability of freeway corridors, in 'The 6th Transportion Research Board Annual Meeting'.

Tu, H., J.W.C Van Lint and H.J Van Zuylen (2007b), The impact of traffc flow on travel time variability of freeway corridors, in 'The 86th Transportion Research Board Annual Meeting'.

Tu, H. (2008), Monitoring Travel Time Reliability on Freeways, Ph.D. thesis, Delft University of Technology.

Turner, S. (2004), 'Defining and measuring traffic data quality, white paper on recommended approaches', Transaction Research Record 1870, 62-69 .

Turner, S.M., W.L. Eisele, R.J. Benz and D.J. Holdener (1998), Travel time data collection handbook, Technical Report FHWA-PL-98-035,Texas Transportation Institute, The Texas A M University.

Usami, T., K. Ikenoue and T. Miyashako (1986), Travel time prediction algorithm and signal operations at critical intersections for controlling travel time, in 'Second International Conference on Road Traffic Control', Institute of Electrical and Electronics Engineers, pp. 205-208.

Van Lint, J.W.C (2004), Reliable Travel Time Prediction for Freeways, Ph.D. thesis, Delft University of Technology.

Van Lint, J.W.C and H.J Van Zuylen (2005), 'Monitoring and predicting freeway travel time reliability - using width and skew of day-to-day travel time distribution', Transaction Research Record 1917 , 54-62.

Vanajakshi, L.D. (2004), Estimation and Prediction of Travel Time from Loop Detector Data for Intelligent Transportation System Applications, Ph.D. thesis, Texas A M University.

Viti, F. (2006), The Dynamics and the Uncertaity of Delays at Signals, Ph.D. thesis, Delft Unviersity of Technology.

Viti, F. and H.J. Van Zuylen (2004), 'Modeling queues at signalized intersections', Transportation Research Record 1 1883, 68-77.

Webster, F.V. (1958), Traffic signal settings, Technical Report Road Research Technical Paper No.39, Road Research Laboratory, Her Majesty Stationary Office, London, UK.

Weijermars, W.A.M. and E.C. VanBerkum (2006), 'Detection of invalid loop detector data in urban areas', Transaction Research Record 1945, 82-88.

Wessels, L.F.A. and E. Barnard (1992), 'Avoiding false local minima by proper initialization of connections', IEEE Transaction of Neural Network 3,899-905.

Wilson, C. J., Millar, G., Tudge, R., (2006), Microsimulation Evaluation of Benefits of SCATS-Coordinated Traffic Control Signals, in the TRB 85th Annual Meeting Compendium of Papers CD-ROM.

You, J. and T.J. Kim (2000), 'Development and evaluation of a hybrid travel time forecasting model', Transaction Research Part C:Emerging Technologies 8(1-6),231-256.

You, J. and T.J. Kim (2006), Methods and Models in Transport and Telecommunication, Springer, Chapter Towards Developing a Travel Time Forecasting Model for Location-Based Services-A Review, pp.45-61.

Zhang, H. M. and E. Kwon (1997), Travel time estimation on urban arterials using loop detector data, Technical report, Public Policy Center, University of Iowa.

Zijderveld, A. (2003), Neural Network Design Strategies and Modelling in Hydroinformatics, Ph.D. thesis, Delft University of Technology.

Zuylen, H.J. Van (1985), De Dynamiek Van achtri8en Voor Een Verkeerslicht (The Dynamics of Queues at traffic Lights), Dutch lecture notes VB41. Tilburg, Verkeersakademie.

Zuylen, H.J. Van (2002), traffic Control for Intersections: Design and Evaluation, Delft University of Technology.

Zuylen, H.J. Van and F. Viti (2003), Uncertainty and the dynamics of queues at controlled intersections, in 'Proceedings CTS-IFAC Conference 2003', Elsevier, Tokyo.

Zuylen, H.J. Van and F. Viti (2007), A probabilistic model for queues, delays and waiting time at controlled intersections, in 'The 86th Transportation Research Board Annual Meeting, CD-ROM'.

Zuylen, H.J. Van and T.H.J. Muller (2002), Regiolab delft, in 'The 9th World Congress on Intelligent Transport Systems', Chicago, Illinois, USA.

Appendix A

Performance Indicators

Let y_i be the ith output calculated from the model, and d_i be the ith observation. The mean of model output and observations are expressed as

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad \text{and} \quad \overline{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{A.1}$$

where N denotes the total number of observations.

Table A.1 lists the performance indicators used in this dissertation.

Table A.1: Performance indicators

Abbreviation	Meaning	Formula
MARE	Mean Absolute Relative Error	$100\frac{1}{N}\sum_{i=1}^{N}\left\lvert\frac{y_i-d_i}{d_i}\right\rvert$
MSE	Mean Squared Error	$\frac{1}{N}\sum_{i=1}^{N}(y_i-d_i)^2$
RMSE	Root Mean Squared Error	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-d_i)^2}$
RMSEP	Root Mean Squared Error Proportional	$100\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-d_i)^2}}{\overline{d}}$
SSE	Sum Squared Error	$\sum_{i=1}^{N}(y_i-d_i)^2$

Appendix B

Various Travel Time Collection Systems

For a comprehensive overview of travel time data collection systems we refer to (Turner et al. 1998).

Distance Measuring Instrument

The distance measuring instrument (DMI) calculates distance and speed using pulses from a sensor attached to the vehicle's transmission. The DMI coverts the pulses to units of measure and calculates a speed from an internal clock.

License Plate Matching

In general, license plate matching (LPM) techniques consist of collecting vehicle license plate numbers and arrival times at various points, matching the license plates between consecutive points, and then deriving travel times from the difference in arrival times. There are four basic methods of collecting and processing license plates: manual, portable computer, video with manual transcription, and video with character recognition.

Vehicle Signature Matching

By using unique vehicle features, so called vehicle "signatures" (VS), instead of license plates to identify the same vehicle passing by various points, existing point detection devices (e.g. inductive loop detectors, laser sensors, weigh-in-motion sensors, and video cameras) are able to be extensively utilized. For example, the vehicle signature from a loop detector can be defined as frequency detuning curve. Different types and classes of vehicles provide somewhat characteristic detuning curves. The travel time is the time difference between the arrival times when a matched vehicle passes two consecutive locations.

Automatic Vehicle Identification

Similarly, automatic vehicle identification (AVI) uses radio frequency (RF) signals as vehicle 'signatures'. The AVI technology was originally applied for electronic toll collection (ETC). Tags, known as transponders, are electronically encoded with unique identification numbers, and attached on vehicles. Roadside antennas emit radio frequency signals and receive the reflected signals from the tags. If a same identification number is recorded between two roadside antennas, the time difference passing these antennas are the travel time.

Global Positioning System

The Global Positioning System (GPS) was originally developed by the Department of Defense for the tracking of military ships, aircraft, and ground vehicles. Signals sent from several satellites orbiting the earth are utilized to monitor location, direction, and speed of the vehicles of interest. The GPS records very detailed position information, which can easily be used to calculate the travel time.

Emerging Technologies

Platoon matching (PM) is similar to vehicle signature matching in that it relies on identifying, extracting, and matching unique features between two consecutive roadway locations. The underlying concept of platoon matching is based on identifying unique relationships between vehicles, whereas vehicle signature matching relies on the specific characteristics of a single vehicle or a sequence of vehicles. *Cellular phone tracking* (CPT) is similar to GPS in the sense of using satellites to track the position for the customs of interest. *Aerial survey* (AS) is conducted from fixed wing aircraft, helicopter, observation balloons, or even satellites. The sequence of images record very detailed information of vehicles within given section, which can be used to compute individual travel time.

Appendix C

Measured Mean Speed with respect to Detector Locations

For a stationary and homogeneous traffic flow, a equilibrium state, so called fundamental diagram, exists between density and flow (top Figure shown in Figure C.1). When traffic state changes from one to another state, a boundary between these two states is established. This boundary is referred to as shock wave. A concept of shock wave at signalized intersection is illustrated in the bottom Figure of Figure C.1. For example, when traffic signal turns to red, traffic stop still (here ignore deceleration time), which is the state 4 (critical density) in the fundamental diagram. With this shock wave concept the time/space mean speed and travel time can be precisely calculated.

Let vehicular traffic flow q denote the number of vehicles passing a location per unit time, vehicular density ρ denote the number of vehicles present on a unit space at a specific time instant, space mean speed v_s denote the mean speed of vehicles present on a unit space at a specific time instant, and time mean speed v_t denote the mean speed of vehicles passing a location in a unit time.

$$v_s = \frac{q}{\rho}$$

It is important to note that any one of these three variables can be deduced from the other two. When traffic state changes from one to another state, a boundary between these two states is established. This boundary is referred to as shock wave. A concept of shock wave at signalized intersection is illustrated in Figure C.1. The speed of shock wave is calculated by

$$w_{ij} = \frac{q_i - q_j}{\rho_i - \rho_j}$$

where w_{ij} represents the speed of shock wave between traffic state i and j. Symbols q_i and q_j, ρ_i and ρ_j represent flow rate, density of any traffic state i and j, respectively. According to Figure C.1, traffic states 1, 2, 3, and 4 refer to the approaching traffic, stopped traffic, capacity traffic, and the zero flow traffic states, respectively. Given homogeneity and stationarity, the queue builds up at t_1 (traffic light turns red), spills back up to location

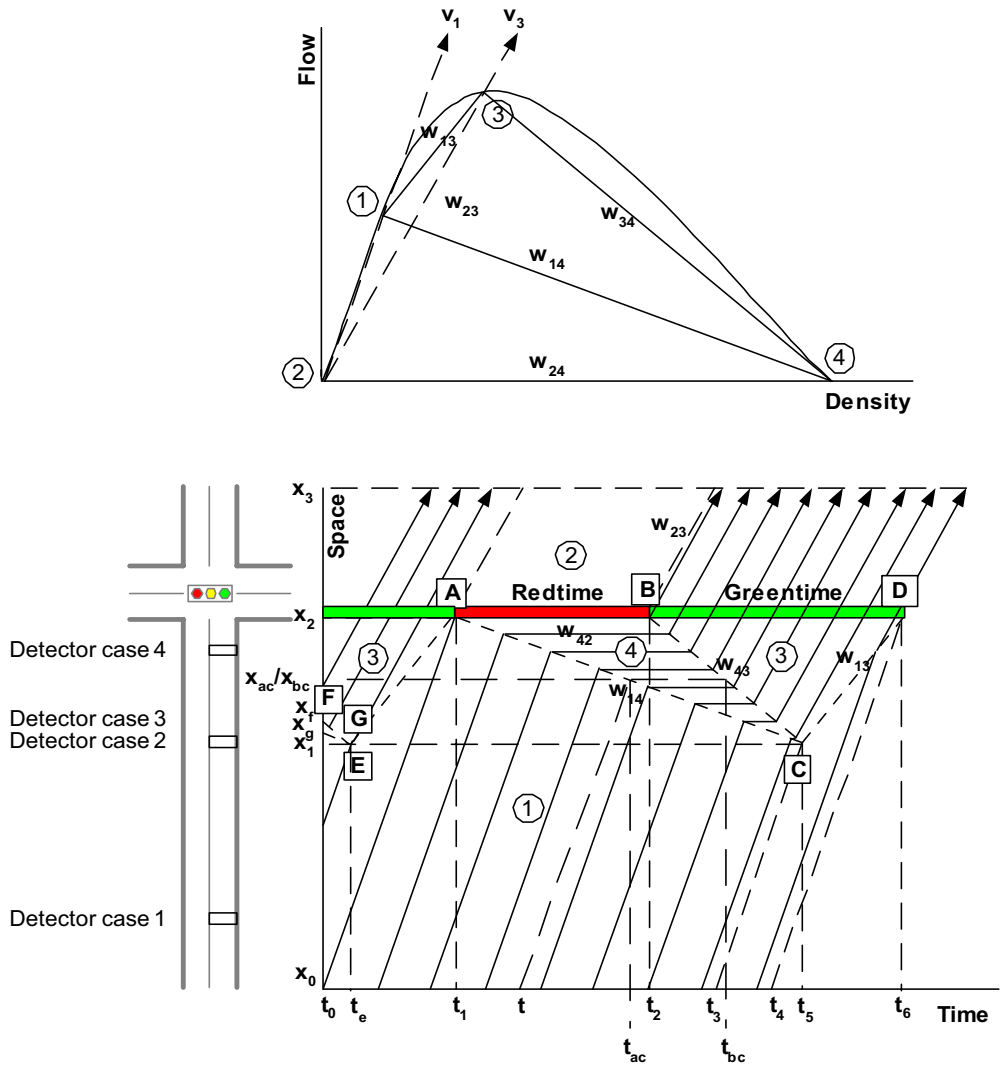


Figure C.1: Shock wave analysis at a signalized intersection. (a) fundamental diagram of flow and density, (b) shock wave analysis

x_1 at t_5, and dissolves completely at t_6 (the end of green time). The coordinate of point C where the queue start to dissipate can be determined by

$$\begin{aligned} x_1 &= x_2 - \frac{w_{14}w_{43}(t_1 - t_2)}{w_{14} - w_{43}} \\ t_5 &= \frac{w_{14}t_1 - w_{43}t_2}{w_{14} - w_{43}} \end{aligned} \tag{C.1}$$

in which t_2 can be derived by extending t_1 with red time r_e, that is $t_2 = t_1 + r_e$.

The vehicles departing during time period $[t_0, t_4]$, for example see in Figure C.1, can be classified into two groups: vehicles departing in $[t_0, t_3]$ traverse at approaching speed (state 1) at the beginning, after passing the shock wave w_{14} at point (x_{ac}, t_{ac}) they wait in the queue (state 4), and then drive at capacity speed (state 3) after crossing the shock

wave w_{34} at point (x_{bc}, t_{bc}); while, vehicles departing in $[t_3, t_4]$ run at approaching speed (state 1) at the beginning, and transit to capacity speed (state 3) after moving over the shock wave w_{34}. The two time instants t_3 and t_4 are calculated based on known t_5 and t_6

$$\begin{aligned} t_3 &= t_5 - \frac{x_1 - x_0}{v_1} \\ t_4 &= t_6 - \frac{x_2 - x_0}{v_1} \end{aligned} \tag{C.2}$$

in which t_6 can be derived by extending t_1 with cycle time c_y, that is $t_6 = t_1 + c_y$.

In a general term, the crossing points for a vehicle departing at time instant $t \in [t_0, t_3]$ can be calculated

$$x_{ac}(t) = \frac{|w_{14}|\, v_1\, (t_1 - t) + |w_{14}|\, x_0 + v_1 x_2}{|w_{14}| + v_1} = AX_{ac}t + BX_{ac} \tag{C.3}$$

$$\begin{aligned} t_{ac}(t) &= \frac{v_1 t + |w_{14}|\, t_1 + (x_2 - x_0)}{|w_{14}| + v_1} = AT_{ac}t + BT_{ac} \\ x_{bc}(t) &= x_{ac}(t) \\ t_{bc}(t) &= t_2 + \frac{x_2 - x_{ac}(t)}{|w_{43}|} = AT_{bc}t + BT_{bc} \end{aligned} \tag{C.4}$$

in which $AX_{ac} = \frac{-|w_{14}|v_1}{|w_{14}|+v_1}$, $BX_{ac} = \frac{|w_{14}|x_0+v_1x_2+|w_{14}|v_1t_1}{|w_{14}|+v_1}$, $AT_{ac} = \frac{v_1}{|w_{14}|+v_1}$, $BT_{ac} = \frac{|w_{14}|t_1+(x_2-x_0)}{|w_{14}|+v_1}$, $AT_{bc} = \frac{-AX_{ac}}{|w_{43}|}$, *and* $BT_{bc} = t_2 + \frac{x_2-BX_{ac}}{|w_{43}|}$.

Similarly, we get the crossing point for a vehicle departing during $t \in [t_3, t_4]$

$$x_{cd}(t) = \frac{v_1 x_2 - |w_{13}|\, x_0 - |w_{13}|\, v_1 (t_6 - t)}{v_1 - |w_{13}|} = AX_{cd}t + BX_{cd} \tag{C.5}$$

$$t_{cd}(t) = \frac{x_2 - x_0 - (|w_{13}|\, t_6 - v_1 t)}{v_1 - |w_{13}|} = AT_{cd}t + BT_{cd} \tag{C.6}$$

in which $AX_{cd} = \frac{|w_{13}|v_1}{v_1-|w_{13}|}$, $BX_{cd} = \frac{v_1x_2-|w_{13}|x_0-|w_{13}|v_1t_6}{v_1-|w_{13}|}$, $AT_{cd} = \frac{v_1}{v_1-|w_{13}|}$, *and* $BT_{cd} = \frac{x_2-x_0-|w_{13}|t_6}{v_1-|w_{13}|}$.

C.1 Mean Travel Time

Thus, the travel times for a single vehicle departing at time instant t is obtained:

$$TT(t) = \begin{cases} \frac{x_{ac}-x_0}{v_1} + (t_{bc}(t) - t_{ac}(t)) + \frac{x_3-x_{ac}}{v_3} & t \in [t_0, t_3] \\ \frac{x_{cd}-x_0}{v_1} + \frac{x_3-x_{cd}}{v_3} & t \in [t_3, t_4] \end{cases} \tag{C.7}$$

With obtained $TT(t)$, the *mean travel time* for vehicles departing during time period $[t_0, t_4]$ can be expressed:

$$\widetilde{TT} = \frac{\int_{t_0}^{t_4} q_1 TT(t)dt}{q_1(t_4 - t_0)} \tag{C.8}$$

Combining equation C.4, C.6 and C.7, C.8 can be rewritten as

$$\widetilde{TT} = \frac{\left\{\left[\frac{\alpha_1}{2}(t_3^2 - t_0^2) + \beta_1(t_3 - t_0)\right] + \left[\frac{\alpha_2}{2}(t_4^2 - t_3^2) + \beta_2(t_4 - t_3)\right]\right\}}{(t_4 - t_0)} \tag{C.9}$$

where $\alpha_1 = \frac{AX_{ac}}{v_1} + (AT_{bc} - AT_{ac}) - \frac{AX_{ac}}{v_3}$, $\beta_1 = \frac{BX_{ac} - x_0}{v_1} + (BT_{bc} - BT_{ac}) + \frac{x_3 - BX_{ac}}{v_3}$, $\alpha_2 = \frac{AX_{cd}}{v_1} - \frac{AX_{cd}}{v_3}$, *and* $\beta_2 = \frac{BX_{cd} - x_0}{v_1} + \frac{x_3 - BX_{cd}}{v_3}$.

C.2 Space Mean Speed

By definition, space mean speed is the arithmetic mean of the speeds that are present on a road segment at a given moment. As an example, consider a road segment $[x_0, x_2]$ and time period $[t_0, t_4]$. Then, the space mean speed at time instant t can be expressed as:

Case 1 $t_0 < t < t_e$

Three different traffic states along the road segment of interest can be identified. The states between start location (location x_0) and line GE, between line GE and FE, and between line FE and end location (location x_2) are traffic state 1, 4, and 3, respectively. Note that the speed of traffic 4 is zero. Thus, the space mean speed equals

$$v_s(t) = \frac{A1 + A2 + A3}{B1 + B2 + B3} \tag{C.10}$$

$$A1 = k_1 v_1 \left(\frac{x_1(t - t_0) + x_g(t_e - t)}{t_e - t_0}\right) \tag{C.11}$$

$$A2 = k_4 v_4 \left(\frac{x_1(t - t_0) + x_f(t_e - t)}{t_e - t_0} - \frac{x_1(t - t_0) + x_g(t_e - t)}{t_e - t_0}\right) \tag{C.12}$$

$$A3 = k_3 v_3 \left(x_2 - \frac{x_1(t - t_0) + x_f(t_e - t)}{t_e - t_0}\right) \tag{C.13}$$

$$B1 = k_1 \left(\frac{x_1(t - t_0) + x_g(t_e - t)}{t_e - t_0}\right) \tag{C.14}$$

$$B2 = k_4 \left(\frac{x_1(t - t_0) + x_f(t_e - t)}{t_e - t_0} - \frac{x_1(t - t_0) + x_g(t_e - t)}{t_e - t_0}\right) \tag{C.15}$$

$$B3 = k_3 \left(x_2 - \frac{x_1(t - t_0) + x_f(t_e - t)}{t_e - t_0}\right) \tag{C.16}$$

Note that the speed of traffic 4 is zero. equation **??** *can be rewritten as*

$$v_s(t) = \frac{A1 + A3}{B1 + B2 + B3} \tag{C.17}$$

Case 2 $t_e < t < t_1$

Similarly, the space mean speed can be expressed as

$$v_s(t) = \frac{k_1 v_1 \left(x_1 + \frac{(x_2 - x_1)(t - t_e)}{t_1 - t_e}\right) + k_3 v_3 \left(x_2 - x_1 - \frac{(x_2 - x_1)(t - t_e)}{t_1 - t_e}\right)}{k_1 \left(x_1 + \frac{(x_2 - x_1)(t - t_e)}{t_1 - t_e}\right) + k_3 \left(x_2 - x_1 - \frac{(x_2 - x_1)(t - t_e)}{t_1 - t_e}\right)} \tag{C.18}$$

Case 3 $t_1 < t < t_2$

$$v_s(t) = \frac{k_1 v_1 \left(x_1 + \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_1}\right)}{k_1 \left(x_1 + \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_1}\right) + k_4 \left(x_2 - x_1 - \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_1}\right)} \tag{C.19}$$

Case 4 $t_2 < t < t_4$

$$v_s(t) = \frac{k_1 v_1 \left(x_1 + \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_1}\right) + k_3 v_3 \left(x_2 - x_1 - \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_2}\right)}{k_1 \left(x_1 + \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_1}\right) + k_4 \frac{(t_5 - t)(x_2 - x_1)(t_2 - t_1)}{(t_5 - t_2)(t_5 - t_1)} + k_3 \left(x_2 - x_1 - \frac{(x_2 - x_1)(t_5 - t)}{t_5 - t_2}\right)} \tag{C.20}$$

C.3 Time Mean Speed

As we discussed before, time mean speed is a local variable, meaning that it can only be observed at a specific location. Since the interruption of traffic signal to traffic stream leads to a formation of queue, different detector locations would give different values of time mean speed (Pueboobpaphan & Yamamoto 2005). As an example, consider a case where the detector location is at a distance x. The time mean speed during time period $[t_0, t_4]$ can be categorized into four cases: $x_0 \leq x < x_1, x_1 \leq x < x_g, x_g \leq x < x_f$, *and* $x_f \leq x < x_2$. ($x_g = x_1 + |w_{14}|\,(t_e - t_0)$, $x_f = x_1 + |w_{43}|\,(t_e - t_0)$, *and* $t_e = t_1 - (t_6 - t_5)$)

Case 5 $x_0 \leq x < x_1$ *(detector case 1 in Figure C.1)*

Vehicles drives at approaching speed during the whole period $[t_0, t_4]$

$$v_t(x) = v_1$$

Case 6 $x_1 \leq x < x_g$ *(detector case 2)*

The times when shock wave GE, FE, EA, AC and BC cross detector location are defined as td_{ge},td_{fe}, td_{ea}, td_{ac}, *and* td_{bc}, *respectively.*

$$\begin{aligned} td_{ge} &= t_e - \frac{x_d - x_1}{|w_{14}|} \\ td_{fe} &= t_e - \frac{x_d - x_1}{|w_{43}|} \\ td_{ea} &= t_e + \frac{x_d - x_1}{|w_{13}|} \\ td_{ac} &= t_5 - \frac{x_d - x_1}{|w_{14}|} \\ td_{bc} &= t_5 - \frac{x_d - x_1}{|w_{43}|} \end{aligned} \tag{C.21}$$

The time mean speed during the whole period $[t_0, t_4]$ *can be expressed*

$$\begin{aligned} v_t(x) &= \frac{\int_{t_0}^{td_{ge}} v_1 dt + \int_{td_{ge}}^{td_{fe}} 0 dt + \int_{td_{fe}}^{td_{ea}} v_3 dt + \int_{td_{ea}}^{td_{ac}} v_1 dt + \int_{td_{ac}}^{td_{bc}} 0 dt + \int_{td_{bc}}^{t_4} v_3 dt}{t_4 - t_0} \\ &= \frac{v_1 \left(td_{ge} + td_{ac} - td_0 - td_{ea}\right) + v_3 \left(td_{ea} + t_4 - td_{fe} - td_{bc}\right)}{t_4 - t_0} \end{aligned}$$

Case 7 $x_g \leq x < x_f$ *(detector case 3)*

$$v_t(x) = \frac{v_1 \left(td_{ac} - td_{ea}\right) + v_3 \left(td_{ea} + t_4 - td_{fe} - td_{bc}\right)}{t_4 - t_0}$$

where td_{fe}, td_{ea}, td_{ac}, *and* td_{bc} *can be calculated with the equation C.21.*

Case 8 $x_f \leq x < x_2$ *(detector case 4)*

$$v_t(x) = \frac{v_1 \left(td_{ac} - td_{ea}\right) + v_3 \left(td_{ea} + t_4 - t_0 - td_{bc}\right)}{t_4 - t_0}$$

where td_{ea}, td_{ac}, *and* td_{bc} *can be calculated with the equation C.21.*

Based on the above set of equations, if detector location, arrival flow rate and signal setting are given, one can estimate the mean travel time and the value of space/time mean speed. Note that, to do this, fundamental diagram function of flow and density must be known or given in advance.

Appendix D

Levenberg-Marquardt and Bayesian Regularization

Training neural network models refers to finding the appropriate parameters (weights and bias) which minimize an objective cost function. In general, the neural network models can be trained in a supervised manner, given sufficient data pairs (inputs and outputs). There are many possible choice of cost function which can be used, depending on the particular application. Levenberg-Marquardt algorithm is designed specifically for minimizing a sum of squares error. Bayesian regularization aims to achieve good generalization of models so as to avoid over fitting.

D.1 Levenberg-Marquardt Algorithm

Consider a sum-of-squares error function:

$$E = \frac{1}{2}\sum_{p=1}^{M}\left(Y\left(p\right) - \widetilde{Y}\left(p\right)\right)^2 = \frac{1}{2}\left\|\epsilon\right\|^2 \tag{D.1}$$

where W denotes the parameters in the SSNN model, $Y\left(p\right)$ denotes the output calculated from the SSNN, $\widetilde{Y}\left(p\right)$ denotes the desired output, ϵ denotes the vector of errors, and M denotes the total number of data pairs in the training data set.

Suppose we are currently at a point W^{old} in weight parameter space and we move to a new point W^{new}. If the displacement $W^{new} - W^{old}$ is small then we can expand the error vector ϵ to first order in a Taylor series

$$\epsilon\left(W^{new}\right) = \epsilon\left(W^{old}\right) + J \cdot \left(W^{new} - W^{old}\right) \tag{D.2}$$

where $J = \frac{\partial \epsilon}{\partial W^{old}}$ denotes Jacobian matrix of error function with respect to the weights. Then, the error function D.1 can be written as

$$E = \frac{1}{2}\left\|\epsilon\left(W^{old}\right) + J \cdot \left(W^{new} - W^{old}\right)\right\|^2 \tag{D.3}$$

If we minimize this error with respect to the new weights W^{new} we obtain

$$W^{new} = W^{old} - \left(J^T J\right)^{-1} J^T \epsilon \left(W^{old}\right) \tag{D.4}$$

In principle, the update formula D.4 could be applied iteratively in order to try to minimize the error function. The problem with such an approach is that the step size which is given by D.4 could turn out to be relatively large, in which case the linear approximation D.3 might not be valid (Bishop 2005). Levenberg (1944) and Marquardt (1963) addressed this problem by minimizing the error function while at the same time trying to keep the step size small so as to ensure that the linear approximation remains valid. A modified error function is formulated as

$$\widetilde{E} = \frac{1}{2} \left\| \epsilon \left(W^{old}\right) + J \cdot \left(W^{new} - W^{old}\right) \right\|^2 + \lambda \left\| \left(W^{new} - W^{old}\right) \right\|^2 \tag{D.5}$$

where λ governs the step size. For very small values of λ error function D.5 recover the Newton formula, while for large values of λ it recover standard gradient descent. Minimizing the modified error function D.5 with respect to W^{new} yields

$$W^{new} = W^{old} - \left(H + \lambda I\right)^{-1} J^T \left(W\right) e \left(W\right)$$

where H denotes the Hessian matrix of error function with respect to the weights. For simplicity, the Hessian matrix can be approximated:

$$H \approx J^T J$$

For implementation details see for example (Demuth & Beale 1998).

D.2 Bayesian Regularization

This section is largely based on the work of (MacKay 1992). The central idea of Bayesian regularization is to decrease the tendency of a neural network model to over fit the training data. Larger and more weights may perform better on the training data, but also may yield poor generalization with respect to unseen data. Thus, the objective function minimizes not only a sum-of-squares error function while at the same time trying to minimize the sum of squares of weights

$$F(W) = \beta E_D + \alpha E_W = \beta \sum_{p=1}^{M} \frac{1}{2} \left(Y\left(p\right) - \widetilde{Y}\left(p\right)\right)^2 + \alpha \sum_{i=1}^{N} \frac{1}{2} W_i^2 \tag{D.6}$$

where D denotes the data set, W denotes the vector of network parameters, M denotes the total number of data pairs in the training data set, N denotes the total number of weights, α and β are objective function parameters which dictate the emphasis of the training.

In the Bayesian framework the weights W of the neural network are considered as random variables. The density function of W can be expressed according to Bayes' rule

$$P\left(W|D,\alpha,\beta,G\right)=\frac{P\left(D|W,\beta,G\right)\cdot P\left(W|\alpha,M\right)}{P\left(D|\alpha,\beta,G\right)} \tag{D.7}$$

where G denotes the neural network model used; $P\left(W|\alpha,G\right)$ denotes the prior density of weights, which represents our prior knowledge of the weights before any data is used for training; $P\left(D|W,\beta,G\right)$ is the likelihood function, which is the probability of the data occurring given the particular neural network model G, weights W and objective function parameter β; $P\left(D|\alpha,\beta,G\right)$ is a normalization factor, which guarantees the total probability is 1.

If we assume that the noise and the prior distribution for the weights are Gaussian distributed according to $N\left(1,1/\beta\right)$ and $N\left(1,1/\alpha\right)$ respectively, the probability densities can be written as

$$P\left(D|W,\beta,G\right)=\frac{1}{Z_D\left(\beta\right)}\exp\left(-\beta E_D\right) \tag{D.8}$$

and

$$P\left(W|\alpha,G\right)=\frac{1}{Z_W\left(\alpha\right)}\exp\left(-\alpha E_W\right) \tag{D.9}$$

where $Z_D\left(\beta\right)=\left(\pi/\beta\right)^{M/2}$ and $Z_W\left(\alpha\right)=\left(\pi/\alpha\right)^{N/2}$. Substitute equation D.9 and equation D.8 into equation D.7, we obtain the posterior probability of W:

$$\begin{aligned} P\left(W|D,\alpha,\beta,G\right) &= \frac{\frac{1}{Z_W(\alpha)}\frac{1}{Z_D(\beta)}\exp\left(-\alpha E_W-\beta E_D\right)}{P\left(D|\alpha,\beta,G\right)} \\ &= \frac{1}{P\left(D|\alpha,\beta,G\right)}\cdot\left(\frac{1}{Z_F\left(\alpha,\beta\right)}\exp\left(-F(W)\right)\right) \end{aligned} \tag{D.10}$$

In the Bayesian framework, the optimal weights should maximize the posterior probability equation D.10, which is equivalent to minimizing the regularized objective function given in equation D.7. With equation D.8, equation D.9 and equation D.10, $P\left(D|\alpha,\beta,G\right)$ can be expressed as

$$P\left(D|\alpha,\beta,G\right)=\frac{Z_F\left(\alpha,\beta\right)}{Z_D\left(\beta\right)Z_W\left(\alpha\right)} \tag{D.11}$$

Note that $Z_D\left(\beta\right)$ and $Z_W\left(\alpha\right)$ are known in equation D.8 and D.9. What needs to be estimated is $Z_F\left(\alpha,\beta\right)$. Since the objective function has a quadratic shape in a small area surrounding a minimum point with the parameter W^{MP}, we expand $F\left(W\right)$ around the minimum point of the posterior density with Taylor series expansion. This yields

$$F(W) = F(W_{MP}) + \frac{1}{2}(W - W_{MP})^T H (W - W_{MP}) \tag{D.12}$$

where H is the Hessian matrix of the objective function. Then, $Z_F(\alpha, \beta)$ is the Gaussian integral:

$$\begin{aligned} Z_F(\alpha, \beta) &= \int d^N W \exp(-F(W, \alpha, \beta)) \\ &\approx (2\pi)^{N/2} \exp(-F(W)) (\det H)^{-1/2} \end{aligned} \tag{D.13}$$

To optimize the value of α and β, we apply Bayes' rule

$$P(\alpha, \beta | D, G) = \frac{P(D|\alpha, \beta, G) \cdot P(\alpha, \beta|G)}{P(D|G)} \tag{D.14}$$

Assuming a uniform prior density $P(\alpha, \beta|G)$ for the regularization parameter α and β, then maximizing the posterior of α and β is achieved by maximizing the likelihood function $P(D|\alpha, \beta, G)$, whose logarithm can be written as

$$P(D|\alpha, \beta, G) = -\alpha E_W^{MP} - \beta E_D^{MP} - \frac{1}{2} \log \frac{Z_W(\alpha)^2 \cdot Z_D(\beta)^2 \det H}{(2\pi)^N} \tag{D.15}$$

We can solve for the optimal value of α and β at the minimum point by taking the derivatives of equation D.15 with respect to α and β, and set them equal to zero. This yields

$$\begin{aligned} \alpha^{MP} &= \frac{\gamma}{2E_W(W^{MP})} \\ \beta^{MP} &= \frac{M - \gamma}{2E_D(W^{MP})} \end{aligned}$$

where $\gamma = N - 2\alpha^{MP} \cdot trace(H^{-1})$ is called the effective number of parameters. Foresee & Hagan (1997) proposed using the Gauss-Newton approximation to Hessian matrix H as follows

$$\widehat{H} = \beta J^T J + \alpha I$$

where J denotes the Jacobian matrix of error function with respect to the SSNN weights, $J = \frac{\partial E(W)}{\partial W}$, and I denotes the identity matrix.

D.3 Levenberg-Marquardt with Bayesian Regularization

In short, the training procedure can be summarized as

Step 1. Initialize weights W and objective function parameters α and β.

Step 2. Use LM algorithm to calculate new weights with fixed α and β based on output error $e\,(W)$

$$W^{new} = W^{old} - \left(\widehat{H}\,(W) + \lambda I\right)^{-1} J^{T}\,(W)\, e\,(W)$$

Step 3. Optimize α and β given new weights.

Step 4. If stop criteria met (minimum performance goal, maximum number of epochs) then stop, otherwise continue with step 2.

Appendix E

Algorithms for Detecting Travel Time Outliers

E.1 Percentile Test

The percentile test defines outliers to be all values in the time period (usually 5 or 15 minutes) which do not fall within a pre-defined percentile scope (upper percentile PT^u and lower percentile PT^l). The travel time of vehicle i, TT_i, is determined as outlier, when the following conditions are met.

$$TT_i \leq PT^l \text{ or } PT^u \leq TT_i$$

To apply this test, a prior knowledge of the distribution of travel time is required. Clark et al. (2002) proposed to use the 10th and 90th percentiles. For different applications, the percentile scope might be different.

E.2 Deviation Test

The deviation test considers an individual travel time as a outlier, if the value of the individual travel time is further than a critical distance CD from the median of the n travel times in the time period. The travel times that fall out of the scope $[TT_m - CD, TT_m + CD]$ will be treated as outliers. In literature, relatively little research on ANPR data outlier detection has been undertaken. Fowkes (1983) maybe the first to propose an equation as follow:

$$CD = \frac{PC_3 - PC_1}{1.35} \times \left(t_{0.025,n^*}\right) \times F_1$$

where PC_3 refers to upper quartile of the sample, PC_1 refers to lower quartile of the sample, $t_{0.025,n^*}$ refers to t-statistic ($n^* \leq n-1$), F_1 is a correction term which is obtained by

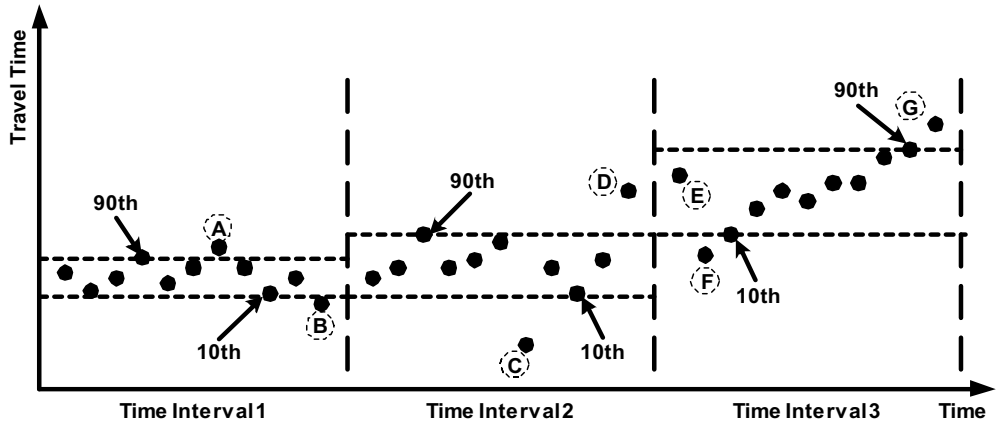


Figure E.1: Difficulty of identifying outliers with percentile test approach

$$F_1 = \begin{cases} \sqrt{\frac{1}{\frac{2}{\pi}+\frac{1}{n}(\frac{6}{\pi}-1)}} & n \text{ even} \\ \sqrt{\frac{1}{\frac{2}{\pi}+\frac{1}{n}(\frac{4}{\pi}-1)}} & n \text{ odd} \end{cases}$$

In (Clark et al. 2002), the critical distance is calculated by

$$CD = \delta_a \times \frac{\sum_{i=1}^{n} |TT_i - TT_{median}|}{n}$$

where δ_a is a parameter. Clark et al. (2002) proposed to use $\delta_a = 3$.

E.3 Critique of Existing Approaches

Clearly, percentile test will be extreme, in that the observations falling out of the percentile scope will be classified as outliers no matter they are actually valid data. For example, for the case of 10th and 90th percentile scope, 20% of all the observations will be classified as outliers. In the example of Figure E.1, the travel time record, C and D which lie in time interval 2 are successfully identified as outliers, while A and B which lie in time interval 1 are misclassified as outliers.

Furthermore, if real outliers lie close to the boundary (start and end) of a time interval, three kinds of mistakes might be caused in condition that travel time increases during the time window of interest: (a) the outlier with value close to the median value, like travel time record E in time interval 3, cannot be identified; (b) the low values at the start of increasing travel time trend which are less than the lower percentile might be misidentified as outliers (travel time record F); (c) the high values at the end of increasing travel time trend which are larger than upper percentile might be misidentified as outliers (travel time record G).

The deviation test screens out all the travel times, which are further than a critical distance from the median/mean of the travel times in the time interval of interest. The same critical distances used for the values larger and smaller than the median implies an assumption of symmetric distribution of travel times. However, the distributions of travel time are skewed to left (congestion onset and dissolve) and right (congestion) (Van Lint & Zuylen 2005). It is certainly inappropriate to remove observations, which are larger and smaller than the median/mean value of travel times in the time interval of interest, with same deviation distance.

E.4 A New Proposed Approach

Our aim is to propose a generic procedure which can overcome the drawbacks of existing approaches. The basic concept of the proposed one is similar to the existing approaches: divide time series of travel time records by a fixed length time window, find out the records that falls out of desired confidence scope. The major features of the new approach are twofold: (a) the desired confidence scope is derived from available data; (b) a moving time window is used to deal with the problem of the above stated travel time record D and E. The former strategy avoids to assume a known distribution for travel time, unlike the existing approaches. This makes the new approach be easily transfer to any location due to travel time distribution might reveal different profiles in terms of different application locations. So far, no any literature shows any known distribution function can fit travel time distribution well in different locations. Also, the distribution varies with different traffic conditions. The second strategy is able to address the problem when travel time increase or decrease significantly within a time window.

E.4.1 A generic procedure of outlier detection for travel time records

Step 1: Initiate time window T_w

Determine confidence scope $\left[T_l^k, T_u^k\right]$ in terms of travel time $TT = k$. For each travel time record $TT_i(t)$, select all the travel time records Ω within the time window $[t-T_w, t+T_w]$. Since the data here is available "off-line" it is possible to use future observations to establish this context. From the records in Ω, calculate travel time upper and lower limits, namely a statistical value

$$\begin{aligned} \lambda_i^u &= TT_{pu}^{\Omega} - TT_i(t) \\ \lambda_i^l &= TT_i(t) - TT_{pl}^{\Omega} \end{aligned}$$

where TT_{pu}^{Ω} and TT_{pl}^{Ω} refer to upper and lower percentile, respectively. Note that TT_{pu}^{Ω} and TT_{pl}^{Ω} are not necessarily symmetric, like $TT_{pu}^{\Omega} = 80th$ and $TT_{pl}^{\Omega} = 20th$, percentile because the distributions of travel time usually are not symmetric in most cases.

For a certain value of travel time $TT = k$, two sets of statistical value $\Psi^u\ (TT = k)$ and $\Psi^l\ (TT = k)$ can be constructed by grouping same value of travel time during different

time and days. Finally, T_l^k and T_u^k will be determined by choosing percentile value from the Ψ^l and Ψ^u, respectively as

$$\begin{aligned} T_l^k &= M_{pr}|\Psi^l \\ T_u^k &= M_{pr}|\Psi^u \end{aligned}$$

where $M_{pr}|\Psi^l$ and $M_{pr}|\Psi^u$ are the percentile values of sets Ψ^l and Ψ^u, respectively.

Step 2: Travel time records Φ are presented as a time series. The time window moves step by step on the time axis. Within each time window, first we look at the sample size S_z. If sample size is too small (i.e. one or two observations in one time window), mark the observations for visual check. Otherwise, we judge whether there are outliers in this time window or not by the rule:

$$\text{outliers exist in the time window} \begin{cases} no & \sigma \leq \sigma_{thr} \\ yes & \sigma > \sigma_{thr} \end{cases}$$

where σ is the standard deviation of the observation within the time window of interest, σ_{thr} is a threshold (critical) value to be determined.

Step 3: If $\sigma > \sigma_{thr}$, check all the observations whether they are in the confidence scope $\left[T_l^k, T_u^k\right]$. Those observations that fall out of the scope are treated as outliers with boolean value, to say 1, otherwise with 0. Then, move the time window to the next one, and continue with step 3.

Step 4: When the whole records are processed, for each observation i we can count how many times N_i the boolean value of this observation equal 1. This is because each observation will fall in time window more than once. For example, we have travel time records $\Phi = [m_1, m_2, m_3 ... m_n]$ and $T_w = 5$. The sixth record m_6 will be evaluated in the $2th$ to $6th$ time window, which the $2th$ and $6th$ time window consists of $[m_2, m_3, m_4, m_5, m_6]$ and $[m_6, m_7, m_8, m_9, m_{10}]$, respectively.

Step 5: Finally, the outliers are identified by the rule:

$$\text{observation } TT_i \text{ is outlier} \begin{cases} no & N_i \leq N_{thr} \\ yes & N_i > N_{thr} \end{cases}$$

where N_{thr} is a threshold value to be determined.

E.4.2 Algorithm parameters

In the above described algorithm, it can be seen that there are different parameters involved which should be estimated rather than determined based on intuition. These parameters are:

1. Time Window T_w

   It is obvious that a too large time window may cover different traffic conditions within this time window (high value of travel time records during congestion are not

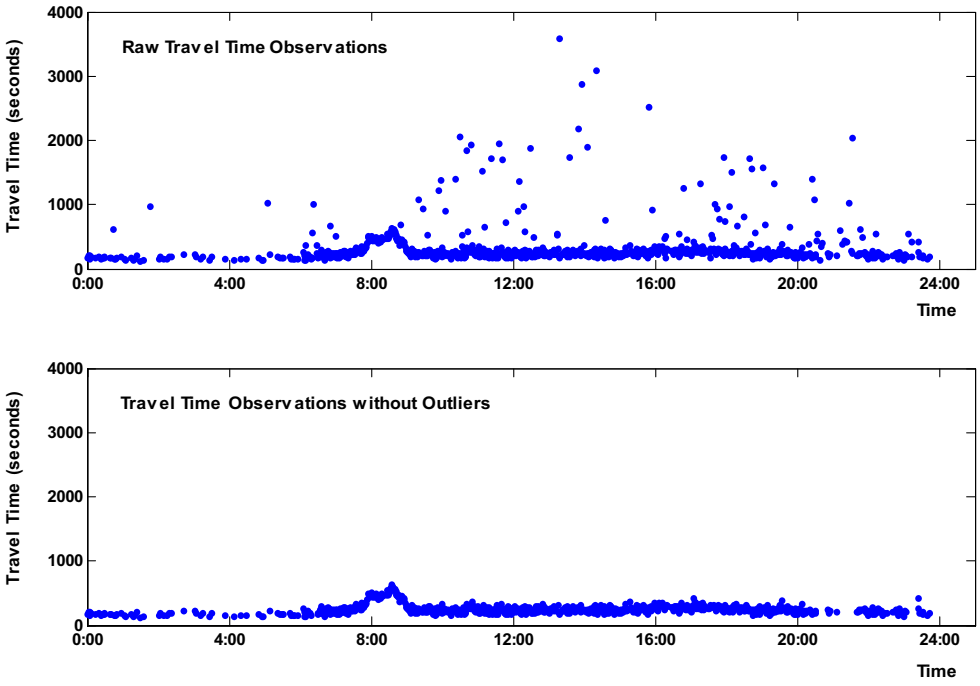


Figure E.2: Scatterplot of travel time observations with outliers and without outliers on November 17 2004. The observations are aggregated in time interval of 1 minute (data source from Regiolab-Delft).

outliers compared with those low value during free-flow conditions in the same time window), while too small time window may fail to provide sufficient observations. Thus, a balance between ability to react to rapid changes and to ensure sufficient data will yield an appropriate T_w.

2. Critical Standard Deviation σ_{thr}

   The critical standard deviation σ_{thr} is used to judge whether outliers exist in the time window of interest or not. Tentatively, we consider that large value of σ_{thr} will ignore outliers, while small value might misrecognize valid data as outliers.

3. Critical Count N_{thr}

   High value of N_{thr} means an observation is considered as a outlier by more count times. This might cause ignoring real outliers. On the other side, low value of N_{thr} might misrecognize valid data.

Performance on filtering outlier Figure E.2 shows two scatterplots of travel time observations with outliers and without outliers by the means of the new proposed algorithm on November 17, 2004. In the raw data, some observations even exceed 20 minutes which are not reasonable for traversing a 2km urban street in recurrent traffic conditions (no accident happened on that date). These outliers can be successfully identified in order to yield cleaned data for model calibration and validation.

Summary

With the advent of Advanced Traveler Information Systems (ATIS), short term travel time prediction is becoming increasingly important. Over the last few decades researches in this field have mostly concentrated on the applications to freeway facilities, while very few studies have focused on urban networks. This preference of researches for freeways can be identified from the literature overview. There are no reliable and applicable methods for urban short term travel time prediction. The intention of this dissertation is to make an attempt on this subject.

With the analysis of existing models, four clear impressions can be identified: (1) Among the existing works on travel time prediction, most of them are designed for urban segments, not for urban routes/arterials. The performance of those models at a large scale has not been evaluated. (2) None of those models presents how to predict future traffic conditions. This implies that the underlying assumption is the stationarity of future traffic conditions. This assumption is obviously hard to hold in real environments. (3) Most of those models were validated in a simulation environment. No evidence shows their performance on empirical data. (4) Due to the difficulties of collecting empirical data, practical investigations of the travel time variability for urban routes are very limited.

Based on the above analysis, we conduct the research to address short term travel time prediction for urban routes. The new proposed model has been evaluated with both simulation data and empirical data.

The main contribution of this dissertation is that a neural network based traffic flow model for urban route travel time prediction (Chapter 4) has been developed. The approach is a hybrid of data-driven and model-based approaches. The concept of a hybrid model for a neural network is applied for the first time in this dissertation.

In the transportation community, there is always a debate on selecting either model based or data driven approaches. Model based approaches can interpret their models with clear physical meanings. However, model based approaches are restricted by the limited knowledge of such complex urban traffic processes and the availability of measured data. Data driven approaches learn the mechanism of the urban traffic processes directly from measured data, liberating us from building sophisticate physical models. Moreover, data driven approaches are fast and easy to implement in practice.

Followed with the discussion above, in this dissertation, we choose a hybrid approach. The development of a neural network based traffic flow model for urban route travel time prediction was described in detail. An urban route can be divided into several segments, which are treated as the basic elements of the urban route. Correspondingly, modeling urban route traffic is decomposed into modeling urban segment traffic. Inspired with the

concept of the decomposition, a single segment model based on the Recurrent Neural Network is developed for modeling the traffic flows on a single urban signalized segment. The segment model, USEG, provides a generic (mathematical) structure. Urban route travel time prediction can be conducted by concatenating each individual segment model. Consequently, the traffic flows on an urban route are modeled by propagating from an upstream segment to a downstream segment.

In fact, the USEG is a kind of SSNN model, which enables the previous states to be temporally memorized in itself. With this mechanism, the USEG is able to predict travel times based on both current measurements and previous internal states. In this sense, the USEG operates like a macroscopic traffic flow model. In addition, the feedback (memory) mechanism in the SSNN allows the inputs to be fed at consecutive time instants sequentially. Compared to the FNNs, a clear advantage of the SSNN is that the selection of input time lag is not required. In addition, compared with augmenting all the spatially separated inputs in a single input layer, modeling an urban signalized route with separate models for each urban segment significantly reduces the number of weight parameters which need to be calibrated.

The outgoing traffic flows leaving from the upstream urban segment, calculated with the USEG, can be used as the inputs for the connecting downstream urban segment USEG. By concatenating all the USEGs which are comprised of an urban route of interest, the UROU is developed to propagate traffic flows through the route of interest. The ability of propagating traffic flows enables the UROU to predict travel times for any long route of interest.

Two main algorithms for training the USEG have been tested in this dissertation: batch training, in which parameter optimization is carried out with respect to the entire training data set simultaneously, and incremental training, where model parameters are updated after the presentation of each training example. The Bayesian regulated back propagation algorithm has been used for batch training, which provides a sound way to avoid over fitting. Based on extended Kalman filter (EKF), a new online-learning method has been selected for the incremental training. The new method is called the online-censored EKF method.

In order to fully test the performance of the proposed model and techniques employed in this research, we first choose to use synthetic data obtained from a microscopic traffic simulation tool, VISSIM. Three typical traffic conditions (slightly saturated, moderately saturated and seriously oversaturated conditions) have been generated to test this proposed model. Based on the sensitivity analysis of training the USEG, we choose a state space neural network with 4 hidden neurons in the following analysis.

The proposed model with the incremental training algorithm performs significantly worse than the batch trained model. All performance indicators (MARE, MRE and SRE) of the incremental trained model are approximately double in comparison to those of the batch trained model. In incremental training cases, the proposed model performs even worse than the baseline model when the prediction time ahead is equal or larger than 20 minutes.

In the simulation, the proposed model with the batch training algorithm is able to predict accurate travel time predictions up to 30 minutes of the prediction time ahead. The case of 30-minute prediction time ahead produces MARE of 14.8%, MRE of 4.2% and SRE

of 12.9%. These performance indicators illustrate that the proposed model outperforms the baseline model.

After being validated with simulation data (100% correct), the proposed model is applied in a real world environment, an urban road in Delft, in the Netherlands. To implement the model in practice, dealing with corrupted and missing data is an important task. Two data cleaning procedures have been proposed in order to obtain good quality data. A procedure of dealing with corrupted volume data collected by single loop detectors has been proposed in Chapter 6. In addition, a method to detect the outliers of travel time observations, and then fill in the empty gaps, has been developed in Appendix E.

After processing raw data, we took a close look into the variability of urban travel times. For the cases of the time window larger than 10 seconds, the 90th percentile of the variability of travel times increase significantly with the value of travel times. This shows that urban travel times are very variable. For this 2 km urban street, vehicles departed even within time difference of 10 seconds still had a high possibility of experiencing large variability of travel times.

For the real-time application, three strategies influence the performance of the proposed model: (1) the integration of traffic flow prediction, (2) the use of pre-processed data, and (3) the application of different training algorithms. Those strategies were evaluated with respect to the predictive performance. The former two strategies impose positive influences on the performance of the proposed model. The model with the batch training algorithm outperforms than the one with the incremental training algorithm. Overall, the results show that the predictive performance of the proposed model outperforms the baseline model when the prediction time ahead increases up to 10 minutes.

In conclusion, in this dissertation we present an accurate and robust model for short term urban travel time prediction. This research is the first attempt to combine a model based approach and data driven approach. The model has a generic structure. In that sense, it can be applied on any urban route equipped with traffic data collection systems (single loop detectors, license plate cameras and traffic signal timings). It also can be extended to easily include more influencing factors because of the nature of the flexible structure of the SSNN.

Samenvatting

Met de opkomst van geavanceerde systemen voor het informeren van reizigers (Advanced Traveler Information Systems, afgekort ATIS) neemt ook het belang van korte termijn reistijdvoorspelling toe. In de afgelopen decennia is er voornamelijk onderzoek hierna gedaan naar reistijdvoorsepllers voor autosnelwegen terwijl korte termijn reistijdvoorspellingen op stedelijke netwerken onderbelicht zijn gebleven. De sterke voorkeur voor autosnelweg applicaties blijkt uit het literatuuroverzicht. Daarin wordt duidelijk dat een betrouwbare en toepasbare methode ontbreekt voor het voorspellen van de korte termijn reistijd op stedelijke netwerken. Binnen deze dissertatie zal worden getracht hiervoor een oplossing te vinden.

Bij het bestuderen van bestaande modellen komen vier zaken duidelijk naar voren: (1) de meeste bestaande modellen zijn ontworpen voor het voorspellen van reistijd op het niveau van stedelijk netwerk segmenten en niet op het niveau van routes en doorgaande wegen. Hoe deze modellen presteren bij toepassing op deze grotere schaal is niet bekeken. (2) Geen van deze modellen laat zien hoe toekomstige verkeerscondities kunnen worden voorspeldt. Dit betekent dat er van stationaire verkeerscondities wordt uitgegaan. Deze veronderstelling is duidelijk ongegrond in reële situaties. (3) De meeste van deze modellen zijn alleen gevalideerd in een simulatieomgeving. Bewijs op basis van empirische data ontbreekt. (4) Doordat het verzamelen van empirische data vaak moeilijk is, zijn het aantal praktijk analyses op reistijdvariabiliteit op stedelijke routes zeer beperkt.

Gegeven de bovenstaande bevindingen, ontwikkelden wij in dit onderzoek korte termijn reistijdvoorstelling op stedelijke netwerken. Het nieuw ontwikkeld model is getest met zowel simulatie data als empirische data.

De voornaamste bijdrage van deze dissertatie is het ontwikkelen van een verkeersstroommodel voor stedelijke route reistijdvoorspelling gebaseerd op een neuraal netwerk (hoofdstuk 4). De aanpak is hybride in de zin dat het een koppeling vormt tussen een aanpak gebaseerd op data en op een model. Het concept van een dergelijk hybride neuraal netwerk model is voor het eerst toegepast in deze dissertatie.

Binnen de transportgemeenschap is er een voortdurend debat over het toepassen van modelgebaseerde of datagebaseerde aanpakken. Modelgebaseerde aanpakken kunnen onderbouwd worden vanuit de fysieke modelrelaties. Echter, modelgebaseerde aanpakken zijn beperkt door de beperkte inzichten in complexe, stedelijke verkeersprocessen en de beschikbaarheid van data. Datagebaseerde aanpakken verwerven de mechanismen in deze stedelijke verkeersprocessen direct vanuit de aangeleverde data waardoor het ontwerpen van geavanceerde, fysieke modellen niet meer nodig is. Daarbij zijn datagebaseerde aanpakken snel en eenvoudig toe te passen in de praktijk.

In lijn met bovenstaande discussie wordt in deze dissertatie gekozen voor een hybride aanpak. Het ontwikkelen van een neuraal netwerk gebaseerde verkeersstroommodel voor stedelijke route reistijdvoorspelling wordt in detail beschreven. Een stedelijke route kan worden opgedeeld in verschillende segmenten welke beschouwd worden als basiselementen van de stedelijke route. Op dezelfde wijze wordt het modelleren van stedelijk routeverkeer opgedeeld in het modelleren van stedelijk verkeer op elk van de segmenten. Geïspireerd door de notie van opdeling wordt een enkel segmentmodel gebaseerd op een Recurrent Neural Network specifiek ontwikkeld voor het modelleren van de verkeerstroom op een enkel stedelijke, geregeld segment. Het segmentmodel, USEG, zorgt voor een generieke (mathematische) structuur. Daardoor worden verkeersstromen op de stedelijke route gemodelleerd door het overbrengen van het verkeer van segmenten stroomopwaarts naar segmenten stroomafwaarts.

In feite is USEG een specifiek geval van het State Space Neural Network (SSNN) model waarin de voorgaande toestand tijdelijk wordt onthouden in de modelstructuur. Met dit mechanisme kan USEG reistijden voorspellen gebaseerd op zowel de huidige metingen als de voorgaande toestand. Wat dit betreft opereert USEG net als een macroscopisch verkeersstroommodel. Daarbij laat het feedbackmechanisme (geheugen) in het SSNN toe om invoer op opeenvolgende tijdstippen terug te voeren aan het model. In vergelijking met Feedforward Neural Networks heeft het SSNN het voordeel dat niet vooraf ingeschat hoeft te worden met welke historische invoerdats rekening gehouden moet worden. Daarbij vereist het modelleren van een stedelijke route met geregelde kruispunten met een afzonderlijk model voor elk stedelijk segment aanzienlijk minder gewichtsparameters welk gekalibreerd moeten worden vergeleken met een neuraal netwerkmodel een enkele invoer laag voor de hele route.

De uitstromende verkeersstroom van het stroomopwaartse, stedelijk segment, welk berekend wordt door een bepaalde USEG, geldt als invoer de USEG voor het aansluitend stedelijk segment stroomafwaarts. Door alle USEG's van een bepaalde stedelijke route aaneen te schakelen, wordt de UROU gevormd welk de verkeerstroom doorgeeft over de betreffende route. Doordat de UROU om kan gaan met verkeersvoortplanting kan de reistijd voorspeld worden van elke route van iedere lengte.

In deze dissertatie zijn twee veel gebruikte algoritmen getest voor het trainen van de USEG, namelijk batch training waarbij de parametersettings worden geoptimaliseerd op basis van de gehele dataset ineens, en incremental training waarbij de parametersettings achtereenvolgend worden geoptimaliseerd in verschillende opeenvolgende trainingen. Voor de batch training is het Bayesian regulated back propagation algoritme toegepast waardoor overfitting voorkomen wordt. Voor de incremental training is een nieuw ontwikkeld methode toegepast gebaseerd op extended Kalman filtering (EKF). Deze nieuwe methode wordt de online-censored EKF methode genoemd.

Om de prestaties van dit model en de technieken toegepast in dit onderzoek te testen, maken we in eerste instantie gebruik van modeldata van het microscopisch verkeersimulatiemodel VISSIM. Drie typische verkeerscondities (enigszins verzadigde, middelmatig verzadigde en aanzienlijk verzadigde condities) zijn gegenereerd om het ontwikkelde model te testen. Op basis van de uitgevoerde gevoeligheidsanalyse op het trainen van de USEG is de kezue gemaakt verder te werken met een State Space Neural Network met vier verborgen neuronen.

Het ontwikkeld model presteert aanzienlijk slechter wanneer getraind met het incremental training algoritme dan wanneer getraind met het batch training algoritme. De waarden van alle prestatie indicatoren (MARE, MRE en SRE) zijn ongeveer twee maal zo hoog voor incremental training als voor batch training. Het model getraind met het incremental training algoritme presteert zelfs slechter dan het basis referentie model met voorspellingen van (meer dan) 20 minuten vooruit.

In een simulatieomgeving in het gepresenteerde model met het batch training algoritme in staat om tot 30 minuten vooruit nauwkeurige reistijdvoorspellingen te doen. De voorspellingstijd van 30 minuten levert een MARE van 14.8%, MRE van 4.2% en SRE van 12.9%. Aan deze prestatie indicatoren is te zien dat het gepresenteerde model duidelijk beter presteert van het basis referentie model.

Nadat het model gevalideerd is op basis van simulatiedata (100% correct) is het gepresenteerde model toegepast in de praktijk op een stedelijke weg in Delft, Nederland. Bij het in de praktijk implementeren van het model spelt het omgaan met ongeldige en ontbrekende data een grote rol. Om goede kwaliteit data te verkrijgen, worden twee procedures voorgesteld voor het opschonen van de data. In hoofdstuk 6 wordt een procedure gepresenteerd voor omgaan met ongeldige intensiteitdata verzameld door enkelvoudige detectielussen. In appendix E wordt een methode gepresenteerd voor het detecteren en vervangen van extreme waarnemingen in reistijdmetingen.

Na het verwerken van de ruwe data hebben we gekeken naar de variabiliteit van stedelijke reistijden. Voor de gevallen waar het tijdsinterval groter is dan 10 seconden neemt de 90e percentiel in de reistijdvariabiliteit significant toe met de reistijd. Dit toont aan dat stedelijke reistijden zeer variëren. In het geval van deze 2 km lange stedelijke weg hadden zelfs voertuigen die binnen een tijdsinterval van 10 seconden vertrokken een aanzienlijke kans op een grote reistijdvariabiliteit.

Drie strategieën beïnvloeden het succes van het gepresenteerde model in online toepassingen: (1) het integreren van verkeersstroomvoorspellingen, (2) het gebruik van vooraf bewerkte data en (3) het toepassen van verschillende trainingsalgoritmen. Deze strategieën zijn getest ten aanzien van de kwaliteit van de voorspelling van de reistijd. De eerste twee strategieën hebben daarbij een positief effect. Waarbij het model met het batch training algoritme beter presteert dan het model met het incremental training algoritme. Alles in acht nemend, tonen de resultaten aan dat het gepresenteerde model beter reistijden kan voorspellen dan het basis referentie model wanneer reistijden tot 10 minuten vooruit worden voorspeld.

De conclusies die we kunnen trekken, zijn dat in deze dissertatie een nauwkeurig en robuust model is gepresenteerd voor korte termijn, stedelijke reistijdvoorspelling. Dit onderzoek is de eerste aanzet tot het combineren van een modelgebaseerde aanpak met een datagebaseerde aanpak. Het model heeft een generiek structuur. In die zin kan het toegepast worden op elk stedelijke route waar verkeersdata verzameling plaats vindt (bijvoorbeeld door enkelvoudige detectielussen, camera's met nummerplaat herkenning en verkeerslicht cyclustijden). Daarnaast kan het model eenvoudig uitgebreid worden om meer beïnvloedingsfactoren mee te nemen dankzij de flexibele structuur van het SSNN.

(Dutch tanslation by Adam Pel)

About the author

.

Hao Liu was born on 22 January, 1977 at Dechang, Sichuan Province, China. He started his high-level education in 1994 at Transportation School of Southeast University. He received his Bachelor degree in 1998. Afterwards, he was recommended to continue a M.Sc. study at the Research Institute of Highway (RIOH), Ministry of Communications. His M.Sc. study was divided into two phases. From 1998 to 1999, he studied the fundamental master courses at Southeast University. When he finished all required courses, he moved to RIOH and started the second phase of study. During the second phase, he completed his master thesis and obtained a M.Sc. degree in 2002. He then started to work for Chinese National Intelligent Transport Systems Research Center of Engineering and Technology (ITSC), which is a sub-section of RIOH.

In November 2003, he was selected by RIOH to start his Ph.D.-study at the Transportation and Planning Section of the Delft University of Technology (TU Delft), the Netherlands. Both RIOH and TU Delft provide financial support for his study. During his stay at TU Delft he focused his research mainly on travel time prediction for urban networks, producing several papers and presenting his work at various international conferences. One of those papers was awarded with the prize "Best Scientific Paper Award" at the 2006 TRAIL congress in Rotterdam. After he finishes his Ph.D. study, he will continue his research career at ITSC.

TRAIL Thesis Series

.

A series of The Netherlands TRAIL Research School for theses on transport, infrastructure and logistics. See for a complete overview our website: www.rsTRAIL.nl.

Liu, H., *Travel Time Prediction for Urban Networks*, T2008/12, TRAIL Thesis Series, the Netherlands

Kaa, E.J. van de, *Extended Prospect Theory. Findings on Choice Behaviour from Economics and the Behavioural Sciences and their Relevance for Travel Behaviour*, T2008/11, TRAIL Thesis Series, the Netherlands

Nijland, H., *Theory and Practice of the Assessment and Valuation of Noise from Roads and Railroads in Europe*, T2008/10, TRAIL Thesis Series, the Netherlands

Annema, J.A., *The Practice of Forward-Looking Transport Policy Assessment Studies*, T2008/9, TRAIL Thesis Series, the Netherlands

Ossen, S.J.L., *Theory and Empirics of Longitudinal Driving Behavior*, T2008/8, TRAIL Thesis Series, the Netherlands

Tu, H., *Monitoring Travel Time Reliability on Freeways*, T2008/7, April 2008, TRAIL Thesis Series, the Netherlands

D'Ariano, A., *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*, T2008/6, April 2008, TRAIL Thesis Series, the Netherlands

Quak, H.J., *Sustainability of Urban Freight Transport. Retail Distribution and Local Regulations in Cities*, T2008/5, March 2008, TRAIL Thesis Series, The Netherlands

Hegeman, G., *Assisted Overtaking. An assessment of overtaking on two-lane rural roads*, T2008/4, February 2008, TRAIL Thesis Series, the Netherlands

Katwijk, R.T. van, *Multi-Agent Look-ahead Traffic Adaptive Control*, T2008/3, January 2008, TRAIL Thesis Series, the Netherlands

Argiolu. R., *Office Location Choice Behaviour and Intelligent Transport Systems*, T2008/2, January 2008, TRAIL Thesis Series, the Netherlands

Houtenbos, M., *Expecting the Unexpected. A study of interactive driving behaviour at intersections*, T2008/1, January 2008, TRAIL Thesis Series, the Netherlands

Negenborn, R.R., *Multi-Agent Model Predictive Control with Applications to Power Networks*, T2007/14, December 2007, TRAIL Thesis Series, the Netherlands

Nederveen, A.A.J., *Ruimtelijke Inpassing van Lijninfrastructuur. Een onderzoek naar de geschiktheid van inspraakreacties voor het ontwerpen van lijninfrastructuur*, T2007/13, December 2007, TRAIL Thesis Series, the Netherlands

Nuttall, A.J.G., *Design Aspects of Multiple Driven Belt Conveyors*, T2007/12, November 2007, TRAIL Thesis Series, the Netherlands

Gietelink, O.J., *Design and Validation of Advanced Driver Assistance Systems*, T2007/11, November 2007, TRAIL Thesis Series, the Netherlands

Driel, C.J.G. van, *Driver Support in Congestion: an assessment of user needs and impacts on driver and traffic flow*, T2007/10, November 2007, TRAIL Thesis Series, the Netherlands

Warffemius, P.M.J., *Modeling the Clustering of Distribution Centers around Amsterdam Airport Schiphol. Location Endowments, Economies of Agglomeration, Locked-in Logistics and Policy Implications*, T2007/9, September 2007, TRAIL Thesis Series, the Netherlands

Joksimovic, D., *Dynamic Bi-Level Optimal Toll Design Approach for Dynamic Traffic Networks*, T2007/8, September 2007, TRAIL Thesis Series, the Netherlands

Vlist, P. van der, *Synchronizing the retail supply chain*, T2007/7, June 2007, TRAIL Thesis Series, the Netherlands

Fiorenzo-Catalano, M.S., *Choice Set Generation in Multi-Modal Networks*, T2007/6, June 2007, TRAIL Thesis Series, the Netherlands

Bok, M.A. de, *Infrastructure and Firm Dynamics: A micro-simulation approach*, T2007/5, May 2007, TRAIL Thesis Series, the Netherlands

Zondag, B., *Joined Modeling of Land-use, Transport and Economy*, T2007/4, April 2007, TRAIL Thesis Series, the Netherlands

Weijermars, W.A.M., *Analysis of Urban Traffic Patterns Using Clustering*, T2007/3, April 2007, TRAIL Thesis Series, the Netherlands

Chorus, C., *Traveler Response to Information*, T2007/2, February 2007, TRAIL Thesis Series, the Netherlands

Miska, M., *Microscopic Online Simulation for Real time Traffic Management*, T2007/1, January 2007, TRAIL Thesis Series, the Netherlands

Makoriwa, C., *Performance of Traffic Networks. A mosaic of measures*, T2006/10, December 2006, TRAIL Thesis Series, the Netherlands

Feijter, R. de, *Controlling High Speed Automated Transport Network Operations*, T2006/9, December 2006, TRAIL Thesis Series, the Netherlands

Huisken, G., *Inter-Urban Short-Term Traffic Congestion Prediction*, T2006/8, December 2006, TRAIL Thesis Series, the Netherlands

Viti, F., *The Dynamics and the Uncertainty of Delays at Signals*, T2006/7, November 2006, TRAIL Thesis Series, the Netherlands