Reliable Travel Time Prediction for Freeways

J.W.C. van Lint

May 3, 2004

Cover illustration: Hans van Lint
Cover design: Joke Herstel

Reliable Travel Time Prediction for Freeways

Bridging Artificial Neural Networks and Traffic Flow Theory

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 maandag 7 juni 2004 om 15:30

door

Johan Willem Christiaan Van Lint

civiel ingenieur

geboren te Delft

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
Dr. Ir. S.P. Hoogendoorn	Technische Universiteit Delft, toegevoegd promotor
Prof. Dr. Ir. P.H.L. Bovy	Technische Universiteit Delft
Prof. S. Kikuchi, Ph.D., P.E.	University of Delaware, VS
Prof. Dr.-Ing. W. Brilon	Ruhr-University Bochum, Duitsland
Prof. L.R. Rilett, Ph.D., P.E.	Texas A&M University, VS
Dr. T.M. Heskes	Katholieke Universiteit Nijmegen

This dissertation thesis is funded by the Regiolab Delft project, a joint research program of the Delft University of Technology, The Dutch Ministry of Transport, Public Works and Water Management, The Municipality of Delft, The Province of South-Holland, the TRAIL Research School, Connekt, Vialis and Siemens.

Trail Thesis Series no. T2004/3, The Netherlands TRAIL Research School

This thesis is the result of a Ph.D. study carried out from 2000 to 2004 at Delft University of Technology, Faculty of Civil Engineering and Geosciences, Transportation and Planning Section.

Published and distributed by:

TRAIL Research School, P.O. Box 5017, 2600 GA Delft, (t) +31 15 278 60 46, (f) +31 15 278 43 33, (e) info@rsTRAIL.nl, (i) www.rsTRAIL.nl

ISBN : 90-5584-054-8

Printed in The Netherlands

"Prediction is difficult, especially the future"
- Niels Bohr

Preface

When I finished my masters at the faculty of Civil Engineering at the Delft University of Technology in 1997, I solemnly swore never to set foot in that building again. After job-hopping for a number of years I came to realize that scientific research in fact offered exactly those ingredients I had been looking for all along, that is, room for creativity, a fair degree of independence, time to dig in when required, but also enough pressure and deadlines to prevent one from dozing off or drowning in good ideas. And so I started my Ph.D. research part time in May 2000 and continued full time from March 2001 onwards. Three years and a bit later, this dissertation thesis marks the end of one of the most enjoyable but certainly also toughest periods of my life. And now it opens up a brand new period in which I am in the fortunate position to further pursuit my scientific interests and love for teaching and supervising.

I am deeply indebted to every one who has supported (and tolerated) me in the past four years. First of all, I thank professor Henk van Zuylen for giving me the chance to start this endeavor in the first place (science first, funding later) and for his support and invaluable input. Secondly, I would like to express my thanks and gratitude to my daily supervisor Serge Hoogendoorn for his moral, mental, scientific, mathematical and personal support and at times sheer genius, without which this book simply would not have been possible. Special thanks also to Nanne van der Zijpp, who 'discovered' the PLSB trajectory method in my matlab code and collaborated with me on two papers, which led to large parts of chapter 3. Furthermore I thank professor Piet Bovy, professor Werner Brilon, professor Larry Rilett, professor Shinya Kikuchi and Tom Heskes for being part of my promotion committee and for their comments and criticism of review. Similarly, cheers and many thanks to all those colleagues and friends who offered their ideas, thoughts and humor that helped me grow and produce the results I needed. I thank (in random order) Francesco Viti, Karel Lindveld, Mark Miska, Theo Muller, Peter Knoppers, Henk Taale and all the others for their time, support and off course table tennis during the breaks. Furthermore, I owe a big word of thanks to all the supporting staff of the Planning and Transportation Department, especially Nicole Fontein, Bianca Kerkhoff, Cees Landman, and Peter van der Vlist, for their organizational, mental and digital support. I also thank the Traffic Research Department (in Dutch: the AVV) of the Dutch Ministry of Transport, Public Works and Water Management, and especially Frans Middelham, Rob van der Voort and Hans Remeijn for their support and effort to bridge the gap between science and practice and of course

’their’ money. The same thank you goes to the other participants of the Regiolab Delft project and particularly the TRAIL Research School (Arjen van Binsbergen and his staff).

Finally, many thanks, hugs and kisses go out to my friends and family, who were there for me when I needed them. And as they know I needed them for much more than just this piece of paper. Hoping not to do injustice to all the others I especially thank Wieteke ten Horn (simply one of the best friends and proofreaders a person can have), Anne Koch (seems the sun will shine for both of us), Ernst Komen (maximum respect!), Emiel Zwaard (a real friend), Roy Spanjers (keep overestimating me), Sonia Van Bost (one day we’ll look back at this and it’ll all seem funny), all my musical friends of For Absent Friends, Every Dog and Krenny Lavitz who helped keep me sane and creative and off course my mum and dad Beb and Joop van Lint (technique does after all run in the family), and my sister Dorine and her posse Rene, Tim and Luc Zegers.

Last but not least, I dedicate this thesis to the loving memory of my sister Joke van Lint, the one person who has had - albeit unwillingly - the most profound influence on my life to this very day. Wherever you are, it was ultimately for the better. Cheers!

Hans van Lint
May, 2004

Contents

Notation

This section lists the symbols used throughout this dissertation thesis.

Abbreviations and Acronyms

SSNN	:	State Space Neural Network
ANN	:	artificial neural network
FNN	:	Feed-forward Neural Network
RNN	:	Recurrent Neural Network
TDNN	:	Time Delayed Neural Network
AR(I)MA	:	Auto Regressive (Integrated) Moving Average
PCSB	:	Piece-wise Constant Speed Based (trajectory method)
PLSB	:	Piece-wise Linear Speed Based (trajectory method)
MONICA	:	MONItoring CAsco (inductive loop traffic data collection system)
LM	:	Levenberg-Marquardt (ANN training algorithm)
BR	:	Bayesian Regularization
ATIS	:	Advanced Traffic Information System
VMS	:	Variable Message Sign
FOSIM	:	Freeway Operations SImulation Model (microscopic traffic simulator)
AVI	:	Automatic Vehicle Identification
CoV	:	Conservation of Vehicles
LWR	:	Lighthill, Witham and Richards (first order traffic flow model)
FD	:	Fundamental Diagram (of traffic flow)
pdf	:	probability density function
cdf	:	cumulative density function

Indices and Variables

p	:	departure time period (both as index and as independent variable)
i	:	index for individual vehicle
t	:	time or departure time (both as index and as independent variable)
x	:	space (both as index and as independent variable)
r	:	index for route
k	:	index for section, also used to indicate time lag
m	:	index for section or hidden neuron
τ	:	travel time
v	:	speed
$\widehat{var}$	:	approximation of the variable "var"

Travel Time Estimation

τ_{rti}, τ_{rpi}	:	travel time of individual i on route r with departure time (period) t (p)
$\tau_{ri}(t), \tau_{ri}(p)$	:	travel time of individual i as a function of departure time (period) t (p)
$\tau(p), \tau_r(p)$	:	mean travel time (on route r) as a function of departure time period p
$x_i(t)$	:	location of individual i as a function of time
$v_i(t)$	:	speed of individual i as a function of time
$a_i(t)$	:	acceleration of individual i as a function of time
$t_i(x)$	:	time of individual i as a function of space
$f_L(v)$	:	local (at fixed x) probability distribution of speeds
$f_M(v)$	:	instantaneous (at fixed t) probability distribution of speeds
u_M	:	(arithmetic) space mean speed (at fixed t)
w	:	slowness (mean travel time per unit space)
u_L	:	(arithmetic) time mean speed (at fixed x)
$\widetilde{u}_L$	:	(harmonic) time mean speed (at fixed x)
$q, q(t, x)$	:	vehicular traffic flow (at a specific time instant, location)
$\rho, \rho(t, x)$	:	vehicular density (at a specific time instant, location)
ρ_L	:	local density
ρ_L^{crit}	:	critical local density
σ_M^2	:	instantaneous speed variance
σ_L^2	:	local speed variance
O	:	occupancy
$\theta(k)$	:	autocorrelation coefficient for time lag k
$\vartheta^2(k)$	:	auto-covariance for time lag k
T_p	:	length of period p (in units time)
T	:	length of prediction horizon

State Space Neural Networks

An important note beforehand pertains to the meaning of the independent variables of time. For example, in chapter 5 and appendices B, and C, $\mathbf{u}(t)$ depicts the vector of all inputs from time period $[t-1, t]$. In all other chapters this time period (the last known time period) is depicted by $p-1$. As a result the expressions $\mathbf{u}(t)$ and $\mathbf{u}(p-1)$ refer to exactly the same thing, that is, the input vector from the last known time period.

T	:	look-back interval
$\mathbf{u}(t)$	:	vector of all inputs from time period $[t-1, t]$
ψ	:	vector of all weights (i.e. parameters)
H	:	set of all (non-specified) modelling assumptions
$\mathbf{x}(t)$	:	vector of (internal) states (i.e. hidden layer outputs)
$y(t)$	:	SSNN output, SSNN function is denoted by $y(t) = G(\mathbf{u}(p-1), \psi)$
$\mathbf{V}$	:	vector of all output layer bias and weights
$\mathbf{w}$	:	vector of all hidden layer bias and weights
$z, \mathbf{z}$	:	general (vector of) input(s) to a function
$\phi(z)$	:	logistic transfer function
$\Phi(\mathbf{z})$	:	vectorized logistic function
M	:	number of hidden neurons
P	:	number of time periods in data sample
Q	:	total number of parameters
$o(t)$	:	target value (i.e. measured or offline estimated travel time)
$\mathbf{J}$	:	Jacobian matrix of output errors with respect to ψ
$\mathbf{H}$	:	Hessian matrix of SSNN performance with respect to ψ
α, β	:	regularization parameters for weights and output errors
$S_m, S_m(t)$	:	relevance of hidden neuron m (at time t)
$S_m^C, S_m^C(t)$	:	relevance of context neuron m (at time t)
$S_n^U, S_n^U(t)$	:	relevance of input signal n (at time t)

Confidence and Prediction Intervals

$\sigma_{\psi}^2(p)$	:	variance due to uncertainty in SSNN parameters in period p
$\sigma_{\tau}^2(p)$	:	variance due to the distribution of (actual) travel times in period p
$\sigma_P^2(p)$	:	variance due to PLSB estimation errors in period p
$\mathbf{g}(p)$	:	SSNN output sensitivity ($\mathbf{g}(p) = \frac{d}{d\psi} y(p)$, i.e. first derivative of output with respect to weights) in period p

Chapter 1

Introduction

The travel time a traveller experiences when making a trip from A to B is not just the result of his or her own travel choices (destination, mode, route, speed), but also of the choices of many other travellers, not necessarily only those travelling from A to B. Moreover, a substantial component of driver behavior may not be classified as rational (and to a degree predictable) choice behavior, but rather a product of the different characteristics of individual drivers, for example attention level, drive style, risk assessment, etc. and their vehicles, such as acceleration and deceleration capabilities. Finally, travel time between A and B is also determined by processes completely beyond the control of individual or groups of drivers or even the organization responsible for the road facility, such as weather, calamities, incidents and accidents, and so on. Travel times hence are the result of all these processes. Since in our view it is impossible to predict the behaviors (both rational and irrational) of all individual drivers in a road network and all the external circumstances that may affect their travel times, in this dissertation thesis models are sought that deduce general relationships between observable traffic processes and the travel times. Although in the past 5 decades a rich body of research has been developed, in terms of (traffic) theory and models and driver behavior, *predicting* traffic conditions, for example in terms of travel times, is still a very complex and challenging problem.

In this introductory chapter we outline the context and background of the travel time prediction problem in general and motivate why it is a relevant problem to tackle. Next, we present the main objectives of this dissertation thesis and narrow down the scope of the research presented. Particularly, this dissertation thesis concentrates on methods and models for short term travel time prediction on freeways. Subsequently, we explain the research approach, after which the main scientific and practical contributions of this dissertation thesis are reviewed. The final part of this introduction then briefly outlines which subjects are covered in each chapter of this thesis.

1.1 Context and Background

There is an increasing need for advanced traffic information systems (ATIS) that can provide travellers and traffic managers with accurate and reliable real-time traffic information (Abdel et al. 1997), (Van Lint et al. 2000). The assumption underlying the usefulness of traffic information in traffic systems is that individual road users are rational decision makers (homo economicus) who base their choices (e.g. path and departure time) on minimizing their expected costs (in terms of for example travel time and travel time reliability), subject to their personal preferences and attitudes and their knowledge and perception of the traffic system. Thus, providing travellers with traffic information allows them to make more informed decisions, yielding not only cost-benefits for the individual, but potentially also more stable and less congested traffic conditions for all road users. But even if no beneficial effects in terms of cost or time savings result from the application of ATIS, traffic information at least reduces uncertainty and increases comfort for most drivers (Van Berkum & Van der Mede 1993). The potentially beneficial effects of ATIS have been studied extensively in the past decade (e.g. (Khattak, Schofer & Koppelman 1995), (Khattak, Yim & Stalker 1995), (Arnott et al. 1991), and (Mahmassani & Liu 1999)). Although these studies clearly show the heterogeneity of possible responses to ATIS among different groups of drivers (e.g. commuters, non-commuters) under different (traffic) circumstances, they generally emphasize two things. The first is that for traffic information to have beneficial effects, it should be based on *predictions* rather than on current or past traffic conditions (Chen et al. 1999). Note that in these predictions, user response to the information should also be taken into account. From a system point-of-view, predictive information is also preferable to *current* information. Hoogendoorn (Hoogendoorn 1997) shows how provision of current traffic conditions instead of predicted information may lead to oscillatory (choice) behavior causing deterioration of traffic conditions rather than improvement. Secondly, the *reliability* of traffic information greatly influences driver response (Polak & Oladeinde 2000), (Mahmassani & Liu 1999), (Van Berkum & Van der Mede 1993). In the Oxford Dictionary 'reliable' is defined as *"consistently good in quality or performance, and able to be trusted"*, and reliability as *"the quality of being reliable"*. In literature, however, reliability is a term which has many different meanings depending on application and context (see e.g. (Bonsall 2000)).

1.1.1 Criteria for successful travel time prediction models for ATIS

To clarify the discussion we propose the following qualitative criteria for traffic information to have the hypothesized beneficial effects on collective traffic operations. A distinction is made between criteria relating to the traffic information itself and criteria relating to the underlying model generating the traffic information. Secondly, a distinction is made between subjective and objective criteria. The first category of criteria reflects perceptions of travellers with regard to traffic information, while the second

reflects criteria for the (independent) observer, for example a researcher or a traffic manager.

Unambiguity of Information Drivers should be able to understand and (to the degree this is possible) unambiguously interpret the information. We argue that this implies that travel time is the most appropriate choice for traffic information, instead of for example queue lengths. A traffic jam of 2 kilometer due to a major accident may cause a delay of 45 minutes, while a regular traffic jam of 5 kilometers may only incur an extra 15 minutes of travel time. Note, however, that how information is interpreted is subject to personal skills, experience and perception (Van Berkum & Van der Mede 1993). Although beyond the scope of this dissertation thesis, in this context it is crucial *how (travel time) information is presented to travellers*. In this thesis we predict mean travel time and provide error bars on that prediction. Paradoxically, if drivers are provided with an estimate of the uncertainty in traffic information (e.g. the error bars), this may unwillingly complicate the decision making process (Bonsall 2000), increase uncertainty (especially if drivers were apriori unaware of the uncertainty in the information provided), decrease driving comfort and deteriorate traffic flow operations. The point is that inherently, unambiguity is a subjective criterion, which is driver and situation specific, and that user response to (and hence the effectiveness of) ATIS strongly depends on it.

Subjective Validity of Information Given drivers understand the information (say travel time on route r for departure time p), the information provided should comply with the drivers own experiences (for example the travel time they believe is to be expected on route r). In (Van Berkum & Van der Mede 1993) this is referred to as the net value of the information, which is also subject to personal characteristics and perceptions of drivers. Although very relevant, this criterion is inherently very difficult to quantify, since it depends on individual driver characteristics, which are unobserved (at least in real-time).

Objective Validity of Information and Model The information (expected travel time on route r for departure time p) should also objectively comply with what actually occurred (the actual travel time on route r for departure time p). This criterion is quantifiable given that travel times are actually measured (or estimated). However, the outcomes of a model may be objectively valid but not necessarily perceived as valid.

Accuracy of Model Related to the previous two points, the difference between what actually happened (or what was perceived) and the information (in this case travel time) should be as small as possible, which is subject to location and application specific circumstances[1]. Roughly, model output errors can be categorized

[1] On a journey of three hours a 5 minute travel time prediction error is negligeable, however, for a trip of 10 minutes a model making a five minute error would be considered very inaccurate.

into two types, that is, structural errors (bias) and random errors (variance). Put simply, an accurate model makes small (quantitative) mistakes, in terms of both bias and variance. This, in many cases, is the sole criterion with which travel time prediction models are evaluated (see also section 1.2).

Robustness of Model The model producing the information should be able to deal with different (traffic) conditions (free flowing, congested, incidents, the holiday migration, etcetera). Also, if the data to a model is corrupt, which is a common problem in real-time traffic data collection systems as we will show in this thesis, the model should still be able produce (reasonable) outcomes (which could even be a message indicating something is wrong). Note that in the remainder of this dissertation thesis data corruption is referred to as *input failure.* Robustness is a quality which is difficult to assess in the absolute sense, since there are most probably specific circumstances in which a model eventually fails (an earthquake or terrorist attack, or less extreme events such as hardware failure, unforeseen or extraordinary traffic demand patterns, etc.).

Adaptivity of Model Traffic processes are characterized by constant change, due to (structural) changes in both traffic demand patterns as wel as traffic supply characteristics. The model should be able to track these changes and adapt accordingly to preserve its validity. Structural changes may for example be due to behavioral changes (compare drive style and skills in the late sixties with those in the 21st century) but also to technological advances (Intelligent Driver Assistance Systems, GPS) and changes in the infrastructure (safer and better infrastructure).

Reliability of Information and Model Finally, we interpret reliability here as "the mother of all qualities", that is, a reliable model is robust, valid, accurate and adaptive. Reliable traffic information is produced by reliable models, and is valid and presented understandably for most travellers.

The conclusion is, that for a traffic information service (for example a VMS panel) to produce the desired beneficial collective effects, it should be based on a reliable and hence adaptive, robust, accurate and valid travel time prediction model. Note that the criteria here can not be interpreted in absolute terms, but rather in terms of probabilities. An example of a probabilistic criterion could be that a model should be robust with respect to at least 20% random data failure in 95% of the cases.

1.1.2 Synthesis and implications of criteria for ATIS

In Fig 1.1 an abstract scheme of the concepts of reliability, robustness, (objective) predictive accuracy and (objective) validity and adaptivity is presented. A reliable travel time prediction model, by definition is also robust, accurate and valid (it produces results that comply with what actually occurred) and adaptive. Robustness does not

necessarily encompass accuracy and validity. A model could for example be able to deal with all sorts of data problems but consistently produce invalid or inaccurate results. Finally, to a degree, the reliability of our travel time prediction models depends on the reliability of the real world systems (a traffic network and the people that use it) that is modelled. An example of unreliability of real world systems is in errors made by operators of a traffic control centre. Even a model that is objectively robust, valid and accurate would produce unreliable results in case it is fed with data from "the wrong" traffic data collection system, due to erroneous operations of the person responsible for installing the model.

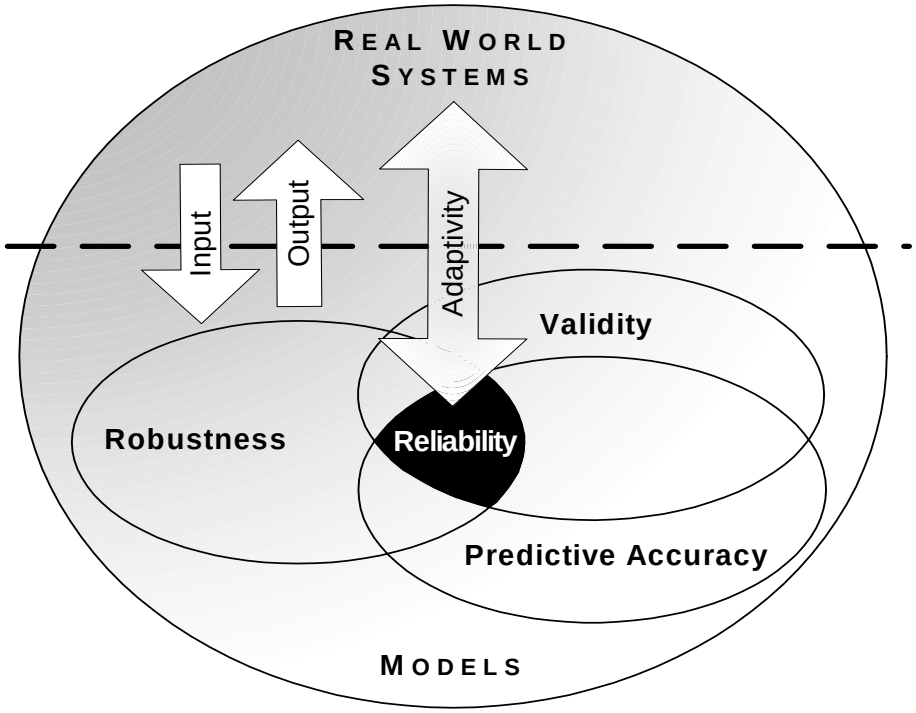


Figure 1.1: Abstract scheme of the concepts of reliability, robustness, predictive accuracy, validity and adaptivity.

Let us conclude this discussion on criteria for successful ATIS with two issues that are not quantitatively dealt with in this thesis, but are nonetheless closely related to the criteria discussed here. The first is that in practice what is being aimed at is the *acceptable degree of reliability (robustness, accuracy and validity)*. For every ATIS application this strongly depends on various situations, e.g., distance being covered, the degree of uncertainty involved, for example due to weather and incidents / accidents, which are beyond the limits of predictability. The general public would most likely understand the difficulty of predicting travel time so that they would accept the wider bands of the predicted time depending on the circumstances. Travel time information is most needed at the time of unusual events, e.g., bad weather, and incidents / accidents. Ironically, these are the times when it is most difficult to predict travel time. From a

travellers' standpoint, the reliability requirement also depends on characteristics of his or her trip (purpose, traveller attitude, travel decision options, etc.).

Secondly, setting criteria for ATIS systems is meaningful in practice *only* if the institutions or organizations deploying these systems take responsibility and accountability for the quality (measured by these criteria) of the information disseminated through these systems. Personal experience[2] dictates that traffic information usually is provided "as is". In policy documents often many claims are made on the value of traffic information, and the inherent quality (reliability, etc.) required, but rarely on accountability or responsibility issues. In case attention *is* paid to these issues[3] the common approach is to provide very limited warranties on the information provided, and few specifics on the legal procedures a consumer (a traveller or for example a commercial service provider) could follow in case of non-fulfilment. Moreover, if based on (objective and subjective) evaluation ATIS are found performing well under expectations, the usual approach is to (temporarily) shut these systems down and - in the worst case - hold the party responsible that built it in the first place. Either way, end users (tax payers, travellers, third party information service providers) are (kept) oblivious in all of this. Under these conditions, the qualitative and quantitative criteria listed above are primarily of academic interest!

1.2 Research Objectives and Scope

1.2.1 Research objectives

The main objective of this dissertation thesis is to develop models and methods that can produce reliable, that is adaptive, robust, accurate and valid travel time predictions. These four criteria are the set of objective criteria from the 6 listed in the previous section. Although for beneficial effects to result from traffic information the two subjective criteria (unambiguity and subjective validity) are of seminal importance, the travel time prediction models in this thesis are assessed on the basis of objective criteria only. In general, quantification of these criteria is location and application-specific, which makes it difficult to apriori set quantitative objectives or targets for the methods and models we derive in this research (e.g. a maximum predictive error of 3% or validity in at least 90% of the cases) . In some cases it is even impossible to objectively quantify a criterion, for example whether or not a model is robust under all prevailing traffic conditions (all conditions can never be observed). In many studies that involve travel time, speed or flow prediction (e.g. (Park & Rilett 1999), (D'Angelo et al. 1999), (Williams 2001), (Van Grol et al. 1999), (Park & Rilett 1998)), the emphasis is on predictive performance, that is, predictive accuracy. As a general rule, the

[2] The author was technical project coordinator for the Rotterdam Regional Traffic Information Center (RegioTIC Rotterdam) in the Netherlands in the period 2000-2001

[3] As does for example by the National Traffic Information Center (TIC-NL) of the Dutch Ministry of Transport, Public Works and Watermanagement.

models presented in these studies outperform other (less sophisticated) models on a particular location and test data set. Although we fully agree on the importance of predictive performance as (objective) criterion for model evaluation and comparison, we also emphasize that the most accurate model may not necessarily be the most robust or reliable model, and hence not the "best" model for a traffic information service.

Therefore, as a bottom line, the target for the models in this thesis is to offer improvements (or at least comparable results) on the four objective criteria mentioned here, with respect to the models that are currently used. To this end, in chapter 5, the predictive accuracy of our model on simulated data is compared to a naive (but in practice very often used) travel time prediction method, while in chapter 8, predictive accuracy of our model on real data is compared with results of various travel time prediction models reported in literature, using a common performance indicator. Robustness and reliability are dealt with extensively in chapters 6 and 7 respectively, and also in chapter 8. Adaptivity is (in our view) a fundamental property is of the type of (data-driven) models we use (see chapter 5).

A second objective in this dissertation thesis is to develop travel time prediction models that are *general*, and not-location-specific, at least in terms of their mathematical structure and the overall input-output relationships. For example, a freeway travel time prediction model should be applicable on different freeway routes, with different geometrical properties (number of lanes, the locations of exit and egress ramps or the spacing of the detection equipment that is installed). The reason is that - as mentioned above - the reliability of a model also depends on the reliability of the real world, and in particular of the people and systems that operate these models. A model that requires specific design (model and input selection) for every location is not likely to be successfully deployed on a large scale, for example as a travel time prediction model for VMS panels throughout a freeway network.

1.2.2 Research scope

We will, however, limit the scope of our research efforts. First of all, this dissertation thesis will address travel time prediction on uninterrupted roadway facilities such as *freeways*. We adhere to the definition for freeways given in the Highway Capacity Manual 2000: *"A Freeway is a multilane, divided highway with a minimum of two lanes for the exclusive use of (motorized) traffic in each direction and full control of access without traffic interruption"* (Transportation Research Board 2000). This does not imply that the methods and models presented here could not be used for predicting travel time for other types of infrastructure, such as urban (traffic light controlled) motorway facilities. In chapter 9 we briefly discuss some of the fundamental differences between freeway and urban traffic and outline which concepts developed in this thesis could be used for urban travel time prediction and which not.

Secondly, this thesis will focus on *short term* freeway travel time prediction. As chapter 2 will outline in detail, there are a number of fundamental differences between

short term (i.e. predicting the travel time of vehicles departing now or in the near-future) and *long term* (e.g. predicting travel time of vehicles departing tomorrow or next week, month or year) travel time prediction. These differences pertain to both the nature (the dynamics) of the problem, the requirements and constraints they pose on models (e.g. on the input), as well as to their application context (real-time display on VMSs on freeways versus long term planning application).

Thirdly, this dissertation focusses on a particular class of models to tackle the travel time prediction problem, that is inductive (*data-driven*) models, and particular *recurrent neural networks*. This choice is motivated by the complexity of the travel time prediction problem and will be explained in section 1.3.2 and in more detail in chapter 4. Since the focus is on *data driven* models, this dissertation thesis focusses on travel time prediction models for freeways, *given sufficient data are available* on that particular freeway. This implies that some traffic data collection system is installed. Historically, most in practice deployed traffic data collection systems consist of local detection equipment (inductive loops, pneumatic tubes), resulting in local aggregate characteristics (flows, local mean speeds) of traffic streams. Since these systems do not measure travel times directly, a separate chapter (3) is devoted to so-called offline travel time estimation techniques that enable translation of for example local speeds into (section or route) travel times. Note that systems which *do* measure travel times, should record travel times of vehicles with their associated departure time period, rather than the time period in which the measurement becomes available (which is the arrival time period of a vehicle). Counter-intuitively, travel time measurement systems are not necessarily the ideal data collection system for travel time prediction systems, since current measurements (realized travel times) in fact reflect past traffic conditions rather than current, especially on longer routes.

Thus, the scope of this dissertation thesis is the development of a new data-driven short term freeway travel time prediction model. The objectives are this model is reliable, that is accurate, valid, robust (to input failure) and adaptive. This implies we must present quantitative tools to express and assess the reliability of that model, next to measures that assess predictive accuracy and validity and robustness.

1.3 Research Approach

1.3.1 General considerations

Based on the scope of the research we present models that can be applied on a particular freeway route given that

1. A traffic data collection system along the route of interest is installed. This system may consist of infrastructure and / or non-infrastructure bound detection equipment.

2. Actual (mean) travel times (per departure time period) along the route of interest are either measured or can be estimated from data.

3. A sufficiently large historical database is compiled of input (traffic detection - may even be travel times - and optionally ambient and external conditions), and output data (travel times) per departure time period. This database is required for calibration and validation.

The basis for this research is a rigorous analysis of the travel time prediction problem. Therefore, chapter 2 provides definitions of the main concepts relating to travel time (the terminology used), and outlines the main factors influencing travel time (e.g. traffic demand and supply factors). Based on these issues, chapter 2 provides a framework for short term freeway travel time prediction, which is the basis for all the models, tools and methods developed in the main body of this dissertation thesis. The key components in that framework are a traffic data collection system, an offline travel time estimation algorithm, robust preprocessing and data cleaning procedures, and the actual travel time prediction model.

Chapter 3 establishes the relationships between travel times and observable traffic quantities (speeds, flows) mathematically. These mathematical relationships naturally lead to the important issue of *travel time estimation*. Although sometimes the term travel time estimation is (mis)used as a synonym for travel time prediction, travel time estimation is referred to strictly as the offline translation of speeds and/or flows (travel times, occupancies) into mean travel times.

Since the models developed in this thesis should be applicable in real-time, a thorough understanding on the discrepancies between what is actually measured (for example with local detection equipment) and what is theoretically needed for travel time estimation (or prediction) is an important requirement. A prominent example is the well-known difference between arithmetic time mean speed (which some local detection equipment calculates) and space mean speed (which is required for travel time estimation). As a consequence several approximations are proposed that correct for the overestimation of mean speed (and thus underestimation of travel time) through arithmetic time averaging.

1.3.2 Model derivation approach

It is generally accepted that traffic prediction is a complex, non-linear spatiotemporal problem, which is ultimately the result of (inherently complex and non-linear) human behavior (chapter 4). Nonetheless, there exists a wide and detailed body of theoretical and empirical knowledge on the propagation and operation of vehicular traffic (see e.g. (Hoogendoorn & Bovy 2001)). Traffic flow theory provides invaluable insight into the nature of that complexity and the constraints it poses on models and methods that tackle it, and is therefore a solid basis for any travel time prediction model.

Model development is therefore based on the flow of (state) information in traffic processes (chapter 5). State information here depicts average vehicle speed (distance per unit time) and vehicular density (the number of vehicles per unit space). From traffic flow theory it is known that in free-flow conditions state information travels in the same direction as traffic does, while in congested conditions state information may also flow in the opposite direction. A good and easily observable example of the latter is a traffic jam spilling back in the upstream direction.

The obvious solution approach is then to exploit a traffic flow simulation model (see chapter 4) to predict the flow of state information and subsequently derive the resulting travel times. This, however, still requires prediction of the inputs to such a model, in particular traffic demand (Origin-Destination (OD) flow patterns) and supply (capacity, spillback) on the boundaries of the route of interest. In this case separate models should be developed for the short term prediction of these boundary conditions, for example based on statistical assumptions on the data available or equilibrium assumptions. However accurate the model in itself, its predictions strongly depend on the quality of those (predicted) inputs.

As such, we argue a *data-driven* approach to predict travel times from data *directly* is a more appropriate choice, given enough data are available. A second compelling argument in favor of data driven and specifically neural network approaches, is the following. A well trained neural network model has the advantage that also feedback processes (due to user response to the system) are automatically dealt with, since these are also "present" in the data. It simply does not matter for a neural network model whether a specific traffic condition is the result of a feedback process after it is installed, or not, as long as it is familiar with the resulting data. Unless drivers respond with completely different drive behavior (speeds, headways), most likely the response to the travel time prediction system yields traffic conditions also observed in the data with which the model was calibrated. A similar argument can be made for changes in OD flow patterns[4]. A traffic flow simulation model might produce different results for different OD patterns, while a data-driven approach 'does not care' about underlying OD flow patterns, given it is familiar with the resulting traffic patterns observable in the data. A data driven approach is hence intrinsically robust to issues (user response, changes in OD patterns) that are both difficult to observe and difficult to model.

As a basis a state space formulation of the travel time prediction problem is used, which in fact leads to a particular type of data driven model: a recurrent or spatiotemporal neural network, referred to as the state space neural network (SSNN). Its (state space) topology allows the model to track the flow of state information through time and space, much like in a traffic flow model, albeit that the states in the SSNN have no direct physical meaning (as is the case in traffic flow models). There are a number of clear benefits of the proposed state space formulation. The first is that it alleviates model designers from tedious input and (to a degree) model selection procedures, which are

[4]OD estimation is an underspecified problem, since there are many possible (but unobserved) OD flow patterns that result in the same (observed) traffic flow patterns .

inherent to data driven approaches in general (see for example (Pancratz 1991) and (Box & Jenkins 1976)), and especially in the application of artificial neural networks (ANNs), where often trial and error procedures (e.g.. in (Fallah-Tafti 2001)) are used to select the appropriate inputs and to determine the model topology. But even when more structured approaches for input- and model selection are taken, such as cross-correlation analysis (Innamaa 2001), or genetic algorithms (Abdulhai et al. 1999), the proposed solutions are tailored for the specific application and not necessarily transferable to another.

Secondly, we argue that a state space approach suits the travel time prediction problem better than a (black box) time series or regression approach. Traffic patterns are spatiotemporal patterns, that is, they are spatial patterns that evolve over time. The state space structure allows the model to track those spatiotemporal processes in a very efficient and generic manner. Finally, as a result of its generic state space parameterized structure, the SSNN intrinsically also satisfies the adaptivity criterion. Although its mathematical structure remains unchanged, its parameters can be recalibrated on "new" data if the circumstances require this.

1.3.3 Calibration and evaluation approach

As mentioned earlier, this thesis develops travel time prediction models that are *accurate and valid* (in terms of predictive performance), *robust* (insensitive to input failure) and *reliable*, subjects which are covered in chapters 5, 6, and 7 respectively. Although the latter quality encompasses the previous two, reliability as such is quantified by means of confidence and prediction intervals that express our uncertainty in the model outcomes.

In a neural network context calibration (setting model parameters) is referred to as *training*. In the aforementioned three chapters training, testing and rigorous analysis of the internal workings of the SSNN model is performed in a controlled environment, that is, on the basis of synthetic data from a microscopic traffic flow simulator (note that in chapter 8 all results are validated in a real-time environment - see further below). This approach is chosen since a micro-simulation environment provides full control over all the quantities that influence not only the quality of our models, but also the quality of our analysis:

- A micro-simulation environment allows (realistic) simulation of traffic flow operations under different traffic conditions (congested, free-flow, intermediate).
- Micro-simulation provides all required input and output data, that is, aggregate data such as speeds and flows at detector locations, average densities on freeway sections between detectors, mean travel times, but also individual data such as individual vehicle trajectories and travel times.

- Various scenarios of input failure (detector failure) can be easily simulated and methods that enhance the robustness of the models to input failure can be thoroughly tested.

- Micro-simulation also provides travel time distributions with which the various components that constitute the uncertainty in each prediction can de estimated. These include the noise (variability) in the target variables (travel times), but also the uncertainty in the neural network parameters.

With respect to the last item the Bayesian framework introduced in (MacKay 1995) for feed-forward neural networks is applied to the SSNN. This framework prevents that a neural network learns just the idiosyncrasies of the particular training data set, instead of the underlying processes that generate the data. As a bonus it provides quantitative information on the uncertainty in the resulting parameter vector by means of a variance / covariance matrix of the parameters. With this matrix error bars can be obtained on each prediction so that confidence intervals can be constructed.

There are, however, some critical notes the reader should keep in mind. Although the traffic micro-simulation tool used in chapters 5, 6, and 7 (FOSIM - Freeway Operations SImulation Model, see e.g. (Vermijs & Schuurman 1994)) has been extensively calibrated and validated for Dutch freeways, the predominant purpose of this model is to estimate freeway capacity. As such, its (macroscopic) results comply fairly well with observed flows and average speeds in free-flow conditions and conditions just before traffic breakdown. Also mean travel times (which largely depend on capacity) are fairly realistic. However, the model has never been extensively calibrated in congested conditions, certainly not in terms of individual driver behavior, which implies that travel time *distributions* for a particular departure time period may not be very realistic under severe congestion.

1.3.4 Validation and real-time application

In chapter 8 all findings are combined and validated in a real-time environment. To validate the approach, the quality of the offline travel time estimator is discussed, as it provides the target data for our travel time predictor. For that a small data set of actually measured travel times is used. As noted in the previous section most likely significant differences between these actual travel times and the ones obtained from simulation are found. Given the travel time estimator produces unbiased offline travel time estimates (or when this bias can at least be quantified), the SSNN model can be safely trained with a large database with estimated travel times. Predictive performance is validated on a separate (also large) database of (estimated) travel times.

1.4 Contributions and Scientific Relevance

1.4.1 Summary of contributions

Below we list the main contributions to the State-of-the-Art offered in this dissertation thesis

1. A neural network based model for short term freeway travel time prediction, the so-called *state space neural network* (SSNN), which outperforms current models in the Dutch situation by far and performs equally well or better than a range of state-of-the-art travel time prediction models reported in literature. Although the SSNN model is an artificial neural network[5], its design (in terms of input- and model selection) is not "black-box" nor location-specific, which are common criticisms on ANN solutions (e.g. (Smith & Demetsky 1997)). Instead, it is based on the lay-out of the freeway route of interest. Moreover, the internal states in the SSNN model are strongly related to the actual traffic processes. (Chapters 5 and 8).

2. A heuristic based on the backpropagation training algorithm to *explicitly quantify the contribution (relevance) of each of the SSNN parameters and neurons in real-time*, including the inputs (see chapter 5). As such, the SSNN model does not have to be viewed as a "black box", and its internal operation can be analyzed rigorously.

3. *A robust short term freeway travel time prediction framework*, which exploits the SSNN model and preprocessing strategies based on *data imputation*, that is replacement of missing or unreliable data with for example exponential forecasting or spatial interpolation. For both random as well as structural input failure, this framework still produces accurate travel time predictions. Herein, our modelling framework provides a significant step forward in the State-of-the-Art. Although "simple" imputation techniques tend to seriously change the statistical properties of data (Schafer 1997), the SSNN model appears invariant (robust) to the damage done by these preprocessing procedures, even for high degrees of "missingness" (chapter 6).

4. We demonstrate it is also possible to *train the SSNN with missing input data and hence make it intrinsically robust* to missing data. This strategy does improve robustness largely, albeit at the cost of predictive performance. This is due to the

[5] Although many definitions depending on application field and theoretical context exist, we consider Artificial Neural Networks (ANNs) a general class of non-linear parameterized regression and classification models. In this sense, well known statistical models are in fact special classes of ANNs. As (Bishop 1995) convincingly argues "neural networks can be regarded as an extension of the many conventional techniques which have been developed over many decades". For example, a linear regression model is a special case of a feedforward multi-layered perceptron.

fact this inherently makes the travel time prediction problem to be solved more complex (chapter 6).

5. A set of methods and techniques for *quantifying uncertainty* in the predictions of the travel time prediction framework (chapter 7 and 8). Although the techniques as such are established and based on empirical Bayesian statistics (MacKay 1995) and random subsampling, their application in the domain of travel time prediction is new.

6. We show that confidence intervals can be interpreted as *quantitative indicators for the predictive quality* of the SSNN model (chapter 7), that is, they indicate the magnitude of the prediction error. This is a powerful result, since it allows one to real-time monitor the predictive quality of the SSNN, without actually measuring travel times.

7. A novel algorithm for offline estimation of travel times, the so-called *piece-wise linear speed based (PLSB) trajectory algorithm*. This method is an extension of the widely used piece-wise constant speed based (PCSB) trajectory algorithm. The improved PLSB algorithm reduces both bias and residual error with respect to the well-known PCSB method (chapter 3).

8. *A method to estimate and correct for the bias caused by local arithmetic mean speeds*. Using local arithmetic mean speeds leads to serious underestimation of travel time. The bias is related to speed variance, for which we propose an estimator based on local density and time series analysis respectively . With this correction method, the bias can be almost completely removed (chapter 3).

9. A taxonomy for travel time prediction models (chapter 4), which includes the main factors that influence travel time. The principal division is between short term and long term travel time prediction, which are very different problems requiring very different types of models to solve. Although according to this taxonomy, this thesis focusses on just a small branch of travel time prediction models, the taxonomy in itself can be used to quickly classify any travel time prediction model.

1.4.2 Theoretical and scientific relevance

In this dissertation thesis a number of issues are discussed which in our view are theoretically and scientifically relevant.

1. We show that domain knowledge (in our case traffic flow theory) can be successfully integrated into ANN solutions. The benefits of such an approach are twofold.

(a) It leads to more efficient models (why learn a model something that is already known)

(b) It allows for qualitative as well as quantitative analysis of the internal workings of the ANN. The ANN model used in this dissertation thesis (the prementioned SSNN) has the same general state space form as for example a macroscopic traffic flow model, which makes it possible to deduce which freeway sections contribute the most to delays in a particular traffic situation and relate the model's internal dynamics to the actual traffic processes.

2. More generally, but related to the previous point, we demonstrate that there is fundamental scientific value in combining heuristic (data driven) models with general concepts from traffic theory to describe properties of traffic processes (in this case travel times).

   (a) Traffic processes are generally highly complex and dynamic due to individual actors who all behave differently (stochastically) and also inconsistently over time. Inherently, traffic flow models deviate from practice not only due to random errors but also due to the fact that people do not behave like gas or fluid molecules. The only undisputed concept in (over 5 decades of) traffic flow modelling is still the principle of *conservation of vehicles (CoV)* (Hoogendoorn & Bovy 2001). All other concepts (e.g. anticipation, relaxation, attention levels, psycho spacing, etc.) are at best parameterized approximations to capture the residual non-linear, dynamic and stochastic phenomena observed in traffic data. General data driven models such as SSNN models, lacking apriori behavioral assumptions on those phenomena, may prove better in reproducing these phenomena than detailed models of individual behavior.

   (b) Nonetheless, we argue that the greatest potential lies in combining theory and advanced data driven techniques. Using the techniques described in this thesis to determine relevance of parameters and input one could for example design a general SSNN traffic flow model, which respects *CoV*, and learns the residual complex nonlinearity directly from data. Ultimately, this may result in new empirical findings, which either support known theoretical concepts, or give way to new directions in traffic flow theory. Either way, we argue in this way a heuristic model as the SSNN can be used as a powerful tool in traffic flow theory development.

3. We also emphasize on the scientific and theoretical relevance of Bayesian techniques for controlling model complexity used in this thesis. The implications and opportunities of these methods go well beyond just (parameterized) models for short term prediction of travel times on freeways developed here. Bayes rule embodies Occam's Razor quantitatively and automatically (see appendix C), and as such the Bayesian framework enables automated complexity control for any parameterized (non-linear) model.

1.4.3 Practical relevance

There is a very clear need for accurate and fast travel time prediction tools in practice. In the Dutch situation for example, a large scale traffic data collection system (MONICA) is deployed on the larger part of the national highway network. MONICA collects local variables (time mean speeds and intensities) from dual loop detectors about every 500 meter. On some parts of the network these data are available per lane and per user class. Also an increasing number of variable message signs (VMSs) have been set up at strategic bifurcations that, at current, only display queue lengths calculated by rudimentary algorithms. The practical relevance of this dissertation thesis is in its emphasis to develop models and methods that can be applied in a real-time environment, such as the MONICA system. Both the short term freeway travel time prediction model, the improved offline travel time estimation algorithm and the techniques developed for data cleaning and quantification of uncertainty can be readily deployed on top of MONICA.

There is also a very clear need for robust travel time prediction tools, that is models that still function in case of missing or corrupted data. Again taking the MONICA system as an example, on average on a particular time instant 12% of the measurements are either missing or dubbed unreliable due to maintenance backlogs, temporal power or communication failure or for example incidents and accidents. There are even a significant number of occasions in which over 20% of the measurements are missing or corrupt. We propose robust and easy-to-implement procedures that account for the missing data and allow accurate predictions even at high degrees of input failure. In case those methods still fail the proposed travel time prediction model has a built-in "warning mechanism" that enables the traffic manager to detect that "something is wrong", either with the SSNN model, or the data with which it is fed.

Although we demonstrate how the travel time prediction framework can be deployed in a dual loop based detection system (or any other system measuring speeds and flows), the approach (since it is essentially data driven) could also be applied for traffic data collection systems that measure other quantities that are physically or statistically related to travel times, even systems that measure travel times themselves. Examples include automatic vehicle identification systems (AVI), floating car data collection systems (based on GPS, GSM, or otherwise), or even less sophisticated systems such as single loop detection systems. On the application side, not only VMS systems could benefit from short term travel time prediction models but also in-car navigation systems and web-based online traffic information services.

Finally, due to the automatic and quantitative measures of uncertainty built in the framework we present, there is also a practical value for traffic managers operating road facilities and the traffic data collection systems on them. This quantitative measure indicates whether or not "something is wrong", either with the SSNN model (it needs retraining, because of structural changes in the infrastructure or traffic patterns), or with the information chain feeding the SSNN (detection equipment failure, commu-

nication or power problems, maintenance backlogs, etc.)

1.4.4 Implications and recommendations for future research

In the light of the research presented, we make a number of recommendations for future research efforts in this field.

1. A better understanding and use of artificial neural network (ANN) type models in traffic and transportation is necessary. Too often, and particularly in the case of travel time prediction, ANNs are considered ”last resort" parameterized solutions for which both design and calibration are based on "trial-and-error" and "engineering judgement", rather than sound theory and mathematics. We are convinced this is a serious misconception caused by unfamiliarity of the latest developments in the field of neural networks, for example:

   (a) For improving generalization, (avoiding under- and overfitting) - a prominent challenge in both statistical techniques (regression, ARIMA models) as well as in ANNs - a wide range of heuristic methods (early stopping, cross-validation (Prechelt 1998), pruning), but also Bayesian techniques (see (Papadopoulos et al. 2001) for an overview and comparison) is readily available (chapter 7, and appendices B, C). Particularly, the Bayesian regulated training algorithm, consistently produces efficient parameter settings regardless of the initial number of parameters in the model. In this sense, overfitting is not a problem inherently associated with ANN models but a problem resulting from using out-dated ANN training algorithms. We strongly recommend the Bayesian approach be used, whenever this is possible.

   (b) Given that an ANN is trained with a backpropagation type of training algorithm, and proper care has been taken to avoid overfitting (preferably the Bayesian method) it is in fact straightforward to explicitly quantify the contribution (relevance) of each of its parameters and neurons, including the inputs (see chapter 5). Depending on the ANN topology and application specific circumstances we recommend a measure of relevance be developed similar to the one developed in this thesis.

2. We found large discrepancies between travel time distributions from a microscopic simulation model and actual travel time distributions (per departure time period). Since the microscopic traffic flow model used here[6] has been extensively calibrated with mean speeds and flows measured at Dutch highways in free-flow and near capacity conditions, and contains similar car following and lane changing routines as in many commercial simulation models (e.g. VISSIM,

[6] Freeway Operations SImulation Model (FOSIM), see e.g. (Vermijs & Schuurman 1994)

Paramics), this discrepancy is most likely due to insufficient calibration and validation in severely congested conditions. We are convinced that the parameters in microscopic traffic flow simulation models should preferably be calibrated with mean travel times and particularly travel time distributions per road segment per departure time period. This will lead to much more realistic models of driver behavior in congested traffic. The travel time distribution grows smaller (and not wider!) as mean travel times increase due to efficient but not yet properly modelled driver behavior (car following and lane changing) in congestion.

1.5 Thesis Outline

A detailed overview of the structure of the main body of this thesis is given in fig. 1.2. In chapter 2 we first present and define the 'travel time terminology' used throughout this dissertation thesis and the general framework, particularly the difference between travel time estimation and prediction. This qualitative analysis is followed by a detailed description of the travel time estimation problem in mathematical terms in chapter 3. In that chapter we also introduce a novel algorithm for offline travel time estimation, which we will use in later chapters in a real-time travel time prediction framework. Next, chapter 4 discusses the complexity of travel time prediction, presents a taxonomy for travel time prediction models and overviews the current State-of-the-Art in short term freeway travel time prediction.

Based on these four chapters, chapter 5 then derives, analyzes and rigorously tests a novel freeway travel time prediction model which is based on a state space formulation of the travel time prediction problem, the state space neural network (SSNN). In chapter 6 methods are explored that enhance the SSNN robustness with respect to missing data, both for random and structural detector failure. Subsequently, in chapter 7 methods to quantify unreliability or rather the uncertainty in our SSNN predictions are introduced. Apart from missing data, there are numerous sources of uncertainty (amongst other things the calibration of our SSNN model itself), which we identify and quantify through Bayesian probability theory and random subsampling techniques.

Chapter 8 then combines the findings of chapters 5, 6, and 7 in a real-time travel time prediction framework which is applied and tested in the Regiolab Delft test-bed (appendix E, (Van Zuylen & Muller 2002)). In this test-bed real-time data from a large number of inductive loop detectors are available. In chapter 9, we outline how the proposed SSNN model could be extended. Amongst other things we briefly discuss non-freeway travel time prediction and the use of sources of data other than local infrastructure bound detectors.

Finally, in chapter 10, the main conclusions of this dissertation thesis are presented and directions for future research are outlined. Common performance indicators are listed in appendix A, while appendices B and C present a detailed mathematical description of the SSNN and the Bayesian method for SSNN training. In conclusion, appendix

D, which presents a first order traffic model based Kalman Filter for data cleaning and appendix E briefly summarizes the Regiolab Delft project.

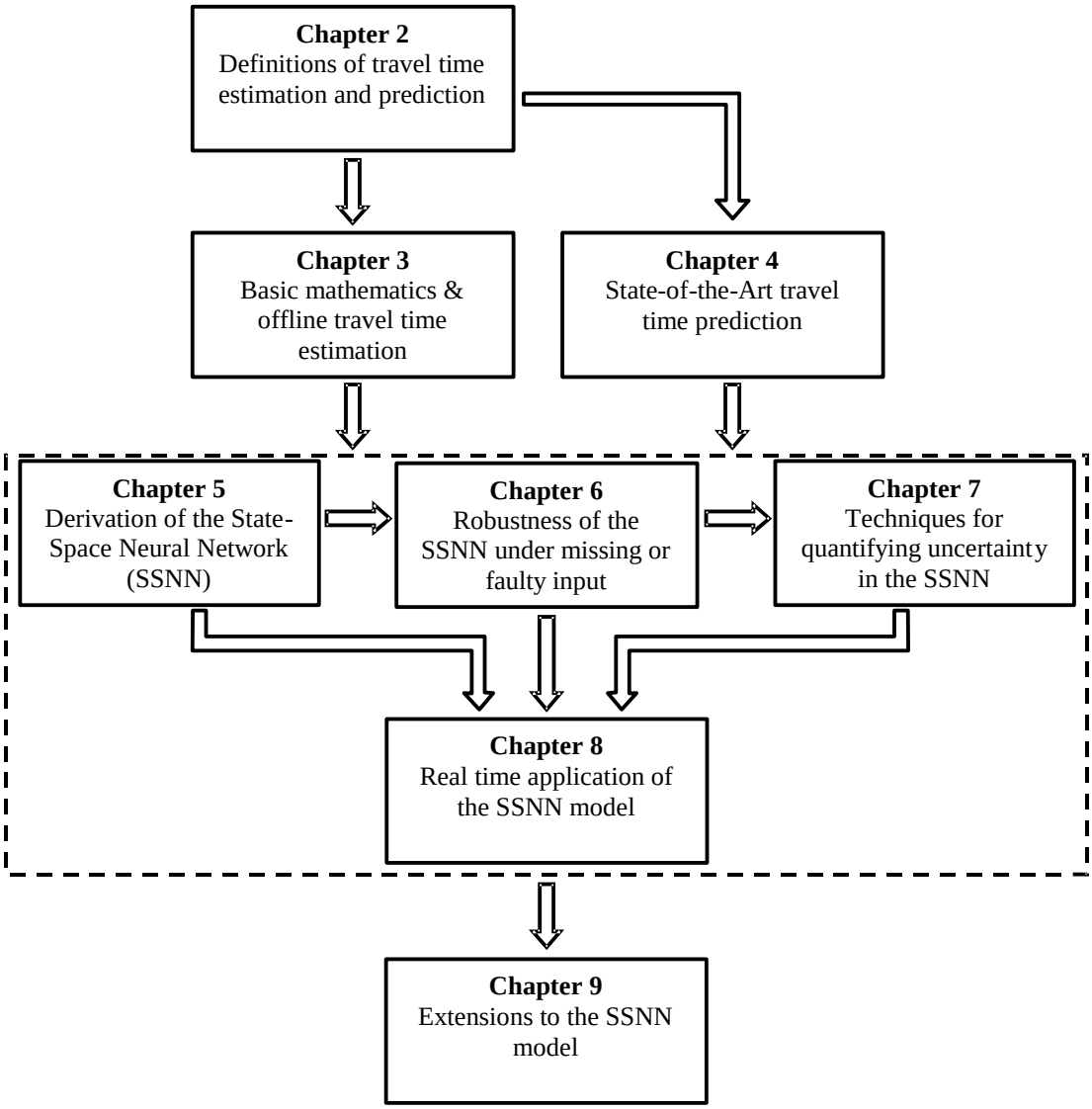


Figure 1.2: Schematic overview of the main body of this dissertation thesis.

Chapter 2

Conceptual Framework

2.1 Introduction

This chapter presents clear definitions of individual and mean travel time, travel time estimation and travel time prediction respectively. Furthermore, factors that influence travel time on uninterrupted roadway facilities such as freeways are explored and categorized. Although these factors are subdivided into (crisp) categories, such as traffic demand and supply, they are closely interrelated in a highly complex, nonlinear and dynamic fashion. In this and the ensuing chapters travel times are defined for two (spatial) entities: freeway sections and freeway routes.

Definition 1 *A freeway section is a stretch of several hundreds to several thousands metres of freeway, characterized by a specific number of lanes, possibly containing one or more exit and / or egress points (on and off ramps).*

Definition 2 *A freeway route is constituted of a finite number of contiguous freeway sections.*

Unless specifically stated otherwise, a section is denoted with index k and a route with index r. In the next chapter (3) we will show how decomposition of a route into adjacent sections enables one to calculate section level travel times and combine these to route level travel times. Ultimately, for ATIS purposes, route level travel times are the quantities of interest. Therefore this chapter will focus on route level travel times. The final part of this chapter presents our framework for short term freeway travel time prediction, on which all subsequent work in this thesis is based.

2.2 Definitions of Travel Time Estimation and Prediction

2.2.1 Individual and mean travel time

For non traffic scientists travel time is probably the most common and relevant characteristic associated with a trip. We define individual travel time as follows:

Definition 3 *The individual travel time on a route r at departure time t is the time it takes an individual traveller (driver, pedestrian, passenger) to traverse that particular route.*

As such, it is the quantity, which is most easily measurable and interpretable by an individual traveller, and is hence often considered the principal attribute affecting that traveller in his or her decisions, for example in terms of mode, route and / or departure time choice. This assumption underlies most contemporary advanced traffic information systems (ATIS), which aim to provide road users with accurate and up-to-date traffic information such as travel times. It is, however, important to note that for individual travellers travel time is considered as a property of an individual trip. This means that the individual driver may interpret traffic information provided (e.g. on VMS panels on freeways) differently (see also (Van Berkum & Van der Mede 1993)), yielding a wide range of responses to that information. Furthermore, also accuracy and reliability of traffic information may be assessed differently amongst different drivers.

Depending on the type of application, in the vocabulary of traffic engineers and scientists travel time is often regarded as a property of either a group of drivers or passengers, or more commonly as a property of a particular section or route on a network. Also, for ATIS applications it is common to define travel time for departure time *periods* (of several time units) rather than time instants. We therefore define:

Definition 4 *The mean travel time on a route r for vehicles departing in period p is the average time it takes these vehicles to traverse the specific route under the prevailing conditions on r during p.*

Here the term *mean* (travel time) is used in the statistical sense, that is: the mean value of individual travel times of drivers departing during some time period p

$$\tau_r(p) = \frac{1}{N} \sum_i \tau_{rti}, \; t \in p \tag{2.1}$$

where τ_{rti} denotes the travel time for an individual i on route r with departure time $t \in p$, and N denotes the total number of vehicles departing in time period p on route

r. In this thesis we will always refer to mean travel time, which inherently is the result of the average behavior of individual travellers.

Note that both individual and mean travel time are defined as "experienced" travel times. This implies that for example mean travel time can only be measured *after* a group of drivers has traversed the route of interest. Inherently, provision of traffic information in terms of measured travel times always contains past information, since the last measured travel time on a particular time instant t is the travel time experienced by a vehicle that finished traversing the route at time t.

2.2.2 Prediction horizons: short and long term prediction

Since travel time is conceived differently depending on context and application field, travel time prediction also means very different things in different contexts. Answering the question "How long will it take me to get from Amsterdam to Brussels by car, when I leave next Monday at 9 A.M.?" requires a completely different set of tools and data than answering "How much time will it take me at this very moment to traverse the Southbound stretch of the A13 highway between Delft and Rotterdam, which is about 10 kilometers long?". Nonetheless, both problems are regarded as travel time prediction problems. In both cases we wish to predict the mean travel time of vehicles on route r departing in time period $p = [t_0, t_1]$, that is

$$\tau_r(p) = G(\Omega_r)$$

where Ω_r depict the traffic and ambient conditions[1] (e.g. weather) on route r for time instants $\geq t_0$. Now let $p^* = \left[t_0^*, t_1^*\right]$ denote the current departure time period in which case

$$T = \max\left(t_0 - t_0^*, 0\right)$$

denotes the *prediction horizon* from t_0^* onwards. T thus reflects the number of time units (minutes, hours, etc.) between the start of the current departure time period and the start of the departure time period for which mean travel time is predicted. Generally speaking, the larger the prediction horizon, the more one has to rely on modelling assumptions regarding the traffic conditions during the departure time period and the time en route ($t > t_0^*$). These can be statistical assumptions based on historic databases of traffic conditions, behavioral assumptions which underlie for example traffic assignment models, or assumptions based on economic, social or demographic considerations. If the prediction horizon, however, is small or zero, that is, predicting travel times for vehicles departing at current or near-future departure time periods, we can (partially) rely on current and near-past traffic conditions, not only on route r but also on adjacent or connecting routes (see fig. 2.1). Traffic conditions can be viewed as spatial patterns that evolve over time. Practically, this means that current and near-past traffic conditions strongly influence the traffic conditions in the near future.

[1] Note this does not imply travel times relate deterministically to a well defined set of attributes.

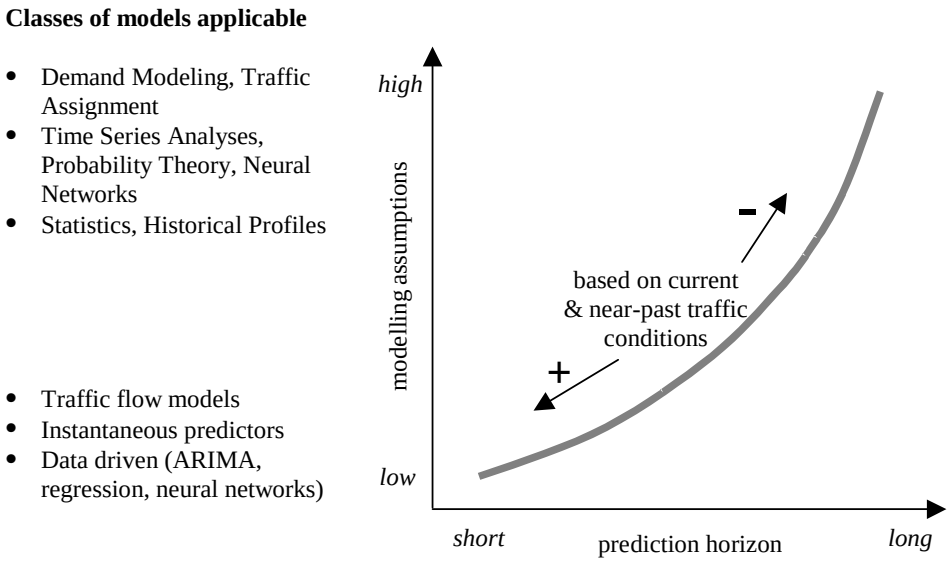


Figure 2.1: Modelling assumptions versus travel time prediction horizons. The larger the prediction horizon the more we need to rely on assumptions (regarding for example traffic demand and driver behaviour).

We will return to these different types of travel time prediction problems in chapter 4, where we discuss the current State-of-the-Art in travel time prediction. This thesis focusses on travel time prediction for short prediction horizons and particularly on *online travel time prediction*, which we define as follows:

Definition 5 *Online travel time prediction on a route r is predicting the mean travel time for vehicles departing in the current departure time period $p = p^*$. In this case the prediction horizon equals zero.*

In many contributions found in literature (e.g. (Zhang & Rice 2003), (Chen & Chien 2001), (Dia 2001), and (Ishak & Al-Deek 2002)) the term *short term travel time prediction* is used. Although different interpretations are given, the following definition captures most of these:

Definition 6 *Short term travel time prediction on a route r is predicting the mean travel time for vehicles departing in the current or near-future departure time period $p = p^* + T$ where T is in the order of 0 to 60 minutes, so that $0 \leq t_0 - t_0^* \leq 60$.*

Consequently, online travel time prediction is a special case of short term travel time prediction. Long term travel time prediction can be interpreted in similar terms as short term prediction with $T \gg 60$ minutes. Fig 2.2 schematically overviews online, short and long term travel time prediction by means of vehicle trajectories (distance plotted as a function of time).

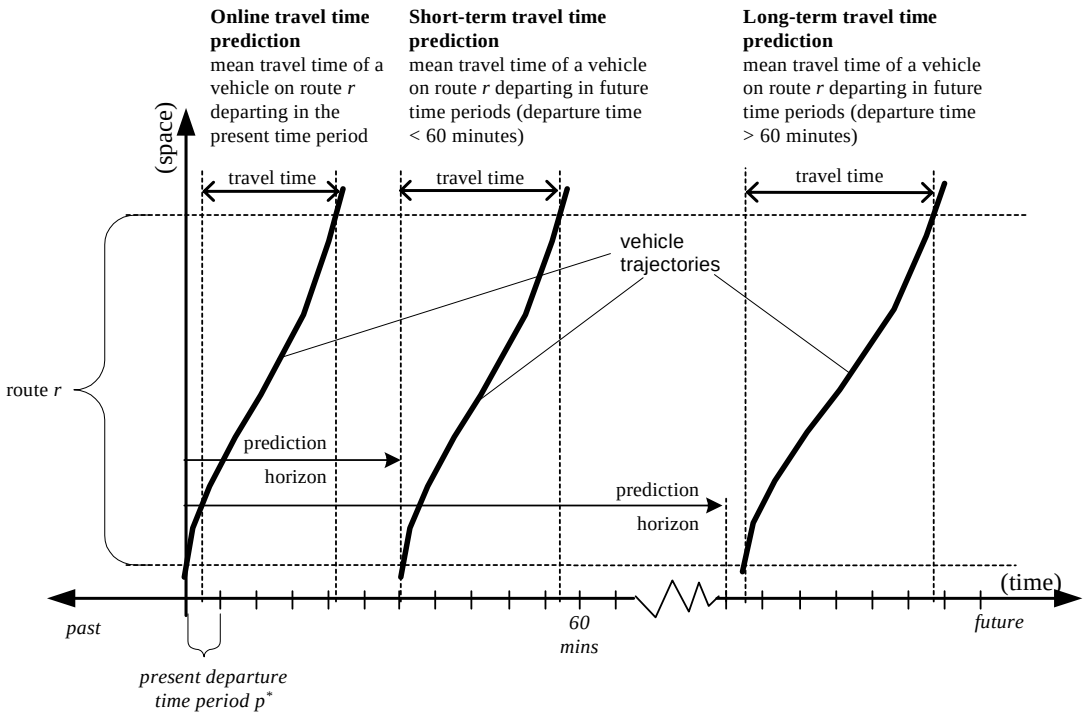


Figure 2.2: The differences between short and long term travel time prediction.

2.2.3 Difference between travel time estimation and prediction

In literature, we often encounter the term *travel time estimation*. Although in some cases the term estimation is (mis)used for prediction we will strictly use the term estimation according to the following definition*:*

Definition 7 *Travel time estimation is calculating (mean) travel times of realized trips or flows based on known speeds, flows, travel times, or other quantities which are mathematically related to (mean) travel times. These "known" conditions may be either measured, estimated or even predicted themselves.*

Thus, travel time estimation pertains to reconstructing travel times from other traffic quantities (speeds, flows, densities), regardless of whether these traffic quantities are measured, predicted or otherwise. This definition implies that for estimating travel times, we do not need to concern ourselves with the dynamics and hence prediction of traffic. In other words, travel time estimation is a static (mathematical) mapping from one set of traffic variables to another. In contrast, *travel time prediction* refers to calculating travel times for unknown (future) traffic conditions (see fig. 2.3), and inherently forces the analyst to deal with the complex dynamics of traffic processes.

With in mind the clear difference between travel time estimation and prediction we introduce two more definitions, which reflect different approaches to travel time prediction:

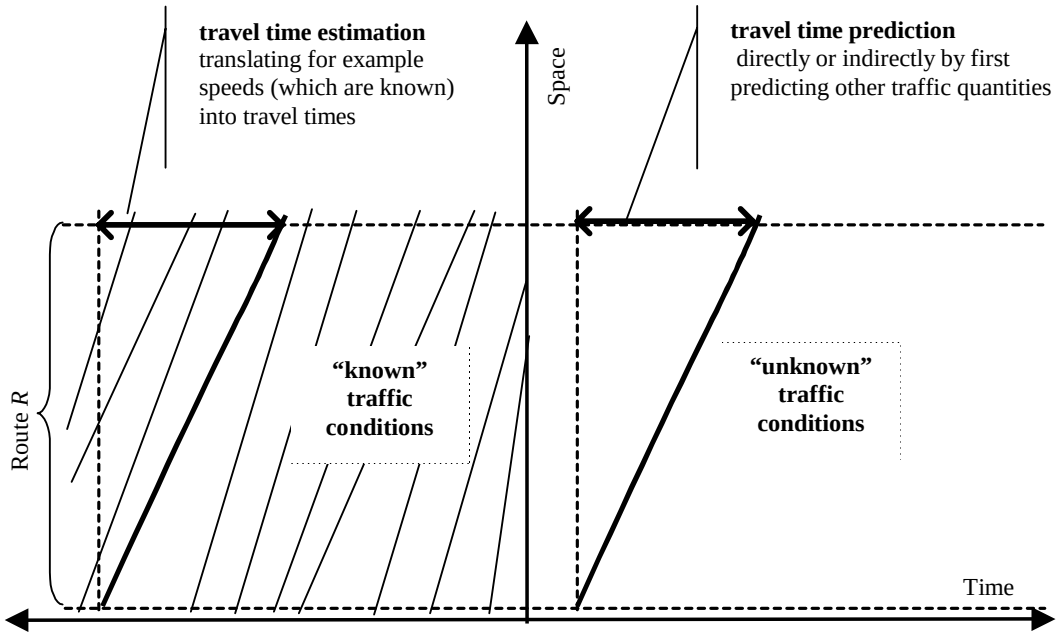


Figure 2.3: The difference between travel time estimation and prediction.

Definition 8 *Indirect travel time prediction is predicting other traffic quantities (e.g. speeds, flows, densities) which are used to derive (estimate!) travel times.*

Hence, travel time estimation techniques can also be used as tools / components in a travel time prediction model. An example of indirect travel time predictors are model based approaches, which we discuss in chapter 4 (State-of-the-Art), in which traffic flow models (for an overview see e.g. (Hoogendoorn & Bovy 2001), or (Helbing 1997)) predict mean speeds and densities based on which travel times can be derived, for example with travel time estimation techniques. Opposed to indirect travel time prediction we define

Definition 9 *Direct travel time prediction is predicting travel times without the intermediate step of predicting other traffic quantities*

2.2.4 Instantaneous travel time versus dynamic travel time

In this last subsection we address another type of travel time which is often used in literature and also applied in traffic engineering practice.

Definition 10 *The instantaneous travel time is the travel time a vehicle would experience on a particular route r departing in period $p = p_0$ (at time instant $t = t_0$) if the traffic conditions on r for periods $p \geq p_0$ (time instants $t \geq t_0$) remain stationary.*

Normally, on a reasonably sized route, no vehicle would ever experience an instantaneous travel time, since the assumption of stationarity will most likely not hold, especially not in congested conditions. Nonetheless, as we will see in chapter 4, instantaneous travel times of the current departure period p^* serve as online travel time predictions distributed by the national Traffic Information Centre of the Dutch Ministry of Transport, Public Works and Water Management. For clarity we will refer to them as *Instantaneous travel time predictors*:

Definition 11 *Instantaneous travel time predictors calculate the instantaneous travel time on a particular route r at the current departure time period $p = p^*$.*

In section 4.3 two examples will be given that clearly illustrate the differences between instantaneous and actually experienced travel times. In general, the assumption of stationarity causes the instantaneous travel time to underestimate real travel time as congestion sets in and vice versa to over estimate travel times as congestion dissolves.

2.3 Factors Influencing Travel Time

In this section we identify factors influencing travel time or more precisely the traffic conditions that influence these travel times. In general, travel times on freeway (and also other types of) networks are the result of the dynamic interplay of traffic demand and supply. The first term reflects the number of vehicles[2] (drivers) using the freeway facility, while the latter reflects the level of service (e.g. travel time, capacity) the facility offers. For example, if on a freeway section (without on or off ramps) traffic demand (in terms of the number of vehicles per unit time) exceeds the maximum number of vehicles that can pass this freeway stretch (capacity[3]), traffic breaks down, congestion sets in and vehicles will inherently be delayed. However, in cases where a freeway stretch is connected to other freeways or other networks (on and off ramps, bifurcations, etc.) congestion also occurs in situations where demand does not exceed capacity of that facility, for example in case of queue spill back. Moreover, capacity (even for an isolated freeway stretch) is not a deterministic and static characteristic of a particular freeway facility, but rather a stochastic and dynamic function of many different (and often ill-predictable) factors. Similarly, traffic demand is a stochastic and

[2] since different vehicles classes may have very different characteristics (length, braking distance), in this context often the term person car equivalent (*pce*) is used. A truck may for example have a *pce* value of between 2 and 3 depending also on infrastructure geometry (e.g. grade, width)

[3] The exact definition of capacity is subject of a lively debate among traffic practitioners and scientists. The most widely used definition is the one proposed in the Highway Capacity Manual (Transportation Research Board 2000): "... the maximum hourly rate at which persons or vehicles reasonably can be expected to traverse a point or a uniform section of a lane or roadway during a given time period, under prevailing roadway, traffic and control conditions (HCM, p. 2-2)". For an overview of defining, measuring and estimating freeway capacity, see for example (Elefteriadou & Lertworawanich 2003).

dynamic function of many factors, including traffic supply factors. Nonetheless, we subdivide the factors influencing travel time into two groups

1. Factors influencing traffic demand
2. Factors influencing traffic supply characteristics

In fig. 2.4 a general overview is presented of both categories of factors. Although each class of factors is discussed separately, this does not imply that they are independent. On the contrary, most of these factors are strongly overlapping and (non-linearly and dynamically) dependent on each other. For example, some effects on traffic supply also have an (positive or negative feedback) effect on traffic demand and vice versa. Due to bad weather more travellers are likely to take the car, while at the same time weather conditions may reduce road capacity (driving comfort, visual abilities and safety), which in turn might persuade people to stay home or choose a different mode. Certain traffic management and control schemes may (indirectly) induce or reduce traffic demand. Nonetheless, for clarity, below a brief overview is given based on the categorization presented in fig. 2.4.

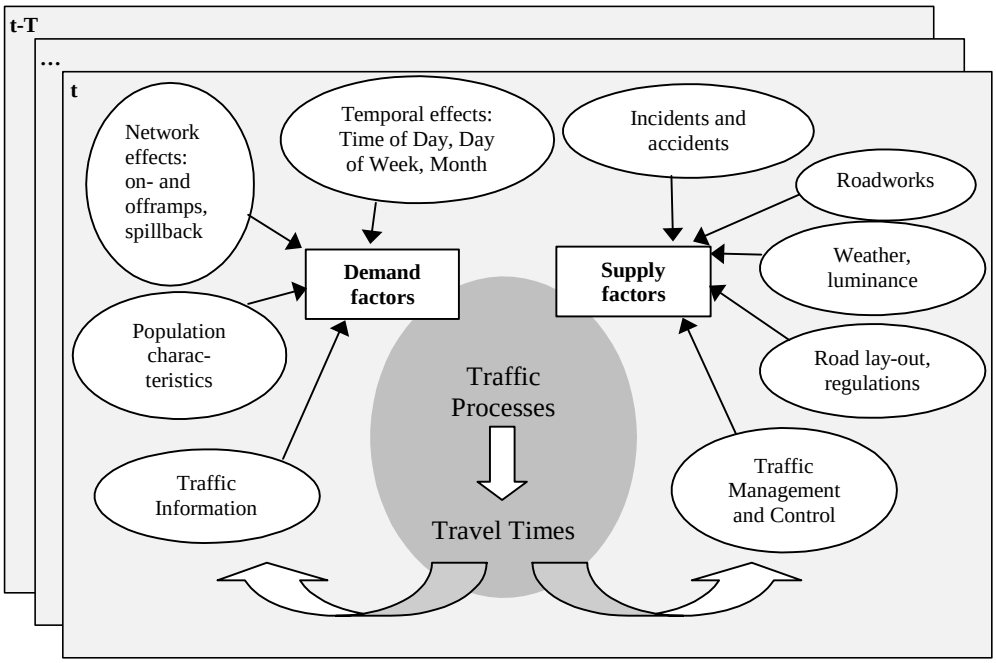


Figure 2.4: General (not exhaustive) overview of factors that influence travel time.

2.3.1 Factors influencing traffic demand

The main group of factors that fall into this category is what we will refer to as *Temporal effects*. These temporal effects are a consequence of for example daily and weekly

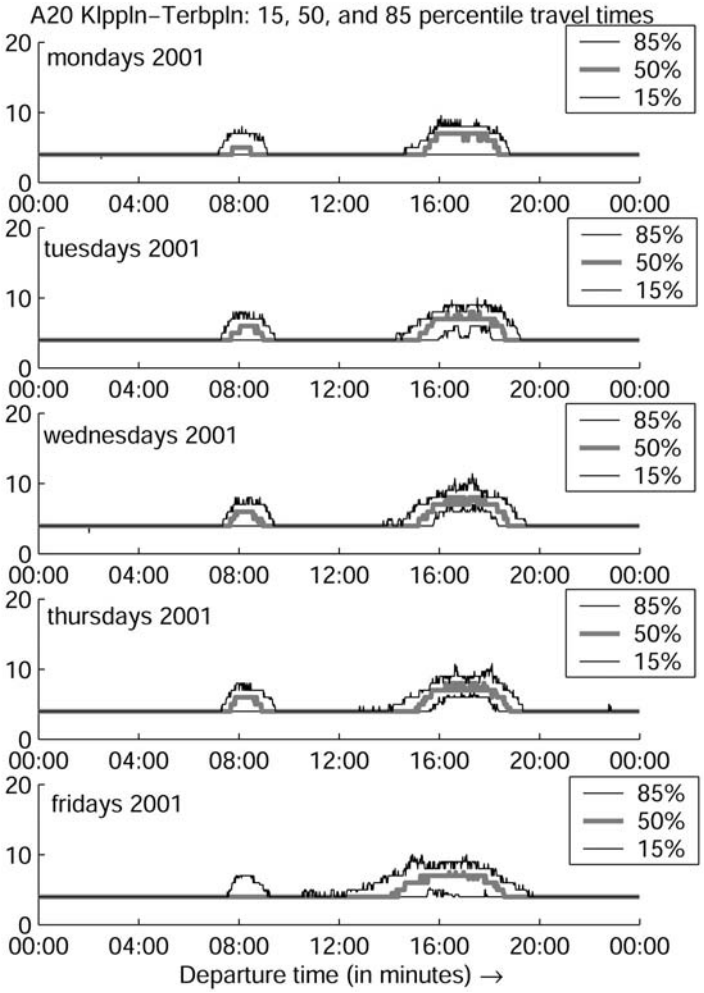


Figure 2.5: Examples of temporal effects on travel time. The graph shows 15, 50 and 85 percentile values of daily travel time profiles on weekdays in the whole of 2001 on a 6 km stretch of the A20 (northern part of the Rotterdam urban beltway in The Netherlands).

commuter patterns, and seasonal patterns reflecting for example holidays. Temporal effects as such do not generate traffic. A vehicle is neither present or absent on a network because it is for example 12:30 in the afternoon. Rather, these temporal factors are only descriptive variables that represent the effects of the temporal (but location specific!) distribution of activities (working, living, recreation) that generate traffic demand. In many applications that employ black-box methodologies to *short* term travel time or traffic forecasting temporal effects do not play a role of significance (e.g. (Ishak & Al-Deek 2002), (Williams 2001), (Lan & Miaou 1999)). Due to the variability of travel times it is not very likely that the expected travel time in the coming minutes equals the expected travel time in the same departure period of the previous week or even previous year. Rather, most of the short term travel time prediction models exploit strong correlation patterns of traffic quantities over time and space. In longer term forecasting algorithms, however, temporal effects play an important role (for example

(Kaysi et al. 1993), (Schrader et al. 2004), (Van Lint et al. 2004)). Algorithms solely based on temporal effects are often referred to as historic profiles (e.g. in (Smith & Demetsky 1997), (Park & Rilett 1998)).

Fig 2.5 shows clearly identifiable daily travel time patterns on weekdays on a 3-lane highway stretch in the Netherlands, with morning and evening peaks. The interesting aspect of these patterns is that on average (or rather in terms of median values) they do not differ all that much between different weekdays. However, in terms of variance (inter-percentile range) the daily patterns differ substantially. Travellers on Friday afternoons face much more uncertainty than travellers on Monday afternoon peaks. Thus, temporal effects do not only affect mean travel time patterns but also variance in travel times, implying that also to quantify uncertainty, temporal patterns play a crucial role.

The second group of factors that are categorized as demand factors are what we refer to as *network effects*, that is the effect of traffic on adjacent, connecting or parallel links and on- and off ramps on the traffic conditions on the link of interest. Depending on scale, these effects pertain to online, short to mid-term prediction problems. A typical example of how network effects influence travel time is phenomenon of *queue spillback*. Although qualitatively spillback effects have been studied extensively (e.g. in the context of dynamic traffic assignment (Ben-Akiva 1998)), quantitative information on the effects of spillback in terms of travel time costs is rare. As an indication, on the Southbound stretch of the densely used 3-lane A13 highway between The Hague and Rotterdam (The Netherlands) the average distance between off- and on ramps is about 2,5 kilometers, while the average queue length on a yearly bases is about 3,3 kilometers (AVV 2002), with regular extremes of up to 8-10 kilometers (Van Zuylen & Muller 2002). Inherently, many vehicles probably spend (unnecessary) time in queues caused at locations downstream of their own destination. Note that the prevailing conditions (measured by traffic data collection systems) on the route of interest also fall into the category of network effects.

A third group of demand factors noted in fig. 2.4 are so-called "*population characteristics*". These factors include the regional and temporal differences in traffic composition (e.g. percentage of trucks, commercial vehicles), but also regional and temporal differences in drive attitude and drive style.

Finally, an important class of demand factors are the (potential) effects of traffic information and *advanced traffic information systems* (ATIS). A generally accepted assumption amongst traffic scientists (economists) is that individual road users are rational decision makers who base their choices (e.g. path and departure time) on minimizing their expected costs (in terms of for example travel time), subject to their personal preferences and attitudes and their knowledge and perception of the traffic system. Providing road users with traffic information allows them to make more informed decisions, yielding not only cost-benefits for the individual, but potentially also more stable and less congested traffic conditions for all road users. In Fig 2.6 the potential effects of traffic information are schematically outlined. ATIS may influence driver's

behavior on all levels of decision making, both pre-trip and en-route. Traffic control systems, which may be classified as supply factors (see below) predominantly operate on en route driver behavior, but may - in the long run - also persuade travellers to change their more strategic (trip or even activity planning) choices.

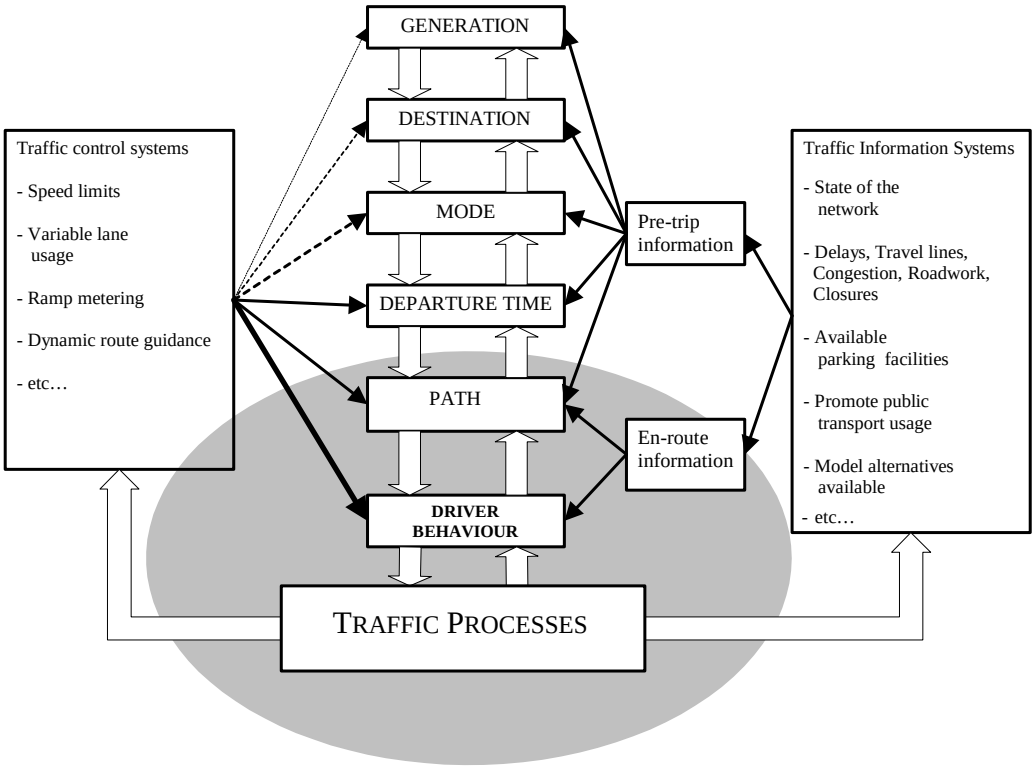


Figure 2.6: Schematic overview of the potential effects of both traffic information and control. The former potentially affects travellers in all their pretrip and en-route choices, while traffic control predominantly affects en-route behaviour (although through experience drivers for example may well avoid those routes with certain traffic control measures).

The potentially beneficial effects of ATIS have been reported in numerous studies in the past decade (e.g. (Khattak, Schofer & Koppelman 1995), (Arnott et al. 1991), and (Mahmassani & Liu 1999)). These studies clearly show the heterogeneity of possible responses to ATIS among different groups of drivers (e.g. commuters, non-commuters) under different (traffic) circumstances, related to for example the quality (accuracy and reliability) of the information provided. In general, if no beneficial effects in terms of cost- or time savings result from the application of ATIS, traffic information at least reduces uncertainty and increases comfort for most drivers (Van Berkum & Van der Mede 1993). But if there are beneficial (direct) effects for particular drivers, the collective effect of these individually optimized (path or departure time) choices can have a positive effect on the experienced traffic conditions, and lead to a more efficient use of the available network infrastructure. Fig 2.7 (adopted from (Van der Zijpp & Lindveld 1999)) shows the direct, indirect and equilibrium effects of the application

of variable message signs (VMSs). In this particular evaluation study (Van der Zijpp & Lindveld 1999), the presence of the VMS panels on the ring road of Amsterdam presenting queue length information was found to lead to a considerable reduction of travel times on a number of corridors in the Amsterdam area. Additionally, a decrease in the travel time variability was found, indicating an improvement of the network's reliability. Both these facts can be cast as indirect effects of demand – supply synchronization.

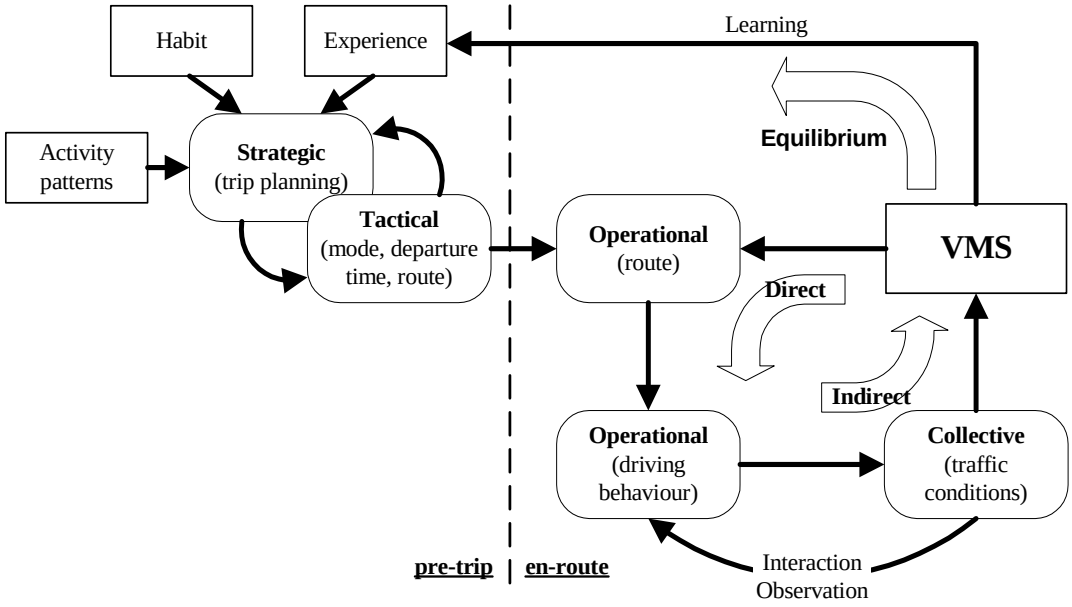


Figure 2.7: Schematic overview of direct, indirect and equilibrium effects of the application of VMS panels providing queue length information (Van der Zijpp & Hoogendoorn 1999)

This improvement can - over longer periods - induce travel-demand and thus increase traffic load, and ultimately also lead to higher travel times. Additionally, drivers may be willing to accept longer expected travel times on a particular route when the reliability of travel time on that route improves significantly. These so-called equilibrium-effects thus partially counterbalance the positive collective effects of the traffic information presented to the drivers (e.g. (Bovy & Thijs 2000), pages 285-296), in terms of travel times.

Finally, let us note that quantification of the effects of traffic information on traffic demand and - indirectly - of travel times is a particularly difficult (non-linear and dynamic) matter.

2.3.2 Factors influencing traffic supply characteristics

As noted earlier, traffic supply reflects the level of service of a particular road facility (section, route or whole network). Traffic supply is hence a much broader term

than capacity, which merely reflects the maximum number of vehicles that can pass a particular facility under prevailing conditions. For example, due to heavy snowfall, travellers will incur higher travel times also in "free-flow" conditions. So even in conditions when traffic demand is much lower than capacity, the travel time of vehicles would still be affected by supply characteristics.

Among supply factors *incidents and accidents* are the most common ones causing non-recurring congestion and hence increasing travel times. Inherent to their (unpredictable) nature, however, these factors do not support travel time prediction. Nonetheless, incident detection has gained considerable attention in the past decade (for example (Zwaneveld et al. 1998)). We argue that fast detection of incidents and accidents from data (e.g. measured by inductive loops) may lead to a considerable increase in the robustness and reliability of online or short term travel time prediction algorithms. Another obvious factor is the occurrence of *roadworks*. Accurate and up-to-date knowledge on time and location of roadworks is conditional to reliable travel time predictions or even estimates based on traffic data collection systems, especially when these do not distinguish between lanes.

In general, traffic supply (and particularly capacity) is strongly related to the geometry and lay-out of the infrastructure facility of interest and the (traffic) regulations that apply to it. As an example, a single lane on a basic 3-lane freeway stretch has a capacity of about 2100 vehicles per hour, while the capacity of a single lane on a freeway stretch with a weaving section may drop well below 2000 vehicles per hour (Transportation Research Board 2000). This is a particularly relevant issue when dealing with data obtained from local detectors. If such a detector is located on a weaving section, upstream or downstream of diverges, merges or on- and off-ramps, the characteristics of its measurements may be significantly different. Homogeneity and stationarity - conditions required for most traffic flow models and travel time estimation techniques (see chapter 3) - certainly do not hold for large regions surrounding detectors in the vicinity for example weaving sections or upstream of merges.

Furthermore, environmental conditions such as weather and luminance affect the supply characteristics of road infrastructure. For instance, results on the effect of weather conditions on freeway capacity have been investigated in (Geuze et al. 1998), and (Van der Vlist 1995). Both reports conclude that heavy precipitation leads to a reduction in road capacity of 10% to 15%. Further research into the effects of weather on road capacity and hence travel time, is still necessary. Note that weather and luminance do not so much physically change the infrastructure but rather complicate the driving task yielding suboptimal use of the available space and thus lower capacity.

A final class of factors influencing traffic supply characteristics is *dynamic traffic management and control*. Examples of these include dynamic speed limits, ramp metering, compulsory route guidance, lane segregation and traffic lights at intersections. The impacts of those measures on traffic supply characteristics (and on traffic demand) are manifold and difficult to interpret independently. We refer to for example (Ben-Akiva et al. 2003) and (Middelham 2001) for examples and references.

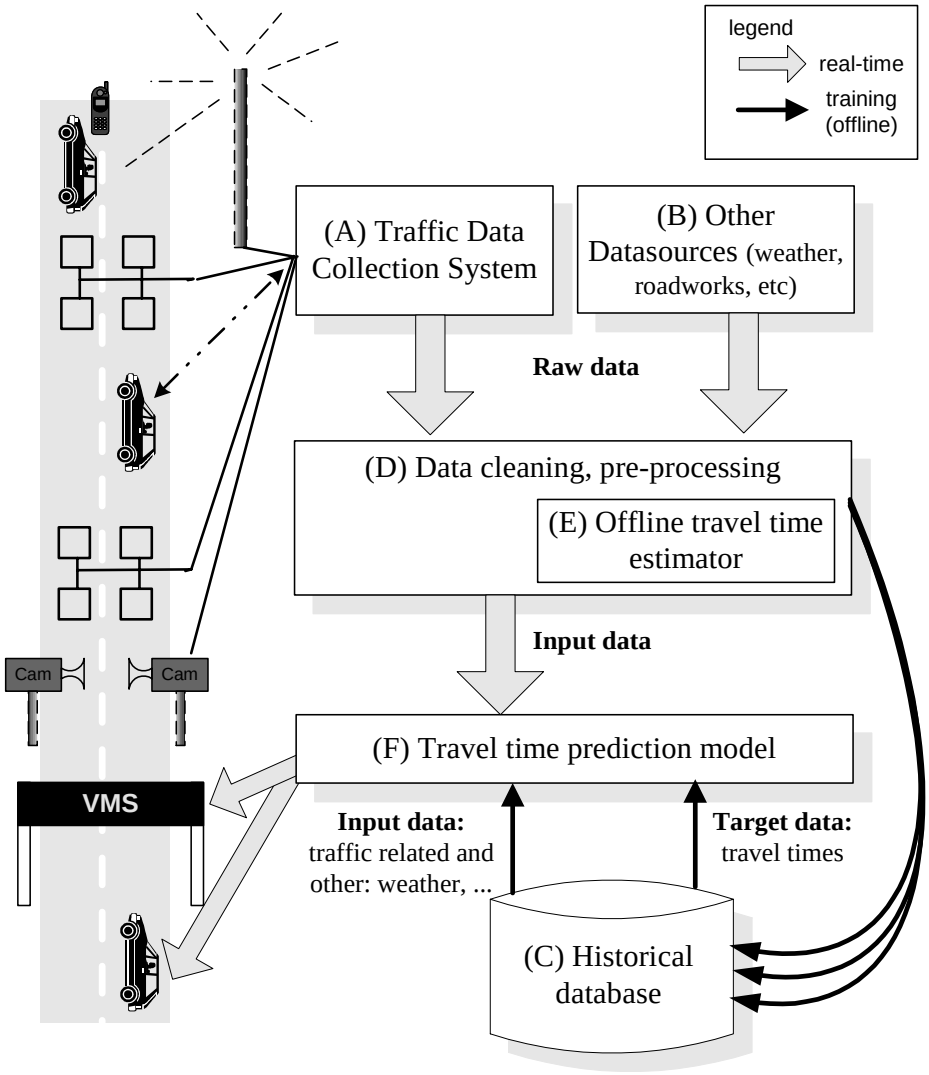


Figure 2.8: General framework for short term freeway travel time prediction

2.4 Framework for Short Term Freeway Travel Time Prediction

The factors influencing travel time presented in section 2.3 should in real life be regarded as functions of both space and time. Moreover, they influence traffic conditions simultaneously, while their relationships are often non-linear, dynamic and (or) stochastic. As noted in the introduction, this thesis focusses on short term freeway travel time prediction models.

This section therefore presents a general framework for short term freeway travel time prediction. As a consequence, this framework, which is shown in Fig 2.8 is the basis of

all subsequent chapters (except chapter 4 - State-of-the-Art). As indicated in fig. 2.8, for the short term travel time prediction problem we can identify two main loops. The first pertains to the offline calibration of the travel time prediction model, the second to its real-time operation. Both exploit the key components which are introduced in the next subsections.

2.4.1 Traffic data collection system and other data sources

The backbone of the short term freeway travel time prediction framework are the data sources which provide for real-time measurements on the basis of which travel time can be predicted. In Fig 2.8 these are denoted as (A) traffic data collection systems, which measure the actual traffic conditions and (B) data collection systems measuring "ambient factors" influencing these traffic conditions, such as weather, roadworks, etc.

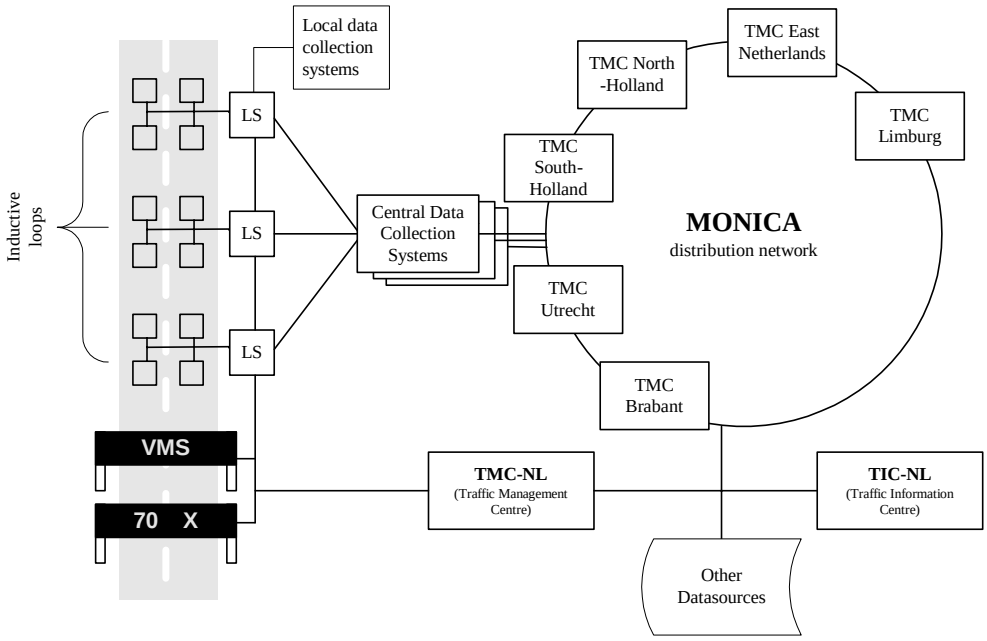


Figure 2.9: The MONItoring CAsco (MONICA) traffic data collection system operational on the Dutch highway network maintained by the Dutch Ministry of Transport, Public Works and Water Management

In this dissertation thesis we build our travel time prediction framework on a typical example of such a traffic data collection system, the MONItoring CAsco (MONICA) system operational on the larger part of the Dutch highway network maintained by the Dutch Ministry of Transport, Public Works and Water Management, see Fig 2.9. The MONICA system comprises of dual inductive loops measuring intensities (number of vehicle passages per minute) and (arithmetic time) mean speeds (also per minute) on

point locations along the highways. In the Randstad area[4] inductive loops are located about every 400 to 800 metres, while on less densely used parts of the network the average distance between inductive loops is about 2000 metres. Spatially distributed servers located at regional Traffic Management Centres collect and store the raw inductive loop data. Both the National Traffic Information Centre (TIC-NL) and the National Traffic Management Centre (TMC-NL) have access to all regional data through a dedicated traffic data communication network. In chapter 3 characteristics and their consequences for travel time prediction of local measurements in general and of the MONICA system in particular are addressed.

Nonetheless, since the travel time prediction model is *data driven*, there is no particular requirement in terms of traffic data collection system. While MONICA measures local traffic variables, also traffic data collection systems measuring variables on section or route level could be used. Examples include automatic vehicle identification (AVI) systems, systems based on probe vehicles (floating car data), camera based systems, which for example match vehicles at two locations by license plate recognition and subsequently deduce (individual) travel times.

2.4.2 Preprocessing module and offline travel time estimation tool

Since in many cases the data will not be 100% correct or even missing to a degree, a preprocessing module (component (D) in fig. 2.8) is imperative to intelligently clean and/or augment missing or corrupt data. In general, such a preprocessing module performs three tasks:

1. **Data Checking**: before possible problems (e.g. missing data) can be adequately tackled, they need to be detected first.
2. **Data Completion**: filling the possible gaps in the data with reasonable replacements
3. **Data Correction**: recheck the now complete data set for validity and consistency and replace / adjust data if required

In chapter 6 a number of algorithms for this purpose are proposed. Missing data and data corruption is a particularly relevant issue in the light of robustness and consequently reliability (two of the criteria for successful ATIS - see previous chapter). A model producing travel times for ATIS should perform well in terms of its predictive accuracy and validity even in case substantial amounts of data are missing. In the MONICA system for example (see also chapters 6 and 8) on average 12% of the data

[4]Denotes the densely populated western part of the Netherlands, and comprises the four major Dutch cities of Amsterdam, The Hague, Rotterdam and Utrecht.

produced by inductive loops is missing, with regular extremes up to 20-25%. The analyst hence needs to balance robustness and predictive accuracy in order to develop a reliable travel time prediction model.

As noted earlier, in case no actual travel times are measured, one needs to resort to tools that can convert those quantities that *are* measured into travel times in order to calibrate and validate travel time prediction models. In chapter 3 such an offline travel time estimation algorithm (component (E) in fig. 2.8) is presented, which can be implemented in case traffic data collection systems measure local quantities such as local mean speeds and intensities. In the same chapter methods and algorithms are developed to deal with some of the typical problems associated with local quantities, such as approximating space mean speed (which is required for speed based offline travel time estimation) from arithmetic time mean speed (which is for example produced by MONICA).

2.4.3 Historical database

Given actual (mean) travel times on a particular freeway route (or offline estimated travel times!), measurements from a traffic data collection system along that route and a set of data from other sources (weather, roadworks, etc.), a large scale historical database (component (C) in fig. 2.8) can be built for a large number of departure time periods. In such a database, for each freeway route and time period a record can be obtained containing the actual (or offline estimated) travel time for vehicles departing in that time period and measurements from all connected traffic data collection systems and other data sources in the same time period. The design and maintenance of such a database is not straightforward, since the data from various data sources may significantly differ in a number of ways. The two most important differences in terms of modeling are the spatial and temporal resolution (and semantics) of the data.

Spatial representation Mean or individual travel times reflect traffic data on section or even route level (line segments), local detection data (time mean speeds, flows) reflect data at single locations (points), while for example precipitation indicators (e.g. from radar equipment at weather stations) reflect information for entire areas of sometimes several squared kilometers. Since the ultimate goal of a travel time prediction model for ATIS is to predict travel time on a freeway route, the aforementioned differences inherently require a GIS[5] type of approach in order to select and link the various data sources used in the model. The basis for each data set (e.g. for calibration or validation) is the geographical lay-out of route of interest. The analyst then consequently selects those data (either reflecting points, lines or areas) associated with the route.

[5]Geographical Information System

Temporal aggregation Local detection equipment usually produces regular temporal aggregates (mean speed, intensity, occupancy), for time periods ranging from some tenths of seconds to several minutes (or even longer), depending on application. Conversely, some data sources may produce individual data (e.g. AVI, floating car data or toll registration systems), while other data sources may produce data in an event-based manner (e.g. bridge openings, roadworks, but also for example local traffic control equipment). The principal problem is that not in all cases one level of temporal aggregation is possible or even desirable. However, for an analyst designing a travel time prediction model a consistent temporal resolution of the data is desirable.

In this dissertation thesis, we will demonstrate the framework based on the MONICA traffic data collection system, which contains local detection equipment (inductive loops) only. The framework presented here, does not restrict the models and methods developed to be applied on local traffic data collection systems only. In the last decennia, with the advent of new Information and Communication Technologies (also in-car) a gradual shift can be observed toward traffic data collection systems that collect both individual as well as aggregate properties of traffic streams over space. Examples include automatic vehicle identification systems (AVI), floating car data collection systems (based on GPS, GSM, or otherwise), and video-based traffic data collection systems. Advanced image processing algorithms enable measurement of macroscopic traffic properties over space and time such as speed, density and flow but also the retrieval of detailed microscopic properties (individual speeds and trajectories) of traffic streams (Chen Shu et al. 2002), not only for car traffic but also for example pedestrian traffic (Daamen & Hoogendoorn 2003).

2.4.4 Travel time prediction model

The central component in the proposed framework is inevitably the data driven short term freeway travel time prediction model (component (F) in fig. 2.8), which predicts travel times on the freeway route of interest based on current and / or near-past traffic conditions. Typically, a data driven model requires (offline) calibration and validation before it can be operated in practice. It is this requirement that makes it necessary to obtain a (large) historical database of actual (or estimated) travel times. Without these so-called target data, it is not possible to calibrate (in neural network terminology "train") or validate ("test") a model. In chapters 5 to 8 a neural network based travel time prediction model is developed, calibrated and validated extensively on the basis of simulated and real data.

The travel time prediction models topology developed in chapter 5 evolves naturally from the locations of the individual detectors with which the route of interest is equipped. However, a similar (straightforward) design approach is possible also when dealing with other types of data sources, that may be characterized by different spatial representations and temporal resolutions (see chapter 9).

2.5 Summary

This dissertation thesis focusses on short term prediction of mean travel times on freeways for Advanced Traffic Informations Systems (ATIS), such as variable message signs (VMSs). To this end, this chapter presented definitions for individual and mean travel time, travel time estimation and travel time prediction. We also illustrated the key differences between travel time estimation and prediction. The former pertains to reconstructing travel times from "known" traffic conditions, while the latter pertains to predicting travel times for unknown (future) traffic conditions. As we will see in the next chapter, this implies that travel time prediction is in fact equivalent with predicting traffic conditions, which is a complex task, given the highly complex and dynamic interplay of factors that influence these traffic conditions. We proposed to categorize these factors as factors influencing traffic demand (e.g. daily, weekly and monthly activity patterns, composition of the population) and factors that influence traffic supply (infrastructure capacity), such as roadworks and weather conditions. Based hereon, we proposed a framework for short term freeway travel time prediction, which is the basis for all subsequent chapters.

As argued in the introduction, the key requirements for reliable travel time prediction models to be applied in a real-time environment are predictive accuracy, validity, robustness with respect to missing or corrupt input data and adaptivity. The next chapter introduces the main mathematical relations that connect travel time with other traffic quantities and focusses on one particular component of the short term travel time prediction framework: an accurate offline travel time estimation algorithm.

Chapter 3

The Freeway Travel Time Estimation Problem

3.1 Introduction

In this chapter we explore the freeway travel time estimation problem. As explained in the previous chapter, *travel time estimation* pertains to reconstructing travel times of realized trips from known traffic conditions, while *travel time prediction* pertains to deriving travel times for unknown (future) traffic conditions. We already established that in case we do not measure travel time directly, an accurate travel time estimation tool is of vital importance to compile databases of (estimated) travel times with which subsequently travel time prediction models can be calibrated and validated. This alleviates one from large scale (real-time) travel time measurements. Also, a thorough understanding of travel time estimation provides insight into the processes which determine travel time and hence enables one to better solve the problem of travel time prediction.

To this end, this chapter introduces in section 3.2 the theoretical relationships that connect mean travel time with other traffic quantities such as vehicular density (the number of vehicles per unit space), vehicular traffic flow (the number of vehicles passing a location per unit time) and vehicular speed under very restrictive conditions. These relationships are required for travel time estimation. Furthermore, in section 3.3 it is shown that traffic data collection systems (such as the MONICA system) do not always measure those quantities that are theoretically needed to correctly estimate travel times, and that we need to resort to traffic flow theory or heuristics to either circumvent or correct these difficulties. Next, in section 3.4 we present an improved travel time estimation procedure (the piece-wise linear speed based (PLSB) trajectory method) that exploits the mathematical relationships between speed and travel time mentioned above. In later chapters we will extensively use the PLSB method as a component in the short term travel time prediction framework.

3.2 Basic Relationships between Travel Time and other Traffic Variables

The ensuing sections present the basic mathematics of traffic processes and variables related to travel time and are based on (Leutzbach 1987), (Daganzo 1997), and (Hoogendoorn et al. 2003). They are provided as background for the second part of this chapter which presents a speed based travel time estimation algorithm and algorithms to correct for the bias caused by time mean speeds.

3.2.1 Individual motion, speed and travel time

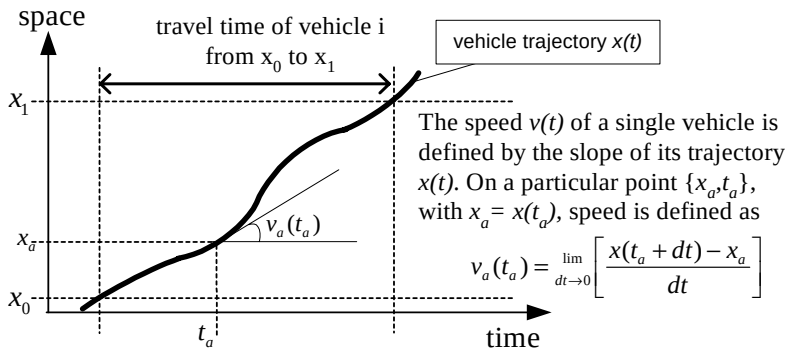


Figure 3.1: The motion of a single vehicle: its trajectory $x(t)$.

Recall the longitudinal motion $x_i(t)$ and speed $v_i(t)$ of a single vehicle i can be expressed as a function of time (Leutzbach 1987):

$$x_i(t) = x_i(t_0) + \int_{t_0}^{t} v_i(s)ds \tag{3.1}$$

where

$$v_i(t) = \frac{dx_i(t)}{dt} = v_i(t_0) + \int_{t_0}^{t} a_i(s)ds \tag{3.2}$$

in which $v_i(t)$, and $a_i(t)$ denote the speed and acceleration (change in speed) of vehicle i as a function of time respectively. fig. 3.1 shows this vehicle trajectory graphically. Conversely, time $t_i(x)$ can also be expressed as a function of space. Given two locations x_0 and x_1, the difference $t_i(x_1) - t_i(x_0)$ then gives the travel time of vehicle i

between these two locations. For this, consider speed as a function of space, that is $v(x) = 1/\frac{dx}{dt} \Rightarrow \frac{v(x)}{dx} = \frac{1}{dt}$, and hence $dt = \frac{dx}{v(x)}$. Analogously to eqn 3.1, this yields

$$t_i(x) = t_i(x_0) + \int_{x_0}^{x} \frac{1}{v_i(r)} dr \tag{3.3}$$

suppose the speed of a vehicle during his journey is constant $v_i(t) = v_i(x) = v_i^0$([1]), the inverse of eqn 3.4 then reads (see (Leutzbach 1987), pages 10-12).

$$t_i(x) - t_i(x_0) = \frac{(x - x_i(t_0))}{v_i^0} \tag{3.4}$$

which reduces to L_k/v_i^0 when travel time for a section k with length L_k is considered. Thus, the travel time for a single vehicle can be unambiguously derived based on speed (and acceleration) expressed as a function of space. Note that strictly speaking, eqns 3.3 and 3.4 are not functions, since as a vehicle comes to a complete stop $t_i(x)$ is undetermined.

3.2.2 The relationship between mean speed, flow and mean travel time

As outlined in the previous chapter, an analyst is not always interested in deriving individual travel times, but rather mean travel times for a sample of vehicles departing in some time period $p = [t_0, t_1]$, see fig. 3.2. Let vehicular traffic flow q denote the number of vehicles passing a location per unit time and vehicular density ρ the number of vehicles present on a unit space at a specific time instant. The former is generally defined as a *local* variable, meaning that it can only be observed on a specific location, the latter as an *instantaneous* variable, meaning that it is defined per unit space at one particular time instant (see fig. 3.2). Fundamental to the analysis below is the assumption of both stationarity and homogeneity on section $k = [x_0, x_1]$ and period $p = [t_0, t_1]$ (space time region $[x_0, t_0] \times [x_1, t_1]$), i.e.

$$q(x,t) = q \; and \; \rho(x,t) = \rho \tag{3.5}$$

Traffic flow, density and travel time

Note in this section mean travel time on section k is not defined for vehicles *departing at* time period p but for vehicles *traversing* section k *in* (during) time period p.

[1] For a vehicle that traverses a section in τ_{ik} time units this constant speeds equals its average journey speed:

$$v_i^0 = L_k/\tau_{ik}$$

.

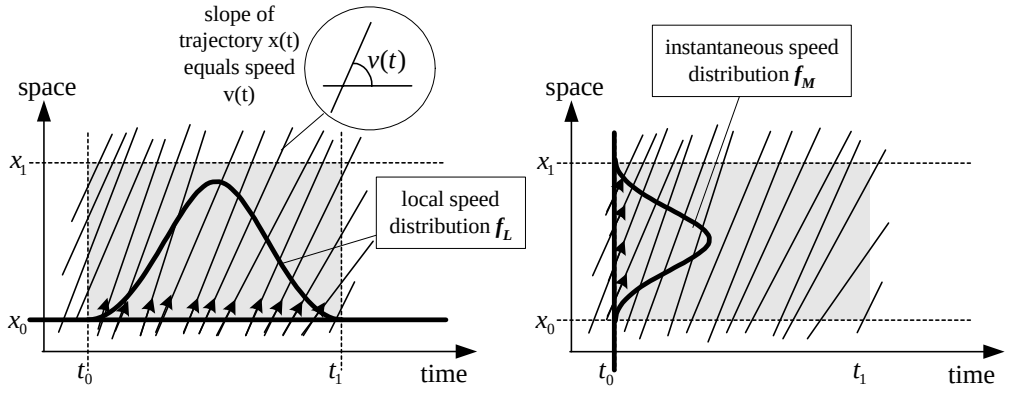
Density and flow The most straightforward theoretical approach is to utilize the fact that in a stationary and homogeneous state on section k during period p (eqn 3.5) the following continuum relation holds[2]

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

that is, mean travel time per unit space or *slowness* w (Leutzbach 1987) equals the (mean) number of vehicles present per unit space divided by the (mean) number of vehicles passing per unit time. As a result, the mean travel time of vehicles traversing section k during period p equals

$$\tau_{kp} = wL \tag{3.6}$$

with $L_k = x_1 - x_0$ denoting the length of section k.



On some location x, the following (continuous) traffic variables can be identified during time period $[t_0,t_1]$:

- speeds v [m/s] are distributed with probability density function f_L. The local mean speed equals

$$\langle v \rangle_L = \int v \cdot f_L dv$$

- traffic flow q [veh/s] is defined as the number of vehicles passing x per unit time

On some time instant t, the following (continuous) traffic variables can be identified over section $[x_0,x_1]$:

- speeds v [m/s] are distributed with probability density function f_M. The space mean speed equals

$$\langle v \rangle_M = \int v \cdot f_M dv$$

- traffic density ρ [veh/m] is defined as the number of vehicles present at time instant t per unit space

Figure 3.2: Local and instantaneous traffic variables. Both graphs show multiple vehicle trajectories $x(t)$.

Cumulative curves The second approach to (theoretically) derive travel times from the basic traffic variables is by means of so-called cumulative curves (see chapter 2 of (Daganzo 1997), an example of its real-time application is described in (Nam & Drew 1996)). Let $N(t, x)$ denote cumulative number of vehicles that pass a particular cross-section x by time t starting from some initial time instant (e.g. $t = t_0$). Furthermore,

[2] for proof see pages 46 to 47. Relation (3.6) follows from substituting eq (3.19) in eq. (3.14).

let $N^{-1}(n, x)$ (the inverse of the cumulative curve) denote the passage time of vehicle n on that particular location x. If FIFO (First In First Out) applies and $N(t, x)$ is recorded at two locations x_0 and x_1 then the travel time of the n^{th} vehicle equals

$$\tau_n = N^{-1}(n, x = x_1) - N^{-1}(n, x = x_0) \tag{3.7}$$

If FIFO does not hold (which is generally the case) we can still use eqn 3.7 if we interpret each observation n not as the n^{th} individual vehicle, but rather as the n^{th} vehicle position in the traffic stream. By definition $N(t, x)$ is a discrete function, since vehicle passages are discrete events. Let $\widetilde{N}(t, x)$ denote a smooth approximation to $N(t, x)$, such that its first partial derivatives exist. These partial derivatives then equal (Daganzo 1997)

$$\frac{\partial}{\partial t}\widetilde{N}(t, x) = q(t, x) \tag{3.8}$$

$$\frac{\partial}{\partial x}\widetilde{N}(t, x) = -\rho(t, x) \tag{3.9}$$

Thus, the partial derivatives with respect to time and space of the continuous cumulative function in fact equal traffic flow (the number of vehicles passing a location per unit time) and (the negative of) vehicular density (the number of vehicles on a specific time instant per unit space) that were introduced in the previous subsection. In the continuous case, we can deduce that the total time (TTS) spent by vehicles in section x_0 to x_1 during time period $[t_0, t_1]$ equals the area between the two cumulative curves recorded at those locations (fig. 3.3) and can be written as

$$TTS(p) = \int_{t_0}^{t_1} \left[\widetilde{N}_1(t) - \widetilde{N}_0(t)\right] dt = \int_{t_0}^{t_1} \widetilde{Q}(t) dt \tag{3.10}$$

where we have simplified notation with $\widetilde{N}_0(t) = \widetilde{N}(t, x = x_0)$ and $\widetilde{N}_1(t) = \widetilde{N}(t, x = x_1)$ and where $\widetilde{Q}(t)$ denotes (the smoothed version of) the accumulation of observations between x_0 and x_1. The mean travel time τ_{kp} for vehicles traversing section k during period p then equals the total time spent divided by the total number of vehicles that actually entered and reads

$$\tau_{kp} = \frac{TTS(p)}{\widetilde{N}_0(t_1) - \widetilde{N}_0(t_0)} \tag{3.11}$$

Alternatively, one can express the total delay $DT(p) = TTS(p) - \tau^{free}$ for vehicles traversing section k during period p which leads to

$$\tau_{kp} = \tau^{free} + DT(p) \tag{3.12}$$

In case of a basic freeway stretch enclosed by two detectors at x_0 and x_1 with no off- and on-ramps, the mean travel time in time period p on that freeway stretch approximately equals (compare eqn 3.11)

$$\tau_{kp} = \frac{t_1 - t_0}{N_0(t_1) - N_0(t_0)} \left[\frac{1}{2}\left[N_0(t_1) - N_0(t_0)\right] + \frac{1}{2}\left[N_1(t_1) - N_1(t_0)\right]\right] \tag{3.13}$$

where again we have simplified notation with $N_0(t) = N_0(t, x = x_0)$ and $N_1(t) = N_1(t, x = x_1)$. eqn 3.13 approximates total time spent by assuming a linear increase in both curves during time period p.

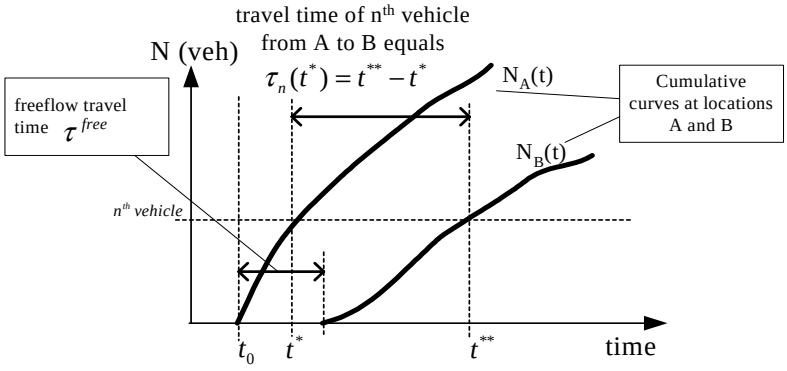


Figure 3.3: The relationship of travel time with cumulative curves.

Mean speed and travel time

Space mean speed and travel time The third route to (theoretically) derive mean travel times is through mean speeds. Assume that (instantaneous) speeds at time instant t are drawn from a speed distribution $f_M(v)$, and (local) speeds at a location x are drawn from a distribution $f_L(v)$. Then, given homogeneity and stationarity (e.g. (Leutzbach 1987)), the following continuous relation holds

$$\rho u_M = q \tag{3.14}$$

where by definition

$$u_M = \int v f_M(v) dv \tag{3.15}$$

is the expected instantaneous speed or *space mean speed*. In words, eqn (3.14) states that in a stationary and homogeneous state the number of vehicles that pass a particular location (q) equals the number of vehicles per distance unit (ρ) times the distance each vehicle passes per unit time (v). Given these assumptions eqn (3.14) allows to derive one of the three variables out of the other two. Furthermore, in a stationary and homogeneous state the space mean speed may be substituted with the local harmonic mean speed

$$u_M = \widetilde{u}_L = \frac{1}{\left\langle \frac{1}{v} \right\rangle_L} \tag{3.16}$$

Proof. By definition, the probability of finding a vehicle in region $[x_0, x_1]$ at time instant t driving with a speed in the interval $[v, v + dv]$, $dv \ll v$ equals

$$(x_1 - x_0)\rho\, f_M(v)dv$$

Similarly, the probability of finding a vehicle at a location x during time period $[t_0, t_1]$ driving with a speed in the interval $[v, v + dv]$, $dv \ll v$ equals

$$(t_1 - t_0)q\, f_L(v)dv$$

Since these probabilities are of equal magnitude, we have

$$\rho\, f_M dv = \frac{q}{v} f_L dv \tag{3.17}$$

integrating over v on both sides leads to

$$\rho = q \left\langle \frac{1}{v} \right\rangle_L \tag{3.18}$$

which equals eqn (3.14) and completes the proof ■

Now it is possible to derive an expression for the mean travel time of vehicles present on k during p. Assume that the traffic stream (q, k) can be decomposed into n homogeneous groups of vehicles driving with constant speeds v_n , such that

$$\begin{aligned} \rho_n &= q_n/v_n, \forall n \\ \rho &= \sum \rho_n \\ q &= \sum q_n \end{aligned}$$

During time period p a total of $q_n T$ vehicles of class n traverse the section with speed v_n,with $T = t_1 - t_0$. The mean travel time of these vehicles then equals L_k/v_n, with $L_k = x_1 - x_0$. The mean travel time of all vehicles then equals

$$\begin{aligned} \tau &= \frac{\sum q_n T(L_k/v_n)}{\sum q_n T} \\ &= \frac{nTL\sum \rho_n}{nT\sum q_n} \\ &= \frac{\rho}{q} L_k \end{aligned}$$

Given eqn (3.14) this implies that the mean travel time simply is the inverse of the space mean speed eqn (3.15) times the length of the section and reads

$$\tau_{kp} = \frac{L_k}{\langle v \rangle_M} = \frac{L_k}{u_M} \tag{3.19}$$

or, with (3.16)

$$\tau_{kp} = L_k \left\langle \frac{1}{v} \right\rangle_L = \frac{L_k}{\widetilde{u}_L} \tag{3.20}$$

Thus, also mean travel time can be easily and unambiguously derived with speeds given that stationary and homogeneous traffic conditions are assumed in space time region $[x_0, t_0] \times [x_1, t_1]$, and either local harmonic averaged speeds on a cross section $x \in [x_0, x_1]$ or space mean speeds on section $[x_0, x_1]$ are collected.

Time mean speed and travel time Evidently, in a homogeneous and stationary state (3.5) the space mean speed on $[x_0, x_1]$ is equal to the local harmonic mean speed on a cross-section $x \in [x_0, x_1]$ over time period $[t_0, t_1]$. Given those assumptions we can also derive an analytical relationship between the arithmetic time mean speed and the space mean speed. By definition the arithmetic time mean speed or local mean speed (at for instance a detector) equals

$$u_L = \langle v \rangle_L = \int v f_L dv \tag{3.21}$$

First, combining local and instantaneous distributions, eqn (3.17) and (3.14) gives

$$f_L = \frac{v}{u_M} f_M \tag{3.22}$$

That these two distributions have quite different shapes, especially at speeds $<$50 km/h, is illustrated by fig. 3.4. In this case the local distribution has higher mean and variance than the instantaneous one. The data (individual vehicle passages) for this fig. were recorded on a detector location the A9 highway from Alkmaar to Amstelveen (the Netherlands) on 13 weekdays in Oct. 1994. Substituting eqn (3.22) in (3.21) gives

$$u_L = \int \frac{v^2}{u_M} f_M dv$$

which can be rewritten as

$$u_L = \int \frac{(v - u_M)^2 + 2v u_M - u_M^2}{u_M} f_M dv = \frac{\sigma_M^2}{u_M} + u_M \tag{3.23}$$

in which by definition the variance of the instantaneous speeds equals

$$\sigma_M^2 = \int (v - u_M)^2 \, f_M dv \tag{3.24}$$

eqn 3.23 states that space mean speed is always equal or smaller than (local) time mean speed, a difference which is proportional to speed variance. In other words, the time

mean speed overestimates the space mean speed if instantaneous speed variance is nonzero. The explanation to this phenomenon is straightforward. Observations of high speeds occur more frequently in a time sample than "slow" observations, implicating that time mean speeds will inherently be biased toward higher speeds, unless each observation has equal and constant speed over the road section of interest.

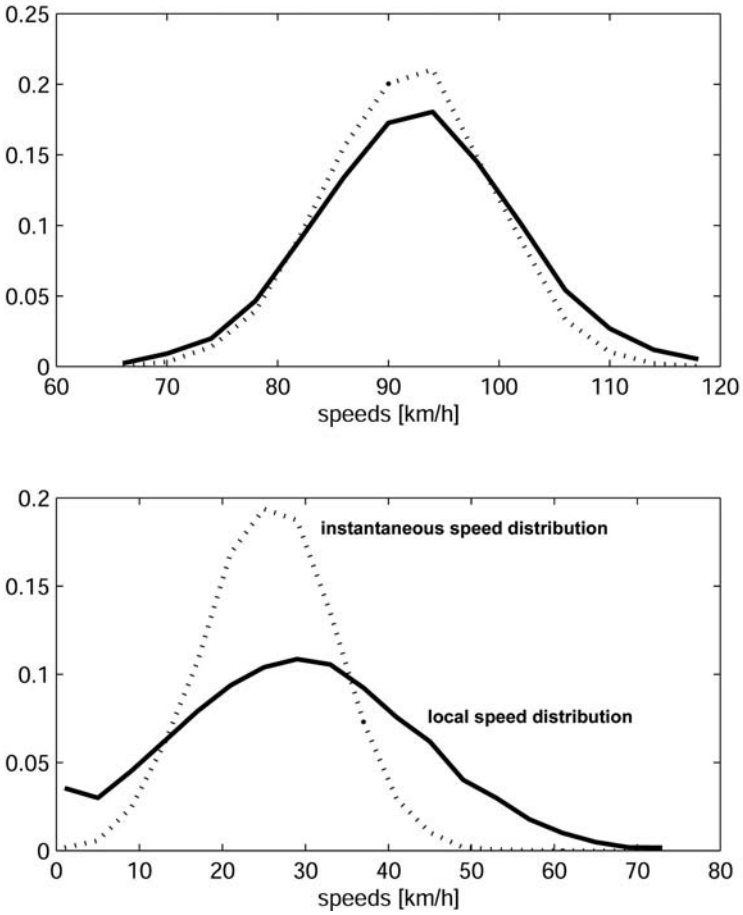


Figure 3.4: Estimated local and momentane speed distributions. In free flowing conditions (top) the two speed distribtutions are alike, while in congested conditions (bottom) they clearly have a different shape. The data (individual vehicle passeges) were recorded on the A9 highway from Alkmaar to Amstelveen on 13 weekdays in Oct. 1994.

As Hoogendoorn shows (in (Hoogendoorn 1999), pages 186-187), the instantaneous variance σ_M^2 can be approximated by taking the harmonic average of the squared dif-

ferences of subsequent individual local speed measurements:

$$\sigma_M^2 = \widehat{\sigma}_M^2 = \frac{1}{2N}\sum_{i=0}^{N-1}\beta_i(v_{i+1}-v_i)^2 \tag{3.25}$$

$$\beta_i = \frac{1}{v_i}\left(\frac{1}{N}\sum_{j=0}^{N-1}\frac{1}{v_j}\right)^{-1} \tag{3.26}$$

where N denotes the number of subsequent observations, and β_i a weighting factor for each observation. As can be seen in (3.26) this weighting factor multiplies the reciprocal of each speed observation ("slowness") with the harmonic mean speed of all observations. What it effectively does is that measurements of slower vehicles are weighted more heavily than fast vehicles, which counter-effects the fact that in time averages slow observations are underestimated (because they occur less frequent during an observation period). In cases individual speed measurements are available, σ_M^2 will be calculated by means of eqn (3.25).

3.2.3 Discussion on theoretical relationships

A couple of remarks need to be made before these theoretical relationships are put to practice. First of all, the assumption of stationarity and homogeneity will most likely not be satisfied on road stretches of any practical size (a couple of hundred metres), especially not in congested and / or unstable traffic in which stop and go waves occur (Helbing 1997). The problem is, however, that the basic mathematics with regard to traffic flow variables q, ρ, and u_M described above are valid *only* on the basis of these assumptions. Relaxing these assumptions and replacing them with more realistic ones would yield more accurate travel time estimates, but would also require a different and much more complex approach. Nonetheless, as we will see in section 3.4, it is possible to make more realistic assumptions (in which for example speed changes gradually over space instead of discontinuously), without having to use a different set of mathematical tools than the ones described above.

The second remark relates to flow and density based travel time estimation. Arguably, the cumulative curve approach (section 3.2.2) is theoretically the most appealing and practically the easiest method for deriving travel times. Its principal benefit is that it does not require anything else but cumulative flow counts at subsequent locations to deduce mean travel times on the stretch between those locations. This requirement, however, is very strict, in that the detector equipment MUST record every vehicle passage, including those of vehicles exiting or entering off and on-ramps. In later chapters we will see that these requirements in practice are rarely met since

- Usually not all entry and exit points are properly equipped with detectors (detectors on a weaving section for example measure a mix of entering and exiting vehicles).

- Detectors (certainly inductive loops) typically suffer from a significant degree of random and or structural failure (see chapter 6), but also a degree of intrinsic failure (miscounts, double counts, rounding off errors, etc.).

The latter is illustrated with the results of a field-test on the detection quality of one of the inductive loops along the A13 southbound freeway between The Hague and Rotterdam (the Netherlands). In this test one hour of minute-aggregate flow counts from this detector are compared against observations made with video cameras. Fig. 3.5 shows a histogram of the relative errors[3] the detector makes during a one hour period in the afternoon. For this (small) data set (65 observations) a mean relative error of 5.9% with standard deviation of 10.4% is recorded, which roughly indicates[4] that in 95% of the cases one (plus or minus 4) out of twenty cars is either missed by the detector or conversely, counted double. A similar test on another loop detector along the same route yielded similar results. Given these particular detectors are representative for these types of inductive loop detectors, significant effort is required in terms of data cleaning and correction in order to estimate travel times based on flows.

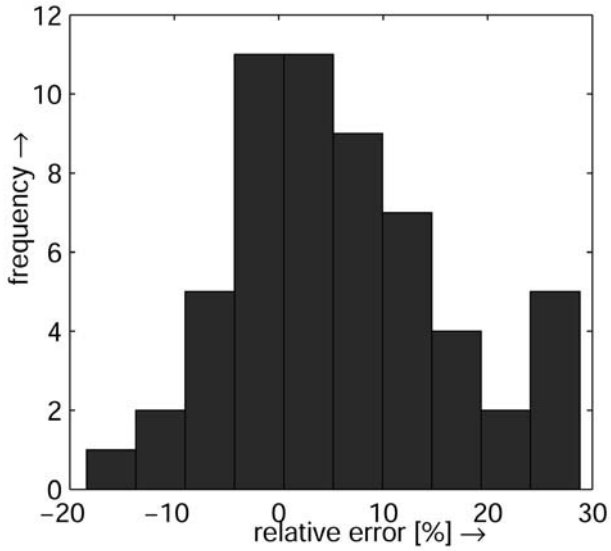


Figure 3.5: Histogram of relative miscount errors at a local detector. Data from inductive loop and camera observations obtained at the three-lane southbound stretch of the A13 highway from The Hague to Rotterdam at location 8050 [m].

Besides the cumulative curve method, there are also a number of alternative flow based travel time estimators, most of which are based on the principle of conservation of vehicles and / or first order traffic flow theory. Examples can be found in (Nam & Drew 1996), (Petty et al. 1998), and (Buisson et al. 1998). In general, most flow based algorithms are parameterized and require location specific calibration. These parameters relate to modelling assumptions (e.g. fundamental diagram[5]), or to algorithms that

[3] relative error $= 100 \frac{\text{no. observed - no. detected}}{\text{no. observed}}\%$

[4] assuming the relative errors are normally distributed

[5] see e.g. appendix D

correct for the errors due the problems listed above. Speed based algorithms are essentially non-parameterized and more generic given the direct (reciprocal) relationship of speed and travel time. They are also more robust since they depend on a mean quantity and not on correct counting of each individual observation. On the down side, they fail in tracking still-standing traffic (zero speeds can not be measured locally). Nonetheless, in this thesis a choice is made to estimate travel times through speeds based on practical reasons. This does not imply that an approach based on flows is unfeasible, it is a specific choice made for this dissertation thesis.

3.3 Traffic Data Collection Systems and their Characteristics with Respect to Travel Time

3.3.1 Brief overview

For a comprehensive overview of traffic data collection systems we refer to among others (Westerman 1995), and (Michalopoulos & Hourdakis 2001). Table 3.1 summarizes briefly a number of detection systems available today. Each of the traffic detection systems presented in table 3.1 can be characterized by amongst other things:

- data semantics (for example: space mean speeds or time mean speeds)
- spatial level of aggregation (for example: distance between inductive loops or cameras)
- temporal level of aggregation (for example: 1 minute or 5 minute aggregates)
- aggregate or individual measurements (for example: detector data versus floating car data)
- availability in terms of frequency (time) and scope (place, link, route)
- accuracy, reliability (probably as a function of time, place and traffic conditions)
- technical aspects (such as data(base) format, communication protocols, etc.)
- infrastructure bound or free (for example: roadside versus in-car GPS/GSM)
- owner / administrator of data (for example: private or public)
- usage cost.

Obviously, their usefulness in terms of travel time estimation (and prediction) strongly depends on these characteristics. In general, for estimating travel times most favorable are densely spaced infrastructure bound detector systems which measure speeds

Table 3.1: Overview (not exhaustive) of traffic detection equipment. Legend: A = aggregate or average values, I = individual values; L = local measurements, X = road section measurements; F: fixed (to road surface), NF = non-fixed; ML = lane specific; MC = vehicle specific; traj = vehicle trajectories, t = passage times, lp = license plates, q = traffic flows, k = traffic densities, o = occupancies, vx = space (mean) speeds, vt = time (mean) speeds, qu = queues and queue lengths.

	Scope	Remarks
Pneumatic tube detector	A/I, F, L, ML, q, vt	Mostly used for temporary measurements (e.g. (Harvey et al. 1993)). When two tubes are placed at a short distance Δx from each other, accurate measurements of speed over Δx can be derived.
Coaxial detector	A/I, F, L, ML, q, vt	Speeds obtained similarly as with pneumatic tube detector.
Infrared detector	A/I, NF, L, ML, q, vt	Detects vehicles and measure speeds through reflection patterns of (infrared) light beams caused by passing vehicles (see e.g. (Hussain Tarik 1995)).
Radar detector	A/I, NF, L, q, o, vt	Detects vehicles and measure speeds through reflection patterns of pulsed, or frequency- or phase-modulated signals (Ervin et al. 2001).
Ultrasonic detector	A/I, NF, L, MC, q, vt	Also capitalizes on deflection patterns in this case of ultra-sonic signals. For technical details we refer to (Jarviluoma & Heikkila 1995).
Laser detection	A/I, NF, L, MC, q, vt	Another non-intrusive detection methodology, based on reflection of laser beams (e.g. (Cheng et al. 2001)).
Induction loop detector	A/I, F, L, ML, MC, q, vt, o	Most widely used detection equipment. Dual loop detectors (see e.g. (Bovy & Thijs 2000), pp 26-27) measure both speeds, flows and occupancies, single loops only the latter two.
Microwave detector	A/I, NF, L, MC, q, vt	Cheap and easy-to-install alternative to inductive loop detection ((Michalopoulos & Hourdakis 2001)). Measures both flow and speeds.
Video / machine vision detection	A/I, NF, L/X, ML, MC, q, vt, o, t, lp, qu	Since there are so many different video camera based detection systems on the market today, theoretically - at least according to commercial leaflets - any traffic property can be measured. Examples of video based camera detection are Autoscope, Golden River, and Traffic Master

and or flows, infrastructure bound systems (cameras, AVI systems) that allow deducing travel times directly, or in car equipment that allows for tracking vehicles along a network (via e.g. GPS, GSM). Since the travel time prediction framework of this dissertation thesis is calibrated and validated with data from a local traffic data collection system (inductive loops), below characteristics of local measurements and particularly inductive loops are discussed.

3.3.2 Characteristics of local measurements

Of the class of local infrastructure bound detection systems inductive loops are most widely used and have been the mainstream detection equipment on both highways and urban roads since the 1950's. For example, *dual* inductive loop detectors (Fig 3.6) are the roadside components in the MONICA (MONItoring CAsco) system installed on most of the Dutch highways, and are maintained by the Dutch Ministry of Transport, Public Works and Water Management (recall fig. 2.9 on page 35).

The symbols used in this section are explained in fig. 3.6. Also note that equations (3.27) to (3.32) are valid for *single* freeway lanes. From each single loop detector intensities (N_p number of vehicles passing a detector within a measurement period T_p) and occupancy (eqn 3.27) can be straightforwardly measured. Occupancy is defined as the fraction of time vehicles occupied the detector during T_p

$$O_p = \frac{1}{T_p} \sum_{i=1}^{N_p} \left(t_{2,i} - t_{1,i}\right) \tag{3.27}$$

Occupancy can also be calculated for the entire width of the dual loop detector (substitute $t_{4,i}$ for $t_{2,i}$ in eqn 3.27). Moreover, a dual loop detector allows for *direct* measurement of harmonic mean speeds (which in stationary and homogeneous conditions equals the space mean speed). Recall that travel time (per unit space) is the reciprocal of speed, hence

$$\frac{d_L}{v_i} = t_{3,i} - t_{1,i} \tag{3.28}$$

Combining eqns (3.28), (3.19) and (3.20) we get

$$\begin{aligned} \frac{d_L}{u_{p,M}} &= \left\langle t_{3,i} - t_{1,i} \right\rangle_L \Leftrightarrow \\ u_{p,M} &= \frac{d_L}{\left\langle t_{3,i} - t_{1,i} \right\rangle_L} = \frac{d_L}{\frac{1}{N_p} \sum_{i=1}^{N_p} \left(t_{3,i} - t_{1,i}\right)} \end{aligned} \tag{3.29}$$

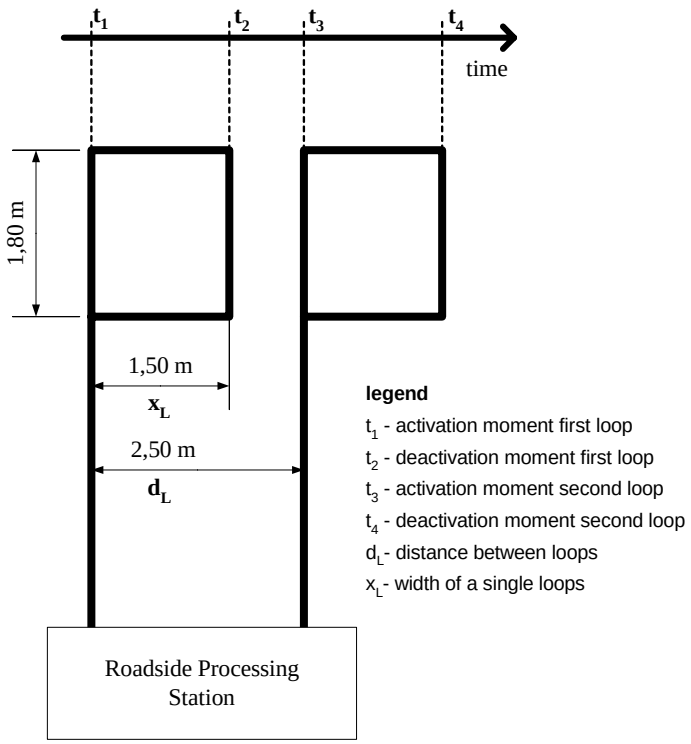


Figure 3.6: Dual inductive loop detector. From the passing of a vehicle over both loops four passage moments are identified and subsequently used for calculation of speeds: $v^i = \frac{2{,}5}{t_3 - t_1}$. Alternatively, one could use t_2 and t_4. (Picture and caption adapted from (Bovy & Thijs 2000) page 27)

Harmonic mean speed can also be derived from a single inductive loop using the continuum relation (3.14). Since single loops do not measure density directly, one has to deduce it from occupancy, for example by means of

$$\rho_p = \frac{O_p}{x_L \cdot L_{veh}} \tag{3.30}$$

in which L_{veh} is the average vehicle length, which is a parameter that needs to be estimated from data (and may be different under different circumstances). eqn (3.30) calculates the average number of vehicles present on the detector (O_p / L_{veh}) during period p and divides it by the width of the detector (x_L) to obtain density per unit space. Since traffic flow q per unit time equals N_p / T_p one can write

$$u_{p,M} = \frac{q_p}{\rho_p} = \frac{x_L N_p L_{veh}}{T_p O_p} \tag{3.31}$$

Remarkably, the inductive loops of the Monica system use neither of the above procedures to obtain mean speeds. Instead, arithmetic time mean speeds are calculated

with

$$u_{p,L} = \frac{1}{N_p}\sum_{i=1}^{N_p} v_i \qquad (3.32)$$
$$v_i = \frac{d_L}{t_{3,i} - t_{1,i}}$$

Besides the obvious theoretical disadvantages of arithmetic time mean speeds (which introduces a non negligible bias as discussed extensively above), this averaging procedure (3.32) is also much more computationally demanding than the "correct" one (3.29). The former requires $N_p + 1$ floating point operations (divisions), the latter only 2 per time period. To make things worse, in MONICA occupancy is also not recorded, which would provide an alternative to derive space mean speeds and also a means to deduce traffic density. At the time of writing, MONICA dual loop detectors provide local arithmetic time mean speeds, which implies that we must correct for the inherent bias when using these to estimate travel times on MONICA equipped freeway routes. In the next section, we analyze the relationship between mean speed and speed variance, which (see eqn 3.23) governs the magnitude of this bias. Subsequently, since dual loop detectors neither measure speed variance, we propose two algorithms to estimate speed variance such as to correct for the bias.

Remark 1 *In the current configuration, not all MONICA detectors distinguish between lanes nor between user classes (trucks and person cars for instance). On the freeway stretch available for this thesis (the A13 Southbound between The Hague and Rotterdam, see e.g. chapter 8) all dual loops measure intensities and arithmetic time mean speeds on a cross section of the entire main carriage way.*

Remark 2 *The alternatives for inductive loop detectors available today, such as radar, microwave or laser detectors, measure in principle the same quantities as the inductive loop detectors in MONICA, and therefore, the analysis presented here applies to these systems too.*

3.3.3 Correcting for bias due to arithmetic mean speeds by estimating speed variance

In this subsection we derive two methods of correcting for the bias due to arithmetic time averaging of speeds. The purpose of these bias correction algorithms is to correct time mean speeds and subsequently use the corrected speeds to estimate travel time. Such a correction algorithm is required if the data collection system available produces time mean speeds only (which is for example the case in the MONICA system). An inherent requirement for a bias correction algorithm is that it must estimate bias with the quantities that *are* measured, that is, arithmetic time mean speed and intensity (traffic flow per measurement period). Since the bias is governed by instantaneous speed

variance, a natural approach is to estimate to instantaneous speed variance, use that estimate to correct speeds and subsequently estimate travel time with these corrected speeds. The correction algorithms thus take the following form:

1. Estimate speed variance σ_M^2 as a function of quantities that are measured, e.g.

$$\widehat{\sigma}_{p,M}^2 = F(u_{p,L}, N_p, ...)$$

2. Solving eqn (3.24) for u_M then gives

$$\widehat{u}_{p,M} = \frac{1}{2}u_{p,L} + \sqrt{\frac{1}{4}u_{p,L}^2 - \widehat{\sigma}_{p,M}^2},\ \widehat{\sigma}_{p,M} \leqslant \frac{1}{2}u_{p,L}$$

3. Use $\widehat{u}_{p,M} \approx u_{p,M}$ for travel time estimation purposes

A desirable property for such a bias correction algorithm is that it contains no or few parameters that are very sensitive to location specific circumstances, since without actual measurements of this bias (or speed variance) it is not possible to calibrate these parameters, specifically not in situations where local detection equipment measure over entire carriage ways. Moreover, the algorithm should exhibit a certain robustness to idiosyncrasies of local measurements, such as miscounts and false counts. Finally, since travel time is the reciprocal of mean speed, the algorithm should be accurate for low speed conditions specifically. To this end, the next subsection first presents some general observations of mean speed and variance, establishing the nature and sign of the bias in various circumstances. In the next two sections we propose two algorithms to estimate speed variance and hence bias. The first uses a naive (linear regression) approach, while the second estimates speed variance through a time series approach and principles from traffic flow theory.

The data set used in this section is kindly provided by Henk Taale of the Dutch Ministry of Transport, Public Works and Water Management, and consists of individual local measurements (from inductive loops) on 13 locations along the 2-lane southbound stretch of the A9 Highway between Amsterdam and Muiden (The Netherlands) during the fall of 1994. Albeit it contains local measurements, recall obtain space mean speed u_M is obtained by calculating local harmonic mean speed (3.16) and speed variance σ_M^2 with the aid of (3.25).

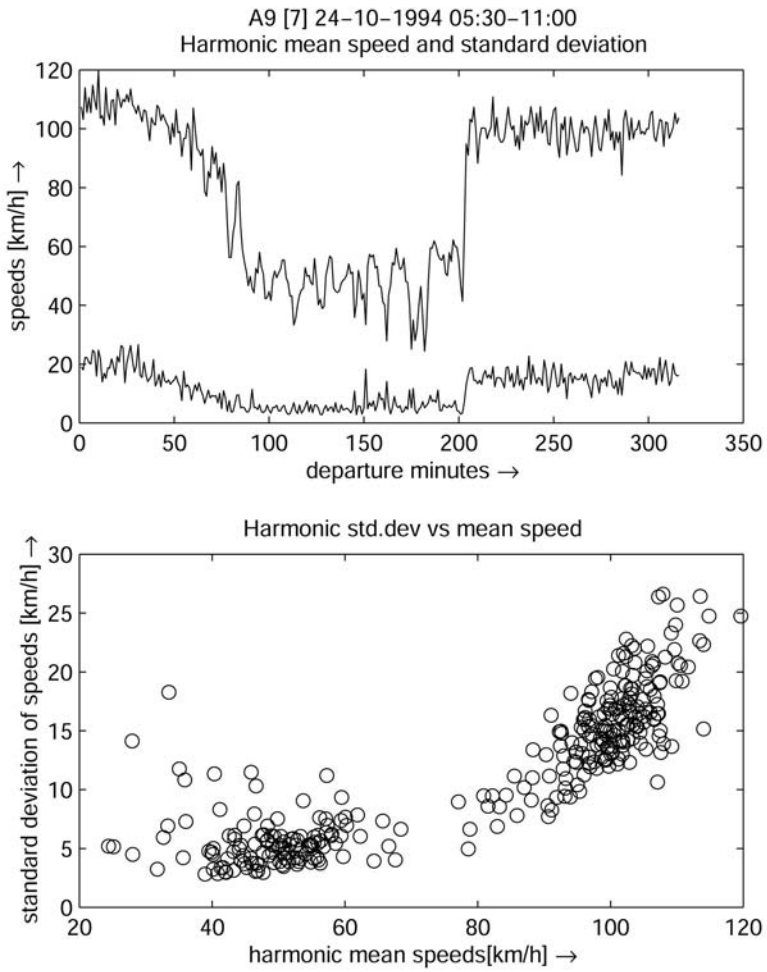


Figure 3.7: Mean and standard deviation of speeds for a typical morning peak. Both graphs clearly show the different behaviour of both mean speed and variance under congested and freeflow conditions (see text). The data was recorded at location 7 (km 37.6) on the A9 highway from Alkmaar to Amstelveen on Friday 25 Oct 1994.

Empirical observations of mean speed and variance

Fig. 3.7 shows mean speed u_M and standard deviation σ_M measured on a typical morning peak averaged over one-minute periods. From the top graph it is clear that as mean speeds drop so does variance. The top and bottom graphs both indicate distinct differences between congested and non-congested conditions. From the top graph it is observed that in congested conditions mean speeds fluctuate more strongly than in free-flow conditions, while speed variance seems almost constant in congested conditions. This is emphasized in the bottom graph, which plots σ_M as a function of mean speeds. Here, a clear distinction can be seen. In this case, in congested conditions u_M fluctuates between 35 and 65 km/h, while σ_M fluctuates around 5 km/h. The erratic behavior of u_M is most probably due to stop- and go waves of congested traffic traversing the detector. The near stationary behavior of σ_M indicates that within an observation

period vehicles are constraint by the speeds of other vehicles. In free-flow conditions, the variability of σ_M is much higher and fluctuating between 5 and 20 km, most probably due to the composition of traffic (a considerable % of trucks). Finally, from the bottom graph, it appears that in free-flow conditions σ_M is an increasing function of u_M.

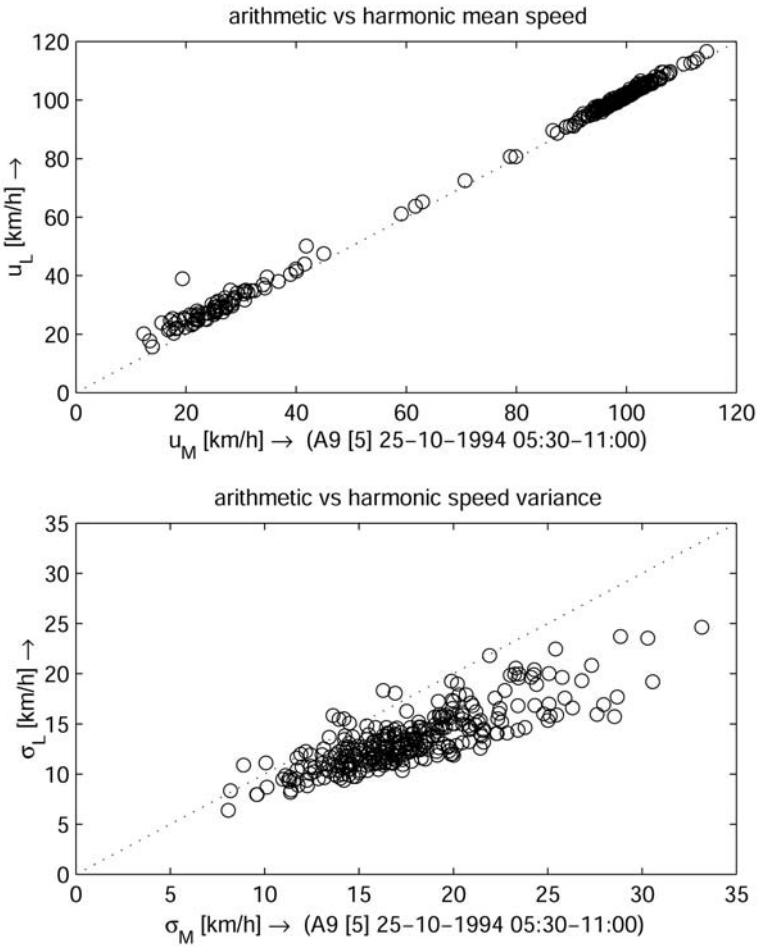


Figure 3.8: Arithmetic mean plotted as a function of harmonic mean (top) and arithmetic variance plotted against harmonic variance (bottom). The graphs indicate that, in general, mean speeds are overestimated by the arithmetic time mean, while variance is underestimated. The data was recorded at location 5 (km 40.8) on the A9 highway from Alkmaar to Amstelveen on Friday 25 Oct 1994.

As fig. 3.8 shows, the arithmetic mean speed has a tendency to overestimate speeds (top), which can be expected based on eqn (3.23), which states that $u_L \geq u_M$. Similarly, the arithmetic variance tends to overestimate the harmonic variance (bottom). This is also due to the fact that faster vehicles are overrepresented in the local speed distribution, yielding (in absolute terms) in larger speed differences and thus larger variance. Relatively this effect is larger in congested conditions than in free flow conditions (compare fig. 3.4).

Fig. 3.9 shows the absolute bias $u_L - u_M$ as a function of u_M (fig. 3.9(a)) and σ_M (fig. 3.9(c)) and the relative bias $(u_L - u_M)/u_M$ in fig. 3.9(b) and (d) as a function of mean speed and variance respectively. From the graphs observe that the bias, which theoretically equals σ_M^2/u_M, is proportional to both mean speed and variance, albeit there are a significant number of outliers. Since variance itself is proportional to mean speed, these observations are in line with theory (eqn 3.23). Quantitatively, for this particular data set, the bias is never larger than 5 km/h and 6%.

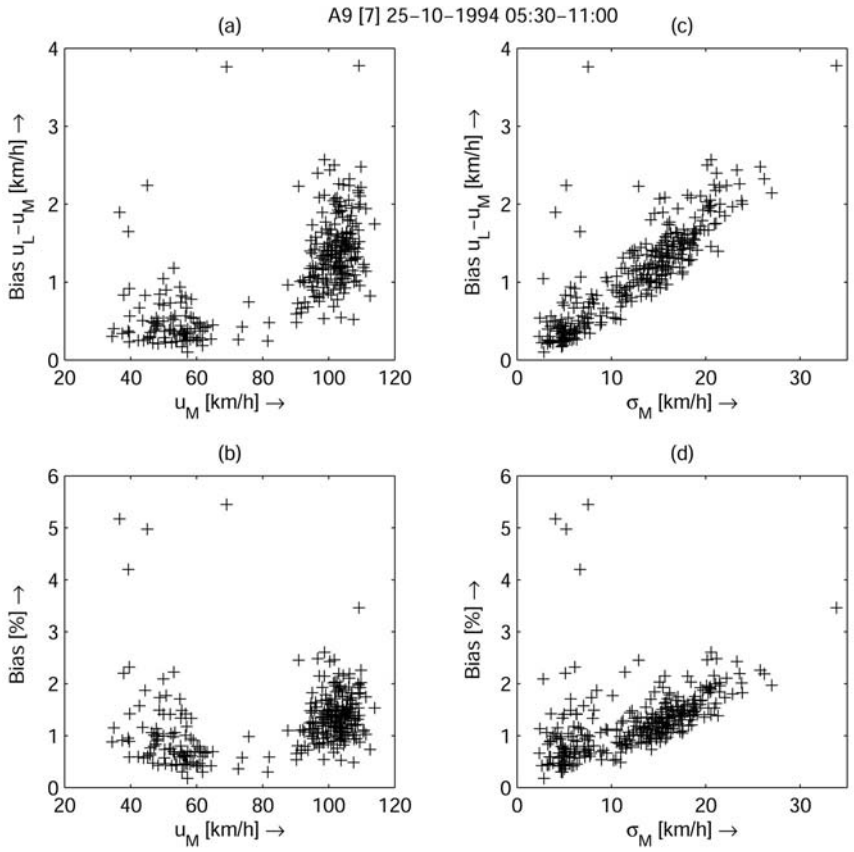


Figure 3.9: The absolute and relative bias caused by arithmetic averaging as a function of harmonic mean speed [(a) and (b)] and variance [(c) and (d)]. The data was recorded at location 5 (km 40.8) on the A9 highway from Alkmaar to Amstelveen on Thursday 24 Oct 1994.

Estimating speed variance through linear regression (naive method)

Recalling Fig 3.7, we can express speed variance as an increasing function of mean speed. In Fig 3.7 (bottom) two distinct branches of this function can be identified. In congested conditions variance is nearly constant, while in free-flowing conditions variance is a steeply increasing function of mean speed. A naive but reasonable approach then is to express standard deviation as a bilinear function of time mean speeds. Each branch is represented with a two parameter linear function while a threshold speed

value distinguishes between congested and free flowing traffic. Based on approximately 13,500 minute observations collected on 12 weekday morning peaks in October 1994 on 4 different detector locations along the 2-lane A9 highway from Amsterdam to Muiden we find the following values that minimize the root mean of squared estimation errors (RMSE).

$$\widehat{\sigma}_M = \begin{cases} 0.5u_L - 34 & u_L \geq 74 \\ 0.02u_L + 5 & else \end{cases} \tag{3.33}$$

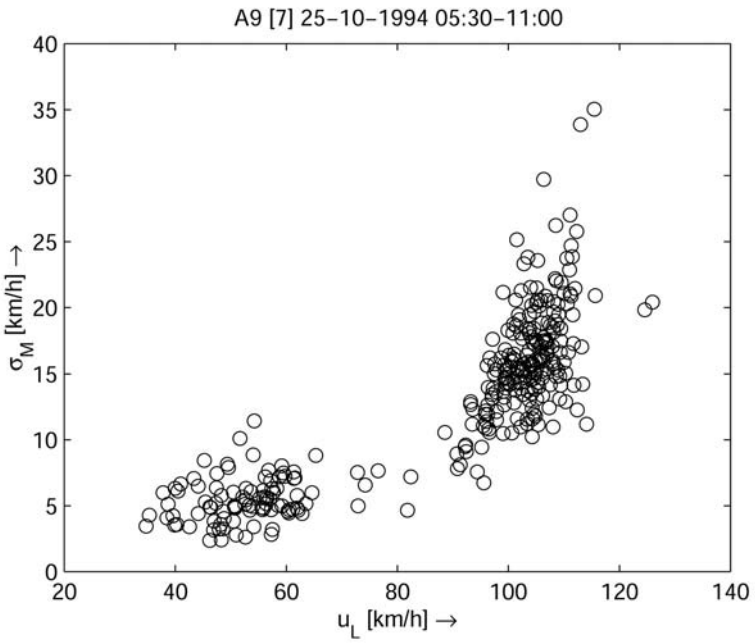


Figure 3.10: Standard deviation as a function of time mean speed.

Estimating speed variance using a time series approach.

The second approach (e.g. (Van Lint & Van der Zijpp 2003*a*), (Lindveld et al. 2000)) is to approximate the instantaneous variance by the local variance, i.e. $\sigma_M^2 \approx \sigma_L^2$. Substituting in eqn (3.24) and solving for u_M gives

$$u_M = \frac{1}{2}u_L + \sqrt{\frac{1}{4}u_L^2 - \sigma_L^2},\ \sigma_L \leqslant \frac{1}{2}u_L \tag{3.34}$$

We then estimate σ_L^2 by looking at the variance of subsequent time mean speeds. Let

the quantities

$$\begin{aligned} U_L &= \frac{1}{P}\sum_{p=1}^{P} u_{p,L} \\ \Theta_L^2 &= \frac{1}{P-1}\sum_{p=1}^{P}\left(u_{p,L}-U_L\right)^2 \end{aligned}$$

denote the mean and variance of the time series of mean speeds $\{u_{p,L}\}_{p=1}^{P}$. Given all individual observations are statistically independent and drawn from the same distribution the central limit theorem states that

$$\Theta_L^2 \approx \frac{\sigma_L^2}{N} \Rightarrow \sigma_L^2 \approx N\Theta_L^2$$

in which each period sample is of equal size N and σ_L^2 denotes the population variance. Note that for relatively large samples (e.g. >20) N could also be taken as the mean sample size. The problem to overcome is that time series of both individual and mean speeds show strong autocorrelation patterns through time, especially in congested conditions and in transient phases between free flowing and congested conditions. This implicates that neither u_L nor σ_L^2 are stationary and thus the central limit theorem does not hold. To illustrate this we calculate the *autocorrelation coefficient* of the time series of mean speeds with (e.g. (Box & Jenkins 1976))

$$\theta(k) = \frac{\vartheta_L^2(k)}{\Theta_L^2} \tag{3.35}$$

where k denotes the time lag for which we calculate the autocorrelation function, and

$$\vartheta_L^2(k) = \frac{1}{P-k}\sum_{p=1}^{P-k}\left(u_{p,L}-U_L\right)\left(u_{p,L}-U_L\right) \tag{3.36}$$

denotes the *auto covariance* for the time series of mean speeds. When we compute autocorrelation coefficients for various time lags. $k \in \{0, 1, 2, ...\}$ we obtain a so-called autocorrelation plot, which indicates the time dependence of the signal under consideration. As Figs. 3.11 (top) and 3.11 (bottom) show, in subsequent speed measurements, autocorrelation patterns differ strongly for free and congested traffic conditions respectively. In the former we see only weak autocorrelation values indicating low time dependence of subsequent values, while for congested conditions a strong autocorrelation is visible for time lags up to 8 minutes. Additionally, the autocorrelation sequence in fig. 3.11 (bottom) shows a cyclic pattern of about 25 minutes, which is probably due to the propagation of (stop-and-go) shockwaves over this particular detector location. Strong time dependence in congested conditions is not surprising: we already saw that variance during congested conditions is lower than during free flow conditions. Statistically, this translates to an increase in the auto covariance.

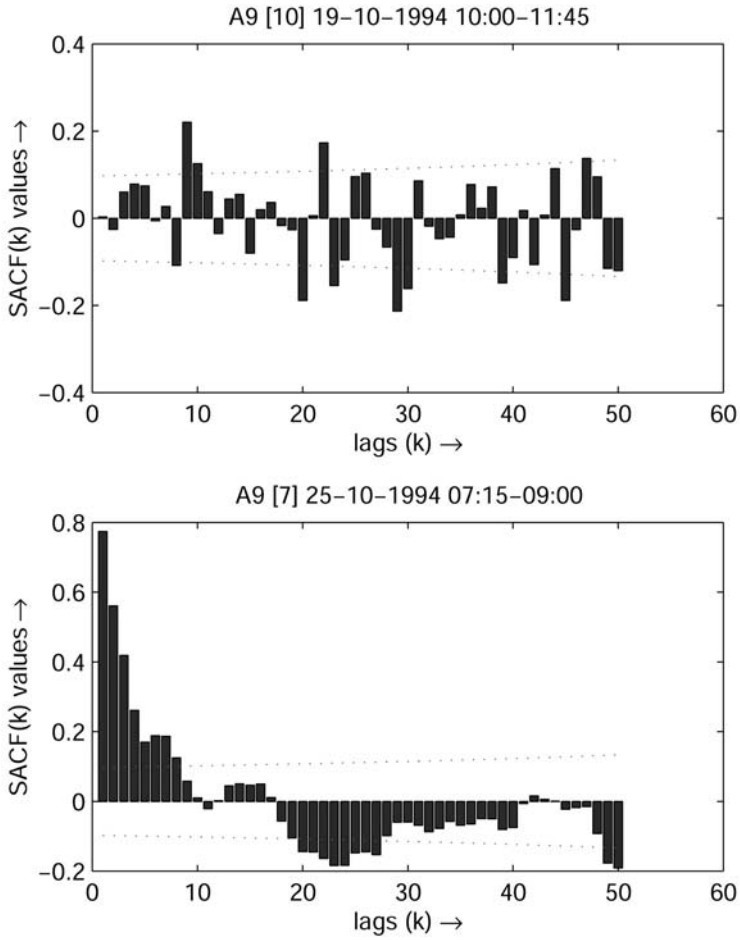


Figure 3.11: Sample Autocorrelation Function for time mean speeds in free (top) and congested (bottom) conditions. The dotted lines give a rough approximation of the significance ($1/\sqrt{P-k}$, where P denotes the number of observations in the time series). The data was recorded at km 39 on the A9 highway from Amsterdam to Muiden on Friday 25 oct 1994.

A common approach to get rid of time dependence (non-stationarity) with respect to the mean is differencing (Pancratz 1991). Let $\widetilde{u}_{p,L} = u_{p,L} - u_{p-1,L}$, and $P+1,\ P \in \{2, 4, 6, ...\}$ denote the size of a time window around p, then analogously to above we can write

$$\widetilde{U}_L = \frac{1}{P+1} \sum_{n=p-P/2}^{p+P/2} \widetilde{u}_{n,L} \tag{3.37}$$

$$\widetilde{\Theta}_L^2 = \frac{1}{P} \sum_{n=p-P/2}^{p+P/2} \left(\widetilde{u}_{n,L} - \widetilde{U}_L\right)^2 \tag{3.38}$$

If $\widetilde{U}_L$ is zero mean normally distributed the procedure effectively removes the non-stationarity with respect to the mean. Fig 3.12 gives strong evidence this is indeed

the case. On a large data set of differenced mean speeds (almost 13,500 records) the calculated differences are very close to zero mean normally distributed.

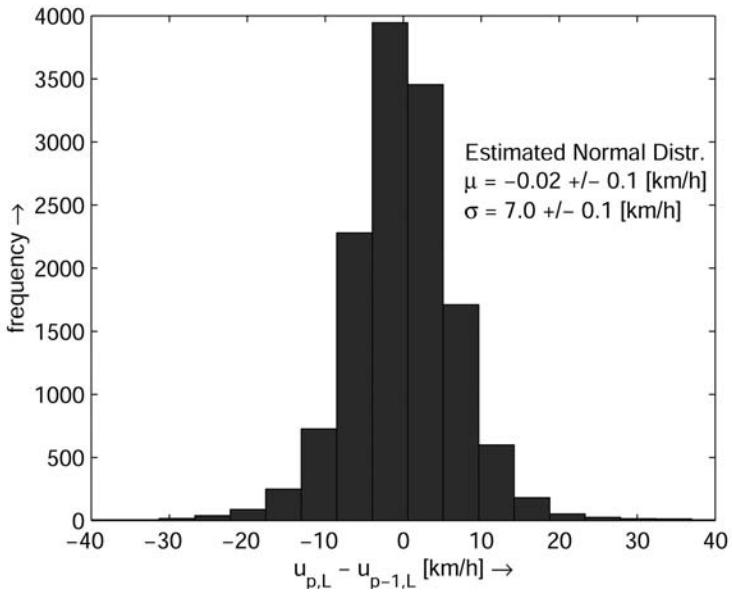


Figure 3.12: Histogram of differenced speeds measured on 12 weekday morning peaks in October 1994 on 4 different detector locations along the A9 highway from Amsterdam to Muiden.

In the limiting case that all observations are independent (which is not the case!) it holds that $\widetilde{\Theta}_L^2 = 2\Theta_L^2$ and thus for all traffic conditions

$$\sigma_L^2 \leq \frac{N}{2}\widetilde{\Theta}_L^2 \tag{3.39}$$

As traffic becomes more congested auto-correlation becomes more significant (see Fig 3.11-bottom) and speed variance becomes smaller. We seek a procedure which lets $\sigma_L^2 \approx \frac{N}{2}\widetilde{\Theta}_L^2$ in free conditions and $\sigma_L^2 << \frac{N}{2}\widetilde{\Theta}_L^2$ in congested periods. To this end we introduce the so-called "local density", which is the time mean speed (in m/s) divided by the traffic flow (veh/s) *per lane* measured in an observation period and equals

$$\rho_{p,L} = \frac{Q_p}{u_{p,L}} = \frac{N_p/\left(I \cdot T_p\right)}{u_{p,L}} \left[veh/m/\ell\right] \tag{3.40}$$

in which I denotes the number of lanes at the specific detector. In various macroscopic traffic flow models (e.g. (Philips 1979)) variance is conceived as a decreasing function of traffic density. For example in (Philips 1979) speed variance is given by

$$\sigma_{p,M}^2 \approx \Theta^0(1 - \frac{\rho_{p,L}}{\rho_{\max}}) \tag{3.41}$$

where Θ^0 (speed variance for $\rho \downarrow 0$) and $\rho_{\max}$ (jam density) are parameters to be estimated from data. A rough approximation then is to consider variance as a decreasing

function of the "local density". Since we found earlier that variance is approximately constant in congested conditions and steeply increasing in free flow conditions we propose the following procedure to estimate within period speed variance

$$\widetilde{\sigma}^2_{p,L} \approx \alpha_p \widetilde{\Theta}^2_L + (1-\alpha_p)\frac{N}{2}\widetilde{\Theta}^2_L \tag{3.42a}$$

$$\alpha_p = \min\left[\frac{\rho_L}{\rho_L^{crit}}, 1\right] \tag{3.42b}$$

Note this procedure contains two parameters. The first (implicit) one is the length $P+1$ of the differenced time series of speeds. Since first order differencing practically removes all non-stationarity with respect to the mean (see fig. 3.12), the choice of P should (theoretically) not influence the results. Nonetheless, P should be chosen not too small to avoid instability and not too large for practical (implementation) reasons. The second parameter, ρ_L^{crit}, denotes the so-called *critical local density* which reflects the (mean) local density per lane at capacity flow (per lane). In macroscopic traffic flow models critical density (on a section) discriminates between free-flowing and congested traffic conditions. Here *critical local density* serves the same purpose. Since density is per definition a variable defined over space, *local* density is likely to be unstable (e.g. due to passing stop-and-go waves and platooning for example), we finally apply an exponential filter (and inherently introduce a third parameter γ) to make the estimator more robust to idiosyncrasies in the local data

$$\widehat{\sigma}^2_{p,M} = (1-\gamma)\cdot\widehat{\sigma}^2_{p-1,L} + \gamma\cdot\widetilde{\sigma}^2_{p,L} \tag{3.43}$$

Typically γ should be small enough to avoid instability but large enough for the algorithm not to lag behind too much. Using a Nelder Mead optimization procedure we find the following values that minimize $\left(\widehat{\sigma}^2_{p,L} - \sigma^2_{p,M}\right)^2, \forall p$ on the same data set used for fig. 3.12.

$$\begin{aligned} \rho_L^{crit} &= 0.02\ veh/m/\ell \\ P &= 30 \\ \gamma &= 0.24 \end{aligned}$$

Note that the value for critical density complies with critical density values reported in literature (e.g. (Hoogendoorn 1999)). As a more qualitative indication, fig. 3.13 shows the performance of the variance estimator compared to "true" speed variance for three detectors along the A9 on the morning peak of October 25 1994. The performance of the procedure on detectors 7 and 10 (fig. 3.13, middle and bottom graph) is satisfactory, while on detector 5 variance is structurally underestimated. In this case the cause is straightforward; on detector 5 traffic was dense but no traffic breakdown (congestion) actually occurred.

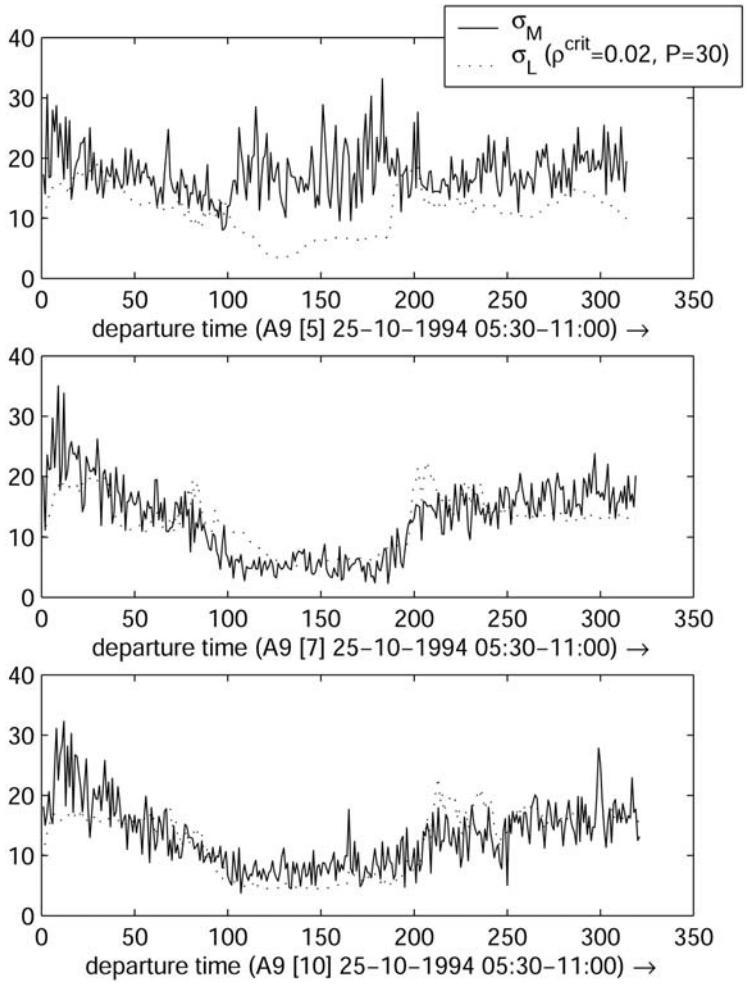


Figure 3.13: Performance of speed variance estimator on three different detector locations during the morning peak of October 25 1994 on the A9 highway from Amsterdam to Muiden.

Algorithm 1 *Time series speed variance estimator*

1. *Set $p = p_0$ and make an initial estimate $\widehat{\sigma}_L^2(p)$. If traffic is free-flowing $\widehat{\sigma}_L = 20$ (km/h) or conversely in congested traffic $\widehat{\sigma}_L = 5$ (km/h) are reasonable values.*

2. *Calculate $\widetilde{\Theta}_L^2$ (eqn 3.38) from a time series window of size $P+1$ around the current time period p.*

3. *Calculate local density $\rho_{p,L}$ (eqn 3.40) and α_p (eqn 3.42b)*

4. *Estimate variance $\widehat{\sigma}_{p,M}^2$ with eqns (3.42a) and (3.43)*

5. *set $p = p+1$ and go to step 2*

Results of the speed variance estimators

Recall the purpose of the two estimators of within sample variance presented above is to provide a good estimate of the space mean speed, since that is the quantity needed to estimate travel time (eqns 3.19 and 3.20). Fig 3.14 shows the performance of each of the two variance estimators in terms of travel time on detector 4 (which was not used for calibration of either method) on a heavily congested morning peak (October 21 1994) along the A9. Travel times are estimated on a 1, 000 meter stretch surrounding this detector on which we assume homogeneous and stationary conditions in each measurement period of 1 minute. As a baseline comparison fig. 3.14 also shows the travel time based on time mean speeds (which equals zero variance). Note that for readability the data in these graphs have been filtered.

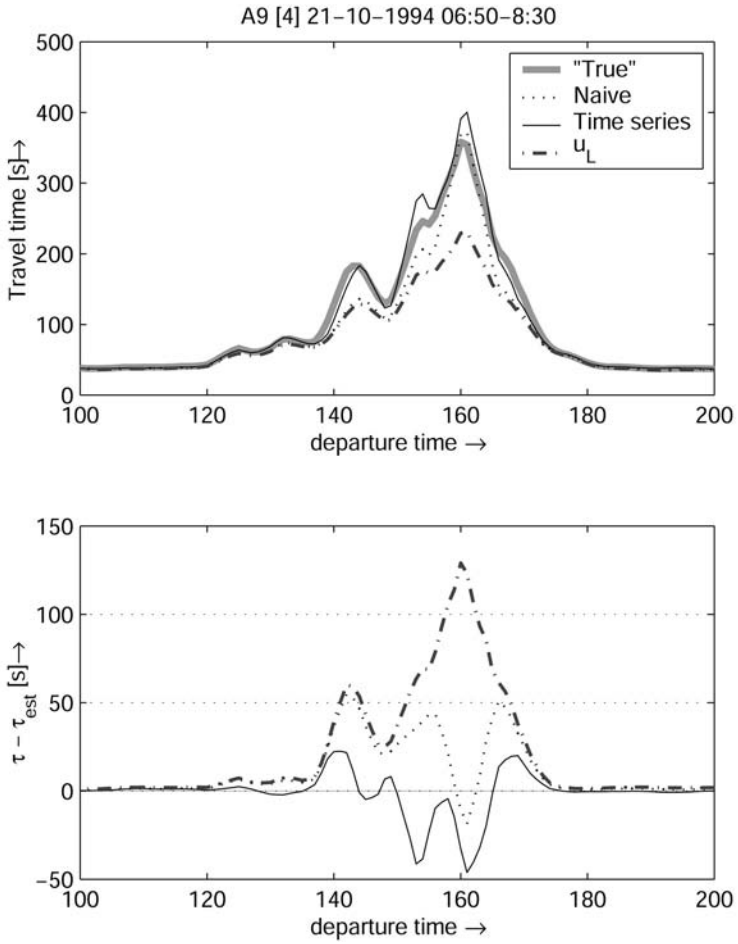


Figure 3.14: Performance of speed variance estimators in terms of travel time estimated on a 1,000 meter stretch surrounding detector 4 on the A9 Highway from Amsterdam to Muiden on October 21 1994 during the morning peak.

Fig. 3.14 confirms that travel times based on local time mean speeds structurally underestimate "real" travel time. Nonetheless, the travel times produced with speeds

corrected for bias still produce substantial errors. For a more qualitative comparison Table 3.2 shows the performance on the estimates of speed and travel time based on time mean speeds, the naive variance estimator and the time series variance estimator. As indicators we use bias (equals mean error) and root residual error (the residual error after removing bias), see appendix A for details. The data used consists of 12 weekday morning peaks (of 5,5 hours) on the same detector as above. The variance estimators significantly improve bias in terms of both speed and travel time, with the time series estimator performing best as it practically removes all bias. Nonetheless, both methods do not improve (even slightly increase) the residual error after bias removal.

Table 3.2: Bias (Mean Error) and Root Residual Error (RRE) - see appendx A - on estimated speed and travel times based on the arithmetic time mean speed, the naive variance estimator and the variance estimator based on a differenced time series of mean speeds. The source data was recorded at 1 locations on the two-lane A9 highway from Alkmaar to Amstelveen during 12 weekdays of October 1994 (1125 observations).

	Speeds		Travel times	
	Bias [km/h]	RRE [km/h]	Bias [s]	RRE [s]
Time mean speed	3.8	2.3	-14.5	43.4
Naive estimator	1.6	2.7	2.0	45.0
Time series	0.1	2.8	-0.3	43.8

3.3.4 Some critical notes on bias correction algorithm

The bias correction algorithm presented in the previous sections, performed reasonably well on a typical location on a two-lane highway during a typical morning peak, practically removing all bias caused by arithmetic time averaging of speeds. The method, however, does not reduce the residual error (variance) associated with arithmetic time mean speeds. Moreover, these results do not imply that the speed based travel time estimation method reproduces actual time travel times accurately, since we do not have actually measured travel times. No matter the bias correction algorithm, the underlying assumptions are homogeneous and stationary conditions within measurement periods on locations between detectors. This may well imply that due to location-specific circumstances[6] which for example seriously affect the validity of homogeneity and stationarity, speed based travel time estimation techniques may still induce serious errors on particular locations with respect to real travel times, even when calculated with harmonic mean speeds. In chapter 8 we revisit the bias correction problem and compare travel time estimates against real travel times.

[6]Examples of these circumstances include designated lanes for particular classes of vehicles, the vicinity of on- and off ramps or weaving sections, and the application of traffic control (speed calming, lane closure or shoulderlane usage).

In the remainder of this chapter two speed based travel time estimation algorithms are introduced of which the first has been widely used in practice on inductive loop equipped freeway routes. Both classify as so-called trajectory algorithms, which let imaginary vehicles traverse through a database of measured mean speeds on detector locations along that route. The first assumes piece wise constant speeds, while the second (improved one) considers piece wise linear speeds.

3.4 Estimating Travel Time on Routes: The PLSB Trajectory Method

In this section we present the so-called *piece-wise linear trajectory algorithm* (fig. 3.17 on page 74), to estimate travel times on routes of adjacent freeway sections (Van Lint & Van der Zijpp 2003*b*). The key to this method is the subdivision of space and time into regions $\{k, p\}$ in which it is assumed vehicle speeds are either constant or some function of the speeds measured during period p on section k. For simplicity, assume that each section k is enclosed by up- and downstream detectors d and $d+1$ which measure local harmonic mean speeds. In the trajectory algorithm, imaginary vehicles traverse this grid through space and time. Obviously, with these imaginary trajectories travel times can be deduced directly (compare fig. 3.1).

First it is explained in detail how at the level of a single region $\{k, p\}$ section level travel times can be calculated. The crux is how to generalize local measurements (from up- and downstream detectors) over an entire section. The classical trajectory method assumes constant speeds on a section surrounding a detector. We propose an improvement, in which we assume speed as a convex combination of up- and downstream speeds. Subsequently, these section-level travel times can be used to constitute path (route) level travel times. Finally, we show (through simulated data) that the improvement indeed leads to more accurate travel time estimates.

3.4.1 Section level travel times based on piece-wise constant speeds

In the classical and widely used trajectory method (see e.g. (Lindveld et al. 2000), (Kraan et al. 1999), and (Van der Zijpp & Lindveld 1999)) speed on a section is conceived piece-wise constant, that is, over the upstream half of a section vehicles are assumed to travel with the speed $\widehat{u}^{d}_{p,M}$ measured on the upstream detector d, and on the downstream half with speed $\widehat{u}^{d+1}_{p,M}$ measured at the downstream detector $d+1$ (fig. 3.15). Note that here it is assumed harmonic mean speed or corrected time mean speeds are available!

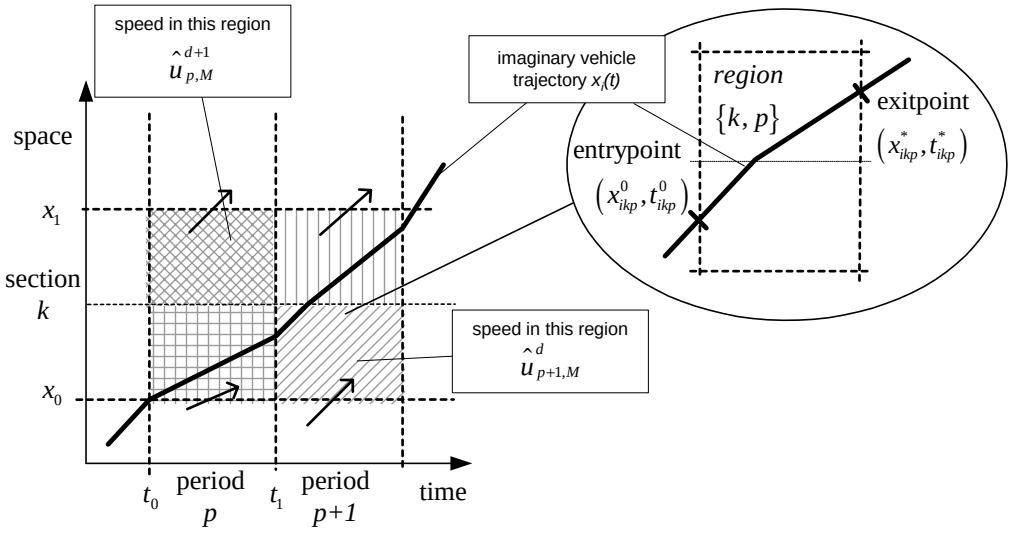


Figure 3.15: Example of a vehicle trajectory $x\,(t)$ through space-time cells $\{k,\, p\}$ and $\{k,\, p+1\}$. In the figure above speeds are conceived piecewise-constant over each cell $\{k,\, p\}$. To the right the entry and exit point of a trajectory through a cell $\{k,\, p\}$ are indicated.

Recall time periods are denoted with $p = [t_0, t_1]$ and sections by $k = [x_0, x_1]$. Consequently, a region $\{k,\, p\}$ in space time is enclosed by the tuples (x_0, t_0) and (x_1, t_1). Furthermore let assume that an imaginary vehicle enters region $\{k,\, p\}$ at $\left(x^0_{ikp}, t^0_{ikp}\right)$. The clue of the trajectory method is to calculate where in region $\{k,\, p\}$ the vehicle will exit. This exit point is denoted by $\left(x^*_{ikp}, t^*_{ikp}\right)$. At section level, the calculation of this exit point is straightforward when recalling equations (3.4) and (3.1). The procedure takes two steps. First the vehicle's trajectory for the upstream half is calculated,

$$\widehat{u}^{d}_{p,M}(t_1 - t^0_{ikp}) + x^0_{ikp} > \frac{1}{2}(x_1 - x_0) \tag{3.44}$$

$$\{x^{1/2}_{ikp}, t^{1/2}_{ikp}\} = \begin{cases} \left\{x_1,\, \frac{(x_1 - x^0_{ikp})}{\widehat{u}^{d}_{p,M}} + t^0_{ikp}\right\} & (3.44)\ holds \\ \left\{\widehat{u}^{d}_{p,M}(t_1 - t^0_{ikp}) + x^0_{ikp},\, t_1\right\} & otherwise \end{cases} \tag{3.45}$$

and next, if the vehicle is still traversing region $\{k,\, p\}$ the vehicle's trajectory for the downstream half is calculated

$$\widehat{u}^{d+1}_{p,M}(t_1 - t^0_{ikp}) + x^0_{ikp} > \frac{1}{2}(x_1 - x_0) \tag{3.46}$$

$$\{x^*_{ikp}, t^*_{ikp}\} = \begin{cases} \left\{x_1,\, \frac{(x_1 - x^{1/2}_{ikp})}{\widehat{u}^{d+1}_{p,M}} + t^{1/2}_{ikp}\right\} & (3.46)\ holds \\ \left\{\widehat{u}^{d+1}_{p,M}(t_1 - t^{1/2}_{ikp}) + x^{1/2}_{ikp},\, t_1\right\} & otherwise \end{cases} \tag{3.47}$$

These equations allow one to compute at what time the end of the section k is reached or which position is reached at the end of period p, whatever comes first. As such, they form the basic building block of what we will refer to as the **piece-wise constant speed based (PCSB) trajectory method**.

3.4.2 Section level travel times based on linear speeds

When the section level travel time estimators presented in the previous section are used at path-level they result in piece-wise linear trajectories. Vehicles are thought to instantaneously change their driving speed once entered a new region of constant speed. In reality, this transition will occur in a more smoothed fashion: vehicles are likely to anticipate to slower or faster speed regimes downstream and gradually adapt their speeds to it (fig. 3.16).

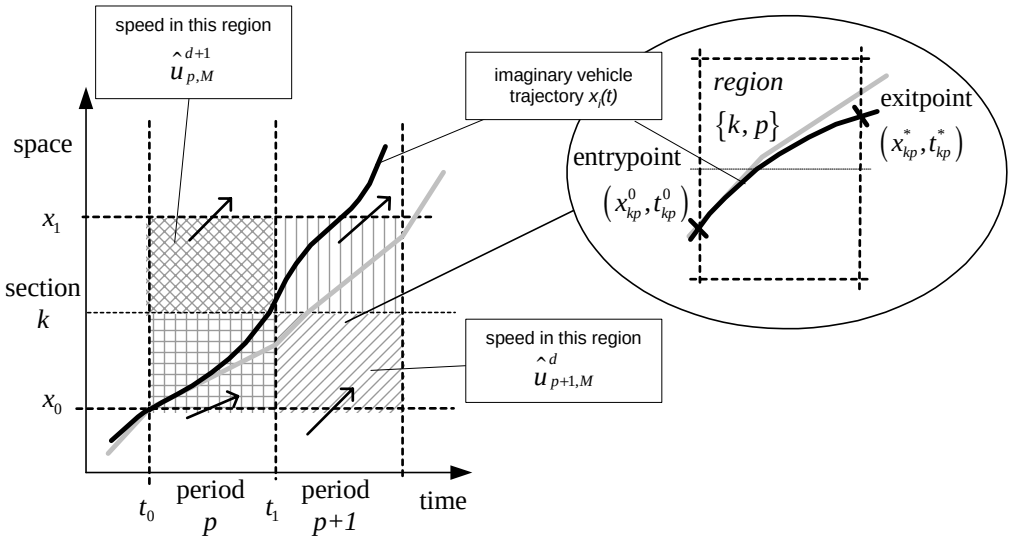


Figure 3.16: Example of a vehicle trajectory $x(t)$ through space-time cells $\{k, p\}$ and $\{k, p+1\}$. In the figure above speeds are conceived piecewise-linear over each cell $\{k, p\}$. To the right the entry and exit point of a smoth trajectory through a cell $\{k, p\}$ are indicated. For comparison, in grey the trajectory based on piecewise constant speeds are shown.

We propose to relax the notion of constant speeds, and consider the speed $v_i(t)$ of a vehicle i traversing a section between detector locations x_0 and x_1 during period p as a function of the distance of that vehicle to these up- and downstream detectors (see fig. 3.16):

$$v_i(t) = \widehat{u}^d_{p,M} + \frac{x_i(t) - x_0}{x_1 - x_0}\left(\widehat{u}^{d+1}_{p,M} - \widehat{u}^d_{p,M}\right) \tag{3.48}$$

Again local harmonic mean speeds are assumed at the up- and downstream detectors. Although traffic conditions over the entire region $\{k, p\}$ are no longer homogeneous,

we can circumvent more complex expressions for expected travel time by considering a small region $[x, x+dx] \in k$ and proving that in the limit that $dx \to 0$ for each such regions homogeneity holds and thus expression 3.23 is justified. The experienced travel time over the entire section then equals the linear sum of all travel time contributions on consecutive small regions.

For simplicity sake let $u^- = \widehat{u}^d_{p,M}$ and $u^+ = \widehat{u}^{d+1}_{p,M}$. The space mean speed on a small region $[x, x+dx]$ at $t \in p$ then equals

$$\begin{aligned}
\langle v \rangle_M &= \frac{u^- + \frac{x-x_0}{x_1-x_0}(u^+ - u^-) + u^- + \frac{x+dx-x_0}{x_1-x_0}(u^+ - u^-)}{2} \\
&= u^- + \frac{1}{2}\left(\frac{x-x_0}{x_1-x_0} + \frac{x+dx-x_0}{x_1-x_0}\right)(u^+ - u^-) \\
&= u^- + \left(\frac{x-x_0}{x_1-x_0}\right)(u^+ - u^-) + \frac{1}{2}\left(\frac{dx}{x_1-x_0}\right)(u^+ - u^-)
\end{aligned}$$

In the limit that $dx \to 0$ we have

$$\langle v \rangle_M = u^- + \left(\frac{x-x_0}{x_1-x_0}\right)(u^+ - u^-)$$

Since we assume stationarity, the harmonic mean at location x during period p equals

$$\left\langle \frac{1}{v} \right\rangle_L = \frac{1}{u^- + \frac{x-x_0}{x_1-x_0}(u^+ - u^-)}$$

and thus, for very small regions $[x, x+dx]$ it holds that

$$\langle v_i(x) \rangle_M = \frac{1}{\left\langle \frac{1}{v_i(x)} \right\rangle_L} = u^- + \frac{x-x_0}{x_1-x_0}(u^+ - u^-)$$

Consequently, since we *do* assume stationarity, the travel time on the entire section k during p can be considered a linear sum of travel times experienced on stationary and homogeneous sections $[x, x+dx] \in k$, which in the the limit that $dx \to 0$ equals the integral

$$t_i(x) = \int_{x^0_{ikp}}^{x} \frac{dx}{v_i(x)} = \int_{x^0_{ikp}}^{x} \frac{dx}{u^- + \frac{x-x_0}{x_1-x_0}(u^+ - u^-)}$$

Taking small steps Δx the integral can be approximated with

$$t_i(x_1) = \sum_{j=\left(x^0_{ikp}-x_0\right)/\Delta x}^{j=x_1/\Delta x} \frac{\Delta x}{u^- + \frac{j\Delta x - x_0}{x_1-x_0}(u^+ - u^-)}$$

Instead of solving this integral numerically, one can observe that eqn (3.48) is an ordinary differential equation, which solution can be derived analytically and yields a closed form expression for a vehicle trajectory:

$$\begin{aligned} x_i(t) &= x^0_{ikp} + \left(\frac{\widehat{u}^d_{p,M}}{A} + x^0_{ikp} - x_0 \right) \left(e^{A(t-t^0_{ikp})} - 1 \right) \\ A &= \frac{\widehat{u}^{d+1}_{p,M} - \widehat{u}^d_{p,M}}{x_1 - x_0}, \ |A| > 0 \end{aligned} \tag{3.49}$$

The term A can be interpreted as the mean acceleration (deceleration) over section k during period p. In the limit A goes to zero eqn (3.49) reduces to

$$x^0_{ikp} + \widehat{u}^d_{p,M} \left(t - t^0_{ikp} \right)$$

In practice this applies when the upstream and downstream observed speeds are nearly equal. Note that in these cases equations (3.47) and (3.45) may be used instead. Analogously to the PCSB trajectory method, It is now straightforward to calculate the point where a vehicle exits region $\{k, p\}$:

$$x^0_{ikp} + \left(\frac{\widehat{u}^d_{p,M}}{A} + x^0_{ikp} - x_0 \right) \cdot \left(e^{A(t_1 - t^0_{ikp})} - 1 \right) > x_1 \tag{3.50}$$

$$\{x^*_{ikp}, t^*_{ikp}\} = \begin{cases} \left\{ x_1, t^0_{ikp} + \frac{1}{A} \ln \left(\frac{\frac{\widehat{u}^d_{p,M}}{A} + x_1 - x_0}{\frac{\widehat{u}^d_{p,M}}{A} + x^0_{ikp} - x_0} \right) \right\} & (3.50)\ holds \\ \left\{ x^0_{ikp} + \left(\frac{\widehat{u}^d_{p,M}}{A} + x^0_{ikp} - x_0 \right) \cdot \left(e^{A(t_1 - t^0_{ikp})} - 1 \right), t_1 \right\} & otherwise \end{cases} \tag{3.51}$$

again with $|A| > 0$. These equations allow one to compute at what time the end of the section k is reached or which position is reached at the end of period p, whatever comes first. As such, they form the basic building block of what we will refer to as the **piece-wise linear speed based (PLSB) trajectory method**.

3.4.3 Route-level travel times

Now practical techniques are established to calculate the exit time and location of a vehicle traversing a particular region $\{k, p\}$ in space-time, it is straightforward to extend these to entire routes consisting of adjacent sections $k = \{1, 2, .., K\}$, each of which are enclosed by up- and downstream detectors measuring (local mean speeds) over periods $p = \{1, 2, .., P\}$. Fig. 3.17 schematically shows how such an algorithm can be implemented. In (Bovy & Thijs 2000),(Thijs et al. 1999) and (Lindveld &

Thijs 1999) this trajectory algorithm is applied in conjunction with the PCSB estimator and referred to as the *dynamic network level travel time estimator*. The figure shows that both PCSB and PLSB algorithms can be plugged in this framework easily. The only difference between the methods is in the calculation of the exit location and time (grey box in fig. 3.17) of a region $\{k, p\}$. In fact any section level speed or flow based travel time estimation method that can calculate the exit time and location of a vehicle given its entry time and location can be plugged in the trajectory algorithm, including flow based methods

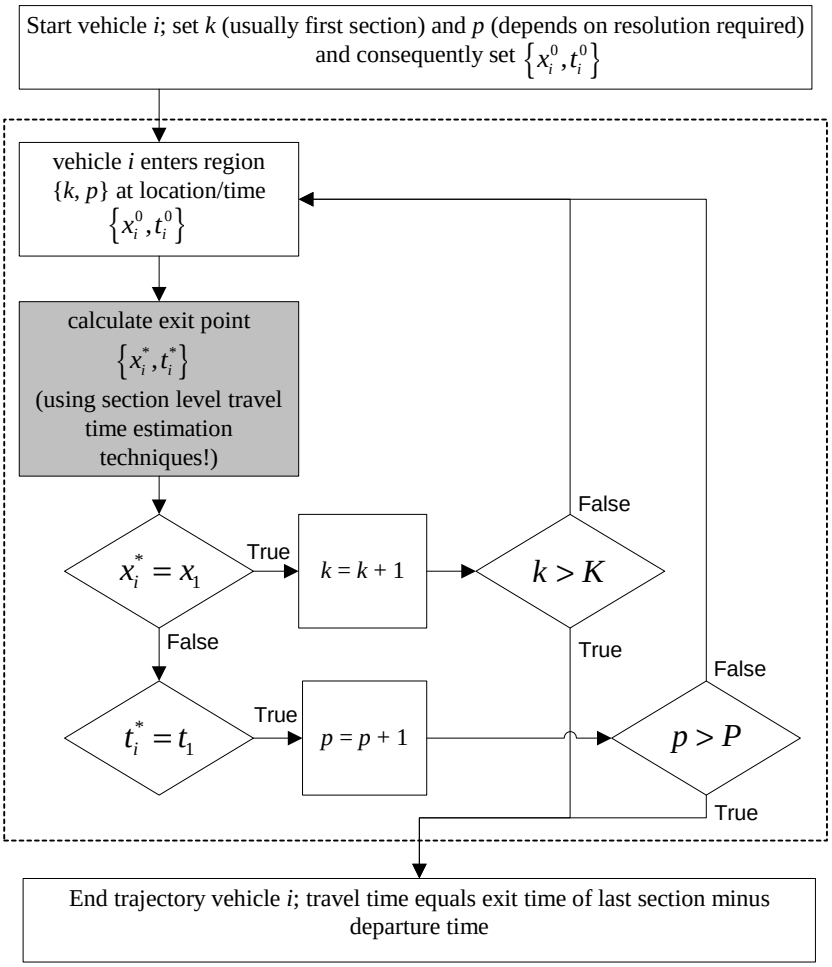


Figure 3.17: Schematic representation of a trajectory method; different section level travel time estimators can be plugged in the framework easily (grey box in the scheme).

In words, the algorithm lets imaginary vehicles i traverse the route starting at time instant t_i^0. Each time a vehicle i enters a section $\{k, p\}$, its exit location and time $\left(x_{ikp}^*, t_{ikp}^*\right)$ is calculated until the vehicle arrives at the final section comprising the route. The exit time t_i^* of that section minus its departure time then represents the mean travel time over the entire route of vehicles departing at t_i^0.

3.4.4 Numerical evaluation of PCSB and PLSB trajectory methods

This section presents the results of a numerical comparison between the trajectory method based on piecewise constant speeds and the newly proposed trajectory method based on piecewise linear speeds. Two types of experiments have been performed. In the first experiment we illustrate the differences between the method by assuming stationary traffic conditions over a long time period. The second experiment consists of a series of tests that are based on simulated data. In chapter 8 we will analyze the performance of the PLSB trajectory method when confronted with real measurements from inductive loops.

Stationary conditions: conceptual difference between the PCSB and PLSB method.

In Fig 3.18 an example is given of the application of both methods on a hypothetical route of 3000 metres. The vertical dotted lines indicate locations of detectors. It is assumed that the speeds these detectors measure are stationary for an indefinite time period. From fig. 3.18 it is clear that the PLSB method structurally produces lower travel times than the PCSB method, in other words, in the PLSB method vehicles spend less time in low speed areas than in its PCSB counterpart.

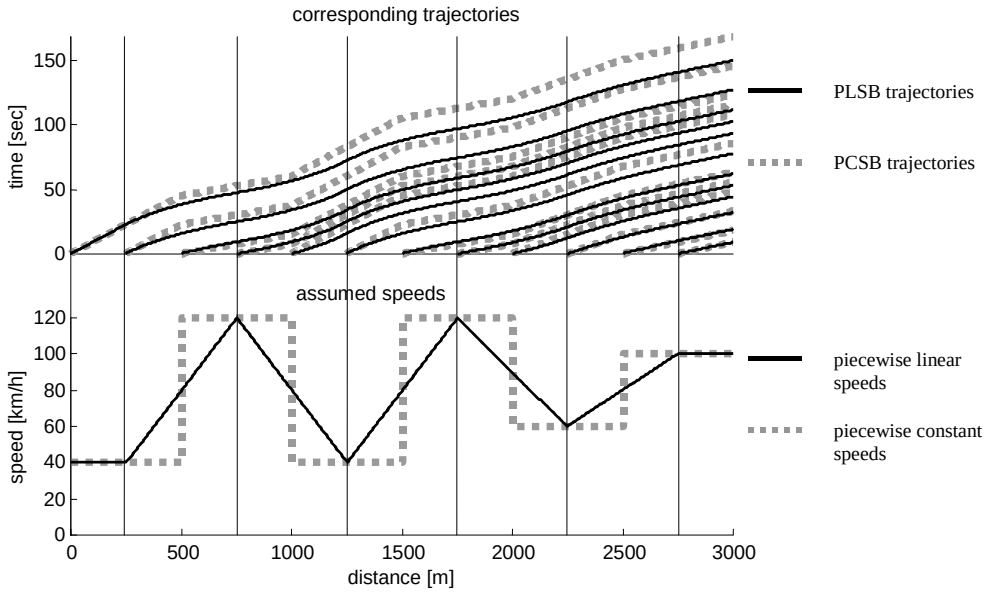


Figure 3.18: The conceptual difference between the classic trajectory method based on the PCSB method and the PLSB method in the case of stationary traffic conditions. Note that in this graph time is plotted as a function of space!

Dynamic conditions (simulated data)

The PLSB method has been evaluated in a series of experiments based on simulated data. Details on the data sets used are given in chapter 5, section 5.5, where the same data sets are used to derive and calibrate travel time prediction models. Below the PLSB and PCSB performance are calculated on five separate data sets of traffic pattern 1 (normal congestion), totalling in approximately 1500 records. These data have been generated using the microscopic traffic simulation model FOSIM (Vermijs & Schuurman 1994). The network has been specified matching the southbound stretch of the A13 motorway between Delft and Rotterdam (the Netherlands). For each simulation run, the traffic demand patterns and the random seed generator were different, resulting in different but realistic travel time patterns.

Fig. 3.19 (top) shows a typical result. The graphs show the estimated travel times using the PCSB trajectory method (dotted), the new PLSB trajectory method (solid) and the reference values obtained from the simulation (grey). The bottom graph in fig. 3.19 shows the difference between the reference value and the PCSB and PLSB trajectory methods respectively. As in the stationary case, the PCSB trajectory method produces higher travel time estimates than the PLSB method, especially during congested conditions.

Table 3.3 shows the MRE, SRE and RMSEP performance (see appendix A) for both methods on all five data sets during congested conditions (travel times $>$ 450 seconds). During free-flow conditions both methods (as expected) perform equally well. On all performance indicators the new PLSB trajectory method performs (slightly) better than the classic PCSB trajectory method.

Table 3.3: Performance indicators of PCSB and PLSB trajectory methods on five synthetic datasets.

	PCSB method	PLSB method
MRE (%)	3.9	-2.5
SRE (%)	7.5	6.6
RMSEP (%)	8.1	6.3

The bottom line is that it depends on the actual traffic conditions in between detectors whether either the PCSB or PLSB method will produce more accurate results. This is a product of the assumption of homogeneity and stationarity within space time regions $\{k, p\}$ on which both methods are based. For example, at the head (or tail) of queues, piece-wise linear speed may prove a better assumption than piece wise constant speed, due to vehicles accelerating (or deceleration) from congested to free conditions (or vice versa), while inside a traffic jam the assumption of constant speed may be just as (in)accurate. In the remainder of this dissertation thesis, however, we will always use the PLSB method.

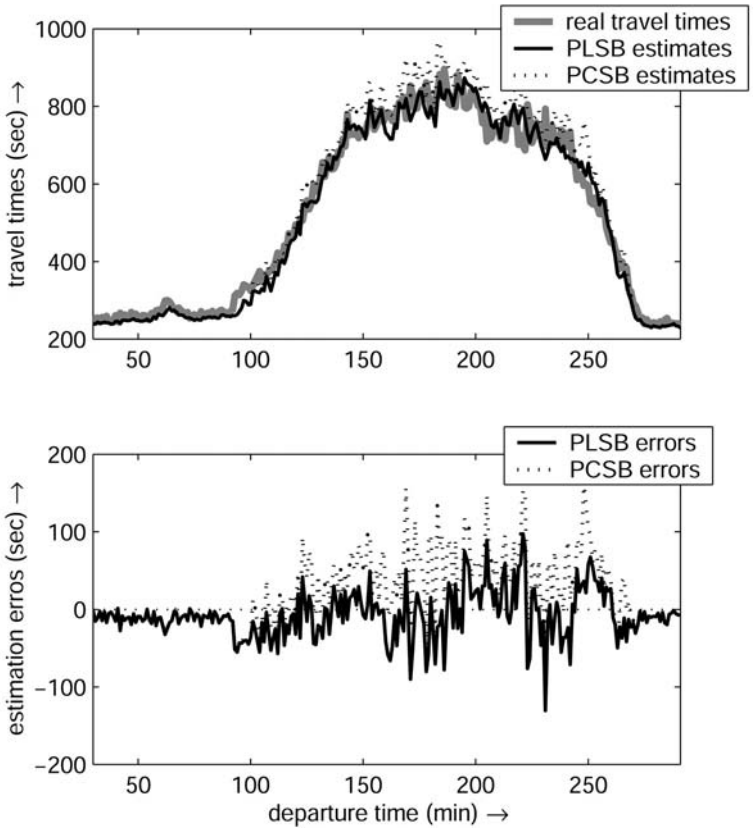


Figure 3.19: Comparison of methods on synthetic data. The top graph shows the estimated travel times as a function of departure time. The PCSB (dotted line) shows more bias and erratic behavior then the PLSB (solid line). The bottom graph plots the estimation errors of both methods.

3.5 Summary

In the previous chapter we defined individual and mean travel times and outlined and defined the travel time estimation and prediction problem qualitatively. In this chapter we took one step further and explored the travel time estimation problem mathematically. First we showed how individual and mean travel time relate to other traffic quantities, that is vehicular speed, traffic flow (number of vehicles that pass a location per unit time) and traffic density (number of vehicles per unit space). Given that we assume stationary and homogeneous traffic conditions on some road section k during a time period p the mean travel time on that section during that period then equals the reciprocal of the *space mean speed* times the length of the section.

Next we showed travel times are not often measured directly nor are space mean speeds with which travel times can be estimated (offline). For example, the MONICA inductive loop system deployed on Dutch freeways gathers local arithmetic time mean speeds, which structurally overestimate space mean speed and hence underestimate travel times. There is, however, a well known analytical relationship between space

and time mean speeds from which one can derive this bias which is proportional to speed variance. We derived two methods to approximate speed variance with the quantities we do measure (time mean speeds and traffic flow at detectors) and used these methods to correct for the bias in mean speeds and travel times.

Finally, we introduced a new algorithm, the so-called piece-wise linear speed based (PLSB) trajectory algorithm, with which travel times can be accurately measured on routes equipped with detectors measuring local speeds. In ideal situations (that is based on simulated data), the PLSB travel time estimates produce a small bias (-2%) and variance in the order of 6% of the simulated travel times. We will return to the PLSB travel time estimation method in chapter 8, where we will exploit it in a real-time travel time prediction framework. The next chapter will introduce the problem of *predicting* travel times on freeways, which is a far more complicated problem, in which we must explicitly address the dynamics of traffic flow operations.

Chapter 4

The Short Term Freeway Travel Time Prediction Problem: State-of-the-Art

4.1 Introduction

In the previous chapter the relationship between travel times and other traffic quantities were established and methods to estimate travel times from these traffic quantities (speeds, flows) were explored. As noted before, travel time estimation does not necessarily concern with the dynamics of traffic flow processes, since it reflects the translation of "known" traffic conditions (speeds and flows) into travel times.

This, however, does not hold for travel time prediction. This chapter illustrates that predicting travel time on a route r is in fact equivalent to predicting traffic conditions on that same route, which is, given the highly non-linear spatiotemporal processes that generate these conditions, very complex indeed. First, in section 4.2 a taxonomy of travel time prediction models is presented. According to that taxonomy, this thesis focusses on short term data driven freeway travel time prediction models. Then, in section 4.3 we analyze the complexity of short term freeway travel time prediction problem and discuss the State-of-the-Art in that field, which can roughly be divided in three strands, namely model based, instantaneous and data driven approaches. This chapter closes with a critical discussion and a brief summary. Based on the findings here, the next chapter subsequently develops a data driven method (based on recurrent neural networks) for short term travel time prediction.

4.2 Taxonomy of Travel Time Prediction Models

In order to better classify and understand the different approaches to travel time prediction below the main distinguishing factors between travel time prediction models found in practice and literature are presented.

Prediction Horizon *Short and long term prediction models* (for definitions see section 2.2). Arguably, this is the most important distinguishing factor, since different prediction horizons yield (require) very different modelling approaches, methodologies, put different constraints on input factors and have very different application types. In general, the longer the prediction horizon, the more models rely on either statistical (e.g. through ARIMA or regression) or theoretical assumptions (e.g. Wardrop equilibrium or Dynamic User Optimum) regarding future traffic conditions. In this thesis we will exclusively deal with short term travel time prediction (online travel time prediction is in fact a special case of short term prediction, see section 2.2).

Modeling Approach *Data driven, model-based, instantaneous.* Data driven models regard the traffic processes generating (mean) travel times as black boxes and use statistical relations to infer future travel times from past data (travel times, speeds, flows, but also temporal information and for example weather conditions). Model based approaches make use of traffic flow simulation models (based on traffic flow theory) to predict the traffic conditions on the route of interest. Given the wide body of research on traffic theory in the past decades, this choice seems most appropriate. However, it inherently forces the modeler to predict the traffic conditions at the boundaries of the model used (traffic demand at origins and on ramps, capacity/supply restrictions at destinations and off ramps). The key to instantaneous approaches[1] is that traffic conditions are considered stationary from the departure time period onwards. This assumption then allows the modeler to use data measured at the time of departure of vehicles only and use travel time *estimation techniques* to derive mean travel time (for these vehicles). Thus, by assuming stationarity, the travel time prediction problem is reduced to a travel time estimation problem, which can be solved with for example the techniques discussed in the previous chapter. Note that some models may be considered Hybrid models that combine modeling approaches.

Methodology *Direct or indirect travel time prediction* (for definitions see section 2.2). Since predicting travel times is equivalent to predicting traffic conditions (see subsequent sections), indirect methods predict traffic conditions and subsequently infer future travel times from these conditions. Most model based methods can be classified as indirect travel time predictors, while time series (e.g. ARIMA, neural networks) models often predict travel times directly.

Spatial Scope *Network, route, section.* Spatial scope is strongly related to modeling approach. A (network level) model based approach yields network wide travel time predictions. Most data driven approaches operate on link or path level. In this thesis, the focus here is on path or route based travel time prediction.

Road Type *Freeways, urban (controlled) roads.* The principal difference between freeway traffic and urban traffic is that the first is uninterrupted by (controlled

[1] see chapter 2 for definition of instantaneous travel time

or uncontrolled) intersections. This strongly influences travel time. Tentatively, we hypothesize that travel time variance (both short and long term) on a freeway corridor will be much larger than on a similarly sized urban road with a number of controlled intersections. As an example, within the Regiolab Delft project (Van Zuylen & Muller 2002) travel times on a four kilometer provincial road with a maximum speed of 70 and 100 km/h and four controlled intersections range from 5 - 10 minutes (peak versus off peak) while a similar length road stretch on the A13 freeway produces mean travel times between 3 and 15 minutes (peak versus off peak). In chapter 9 we devote a section to the fundamental differences between freeway and non-freeway travel time prediction. As already stated the focus here is on freeway travel time prediction.

Application Type *On-trip, pre-trip*. Applicability is strongly related to prediction horizon, spatial scope and road type. On-trip applications of travel time prediction models (e.g. in-car traffic information services, ATIS such as variable message signs (VMSs)), necessarily require predictions relevant at the instant of information provision. Pre-trip applications (e.g. web-based intelligent route planning system) provide both short and long term predictions dependent on the time instant it is consulted by travellers (e.g. a day or 5 minutes before departure). For short term on-trip purposes, predictions at section or route level may suffice, while in longer-term pre-trip planning tools predictions on a network level are required. As noted earlier, this thesis focusses on short term travel time prediction for freeway routes, which can be applied for example in real-time on-trip ATIS systems .

Input Factors *Network effects, temporal effects, population characteristics, ATIS, incidents / accidents, roadworks, ambient factors (e.g. weather), geometry / regulations, ATMS* (see section 2.3). Input factors are closely related to Prediction horizon, Modeling approach and Methodology. For example, a data driven approach would use all factors available (explaining travel time variance) regardless whether physical (mathematical) relationships between those factors and travel time can be formulated, while model based predictors inevitably require (time dependent) traffic data (e.g. dynamic OD matrices) as inputs. Unless a traffic simulation model explicitly incorporates ambient or other factors, these can not be utilized in the model based travel time prediction task.

Input Traffic Data *Inductive loop, radar, infrared, microwave detection systems, video based systems, AVI (automatic vehicle identification), floating car data* (see section 3.3). From an application point of view, this factor governs many of the above mentioned factors. The availability of some traffic data collection system is simply conditional in order to apply instantaneous and data-driven approaches. In the limiting case no such system is available, only model based approaches are possible. Also, a densely spaced local data collection system (radar, inductive loops, infrared, etc.) enables a much more detailed (multiple input / output) data driven travel time prediction model than for example a data collection system

collecting individual egress and exit times on a tolled route. In the latter case, a time series based approach is the more likely choice. for example instantaneous and data-driven approaches However, apriori no quantitative statement can be made which case would lead to the best (reliable, robust, valid and accurate!) model. The type of input data that *can* be used in a travel time prediction model also depends on modeling approach and prediction horizon, for example model based approaches require OD (origin - destination) flow relationships or traffic demand and supply patterns at the boundaries of the network, route or link of interest, which can not be measured directly, but need to be estimated from raw traffic data.

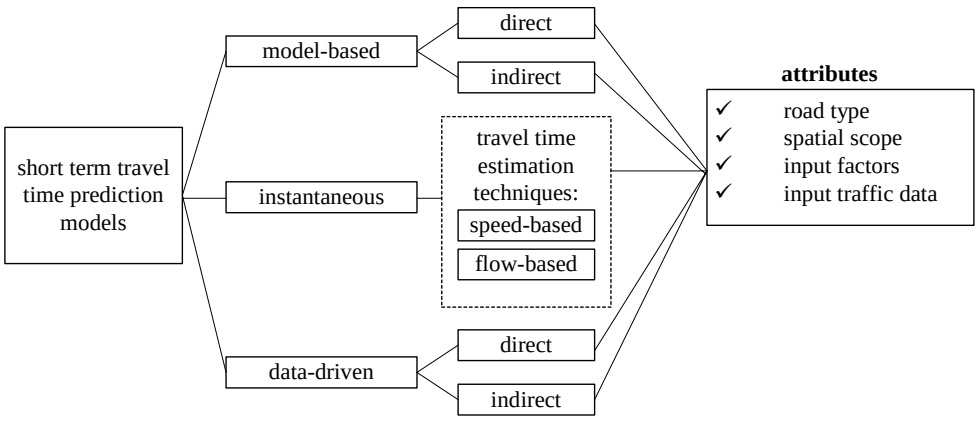
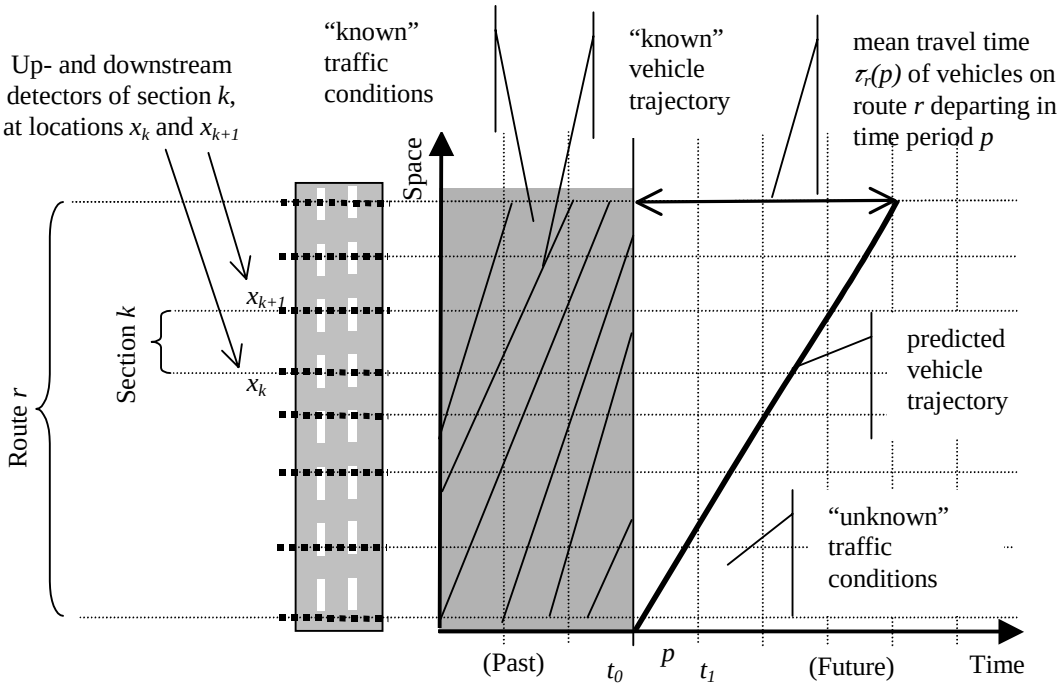


Figure 4.1: Taxonomy for short term travel time prediction models

The factors listed above distinguish between different aspects of travel time prediction models, but do not offer much structural value. Some of these factors may be classified as attributes of particular models, others fundamentally divide between different approaches. We therefore propose the following. We argue the main distinction is between models for different prediction horizons, since long and short term travel time prediction are fundamentally different problems. Focussing on short term prediction the next division is between model based, instantaneous and data driven models. Instantaneous approaches inherently reduce the travel time prediction problem to a travel time estimation problem and hence utilize travel time estimation techniques. The type of travel time estimation procedure (flow or speed based or otherwise) depends on the input traffic data available. Data driven and model based approaches can subsequently be further divided into direct and indirect approaches. The other factors are considered specific attributes which may be different for each approach. Fig. 4.1 schematically outlines this brief taxonomy

4.3 State-of-the-Art in Short Term Freeway Travel Time Prediction

In this section we overview the State-of-the-Art in short term freeway travel time prediction. Recall that *online* prediction is a special case of short term prediction (see section 2.2). Also note that some of the contributions discussed actually aim at short term prediction of other traffic quantities (e.g. traffic flow or speed), but are given the relationship of these quantities and travel time nonetheless interesting to mention. A model predicting for example speeds, could be used in an indirect travel time prediction model. We first present an analysis of the short term freeway travel time problem, such to appreciate the complexity of the problem and (qualitatively) compare the various approaches developed to tackle it.



(a) route *r*, consisting of K-1 adjacent sections, each equipped with up- and downstream detectors at locations x_k and x_{k+1}, k=1,… K, measuring averaged speeds and aggregate flows at discrete time intervals *p*

(b) space-time trajectory of a vehicle leaving in departure time period *p*. The mean travel time $\tau_r(p)$ on route *r* is determined by future (unknown) traffic conditions, which may depend on current and past conditions along *r* (grey area)

Figure 4.2: The general concept of predicting the travel time for a vehicle departing at t_0 on a route r, constituted of $K-1$ adjacent sections, each equipped with up- and downstream detection devices (a). Since that travel time $\tau_r(t_0)$ depends on unknown future traffic conditions, we may predict it using current and past measurements along r (depicted by the grey area in (b))

4.3.1 Schematic representation and overview of the freeway travel time prediction problem

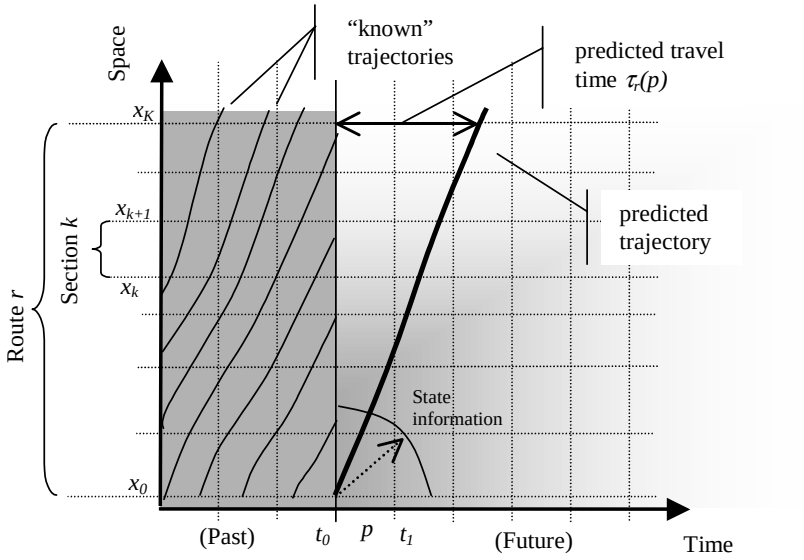
Predicting the travel time on a freeway route requires knowledge on future and inherently "unknown" traffic conditions on that particular route (fig. 4.2). More precisely, to predict the expected travel time of vehicles starting a particular route r at time t_0, we need to know whether or not they will encounter delays (congestion) during the period $[t_0, t_0 + \tau_r(t_0)]$ along their route. The problem, however, is that it is exactly that quantity $\tau_r(t_0)$ (= time spent on the route = travel time), which we wish to predict in the first place. In this sense, travel time prediction is a "chicken and egg" type of problem. Predicting travel times is in fact similar to predicting traffic conditions during period $[t_0, t_0 + \tau_r(t_0)]$, which are governed by complex non-linear interactions of heterogeneous groups of driver-vehicle combinations, each characterized by their own specific technical and behavioral properties, such as vehicle dimensions and acceleration characteristics, drive-style (aggressive, conservative), and motive.

Recall from section 2.2 that predicting $\tau_r(t_0)$ can be mathematically expressed as

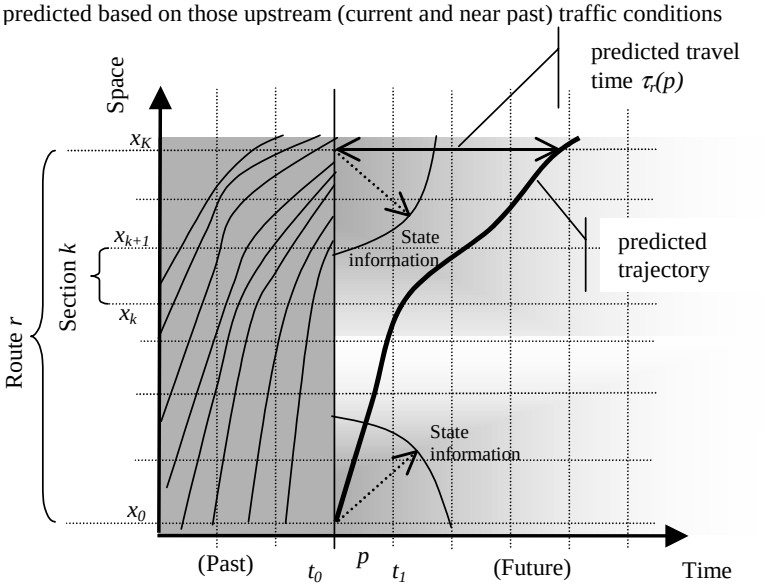
$$\tau_r(t_0) = G(\Omega_r)$$

where Ω_r depict the traffic and ambient conditions (e.g. weather) on route r for time instants $\geq t_0$ and also personal characteristics of drivers. When we solely focus on traffic conditions on route r for time instants $\geq t_0$ macroscopic traffic flow theory provides us with some insight. Since in a practical situation we only know about traffic conditions up till the current time instant, that is $\Omega_r \leq t_0$ traffic flow theory gives us information on how these known conditions may affect conditions in the near future. In macroscopic traffic flow models, so called characteristic curves can be derived, which identify the direction in which state information (mean speeds and densities) along the traffic stream will propagate. In general it is known that in free flow conditions (fig. 4.3a), state information propagates in the same direction as traffic with a speed that is (at least in first order models, see e.g. (Hoogendoorn & Bovy 2001)) assumed equal or lower than the average vehicle speed. This implies in these free-flow conditions we may expect a vehicle departing now to be affected by what currently happens on the upstream half of the route interest. In congested traffic conditions, however, state information may also propagate in the opposite direction. So-called shockwaves occur (fig. 4.3b), as traffic from upstream is forced to slow down due to slower and denser traffic downstream. If the differences in speed and speed-variations of the two “colliding” traffic regimes are large enough, the resulting shockwaves will move in the upstream direction, and even lead to traffic jams (in such cases backward moving shockwaves are model abstractions for queue spill back). This implies that in congested conditions the travel time of a vehicle may also be affected by what currently (or recently) happened on the downstream half of the route or perhaps even on locations beyond the downstream boundary of r. In real life an infinite number of traffic states, ranging from completely free-flowing (no traffic) to completely congested (cars

queued bumper-to-bumber), may occur. The crux is that essentially, we do not apriori know which traffic conditions (states) a vehicle will encounter. A short term travel time prediction model must infer these from current or near-past traffic conditions to subsequently derive travel times. Based on the taxonomy of previous section, we discuss the three modeling approaches to solve the short term travel time prediction problem, that is model based, data driven and instantaneous approaches. Note that we do not address road type and prediction horizon as distinguishing factors, since we focus on short term freeway travel time prediction *only*.



(a) In **freeflow conditions**, state information (vehicle speeds and densities) on route *r*, flows in the same direction as traffic (vehicle trajectories). Predicting future states of sections along *r* then requires only state information from upstream locations on route r (e.g. from detectors x_0 to x_k). Consequently, the trajectory (and hence travel time) of a vehicle departing in period *p* can be predicted based on those upstream (current and near past) traffic conditions



(b) In **congested conditions**, however, state information (vehicle speeds and densities) might also flow in the opposite direction as traffic (vehicle trajectories), due to shockwaves of fast vehicles slowing down to slower vehicles downstream and queue dynamics. Predicting future section states along *r* now requires state information from both upstream locations (e.g. from detectors x_0 to x_k) and downstream locations (e.g. from detectors x_{k+1} to x_K). Depending on the severity of congestion we might even need state information from location downstream of route r (e.g. from detectors $x_d > x_K$). Consequently, the trajectory (and hence travel time) of a vehicle departing in period *p* can only be predicted based on both upstream and downstream (current and near past) traffic conditions

Figure 4.3: Free-flow versus congested traffic conditions. Travel time prediction in congested conditions requires different state information (e.g. average speeds) from different locations than in free-flow conditions.

4.3.2 Model based freeway travel time prediction

The class of model based travel time predictors solve the travel time prediction problem by predicting traffic conditions for a sufficient number of time periods ahead and then subsequently deduce mean travel times from these predicted traffic conditions. In a sense short term model based approaches substitute the "chicken-and-egg" complexity on the route of interest with the problem of predicting the boundary conditions on that route, that is traffic demand at origins (and on ramps) and traffic supply at destinations (off ramps), which determine traffic predictions by vehicular traffic flow models. At the center of model based approaches are traffic flow simulation models which simulate traffic propagation on the network in a rolling horizon. *Microscopic* traffic flow models centre on the prediction of individual vehicle trajectories, based on assumptions on driver-behavior such as car following, gap-acceptance and risk-avoidance. From predictions of the latter category of models mean travel times can be derived directly and used for ATIS (see for example DynaMIT (Ben-Akiva 1998), (Ben-Akiva et al. 2002) and DynaSMART (Hu 2001)). *Macroscopic* traffic flow models aim to predict aggregate properties of a stream of traffic such as vehicular density (vehicles/km), vehicular traffic flow (vehicles/hour), and average vehicle speeds, predominantly based on analogies of vehicular traffic flow with fluid and gas-dynamics. From the aggregate predictions of macroscopic traffic models such as METANET (e.g. (Van Grol et al. 1997), and (Smulders et al. 1999)), mean travel times can be derived indirectly, for example by means of (offline) travel time estimation techniques presented in the previous chapter. For a comprehensive overview on traffic flow modeling see for instance (Hoogendoorn & Bovy 2001).

The clear advantage of model based travel time prediction systems are that they allow inclusion of traffic control measures (ramp metering, routing, traffic lights, and even traffic information) in the prediction, and that they provide full insight into the locations and causes of possible delays on the road network of interest. Furthermore, these model based approaches are generic in the sense that they can be applied even on routes where no detection equipment is installed. Major disadvantages, however, include their computational complexity, the degree of expertise required for design and maintenance, and the fact that they require predictions of traffic demand and supply (capacity) at the model boundaries as inputs. For offline analysis and scenario evaluation purposes model based approaches are invaluable, however, in a real-time setting, the predictive quality of model based travel time prediction systems is strongly influenced by the quality of its (predicted) inputs and boundary conditions.

4.3.3 Instantaneous freeway travel time prediction

The second strand of travel time prediction approaches involve so called *online* or *instantaneous* travel time *predictors* which we defined in the introduction (see also (Bovy & Thijs 2000)). As the previous chapter discussed in detail, travel time estimation pertains to reconstructing vehicle trajectories based on "known" (i.e. historic)

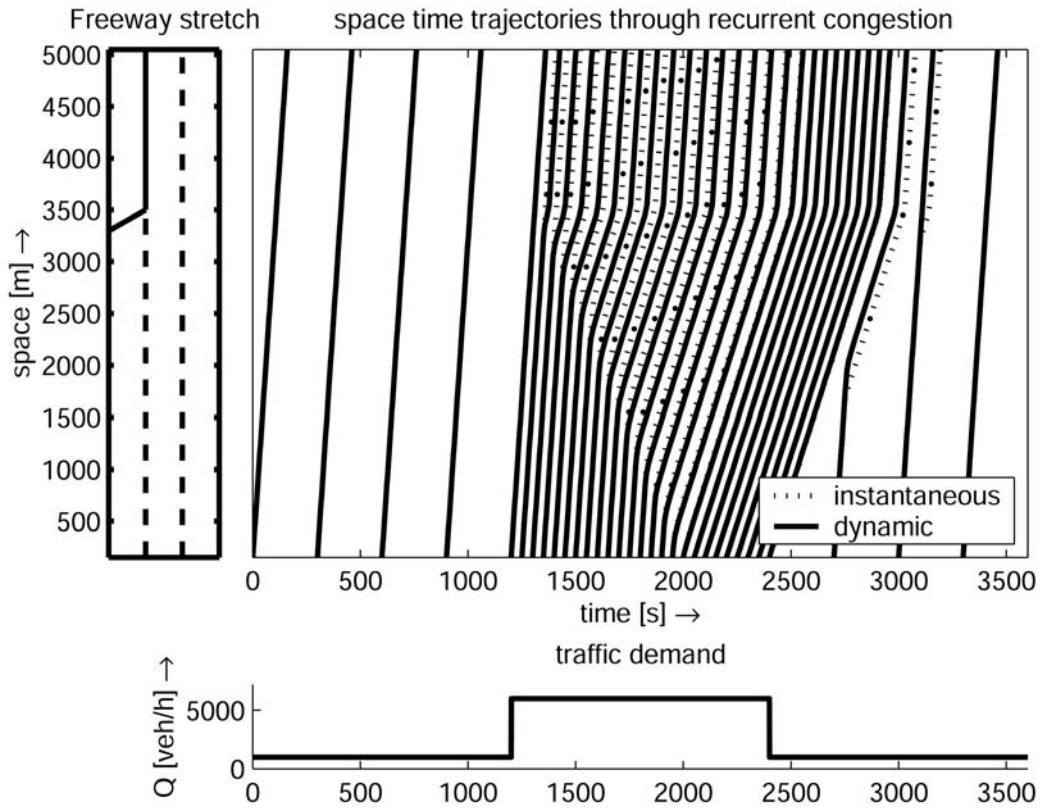


Figure 4.4: Instantaneous travel time versus actual travel time. The figure shows space-time trajectories (top-right) of vehicles traversing a freeway route with a lane drop (left). Particularly as congestion sets in (dissolves), instantaneous trajectories underestimate (overestimate) travel times. The traffic conditions on the freeway are generated by a first order traffic flow model (Lighthill & Whitham 1955).

traffic conditions. Inherently, travel time estimators can only provide estimates of past travel times. If it is assumed, however, that current traffic conditions remain stationary for an indefinite time period, then travel time estimates based on these stationary conditions can serve as travel time predictions. As mentioned in chapter 2, these models are referred to as instantaneous travel time predictors.

Clear advantages of instantaneous travel time predictors are their low computational burden, mathematical simplicity and ease-of-implementation. Especially the latter argument would favor the instantaneous approach. Since only current conditions are considered, instantaneous travel times can be derived for all freeway sections (links) simultaneously. Subsequently, the instantaneous travel time for any freeway route constituted of an arbitrary number of adjacent freeway sections can be derived by simply summing up section level travel times. In (Zhang & Rice 2003) and (Rice & Van Zwet 2001) route level travel times are predicted through a linear regression of the sum of current instantaneous section level travel times and historical (in fact PCSB estimated - see previous chapter) travel times. Also, these authors argue their approach *"does not aim for sophistication but for simplicity and speed"* and *"needs to be fully*

scalable to the very large amounts of data that must be processed to be able to advise motorists in real-time" (Rice & Van Zwet 2001), which indeed is a capacity instantaneous predictors provide. In the Dutch situation, the so-called Astrival algorithm is used to "predict" section level travel times ((Van Toorenburg 1998) - in Dutch). This algorithm combines a flow-based algorithm and an instantaneous speed based travel time predictor to derive section level travel times, which are aggregated (summed up) at path level and subsequently distributed to a number of service providers by the Dutch national Traffic Information Centre. The basic principle behind this combined method is that it puts more weight on the flow based module in congested traffic conditions, during which the speed based method becomes more unreliable. The main causes reported in (Van Toorenburg 1998) are the inability of the latter to detect a queue between detectors and the inaccuracy of speed measurements for low speeds. As the previous chapter outlined, this is amongst other things due to the arithmetic averaging speeds, and more generally to the inability of local detectors to measure zero speeds. Although the flow-based component accounts for delays due to queueing traffic, it still qualifies as an instantaneous predictor, since it does not account for inflow or outflow nor queue dynamics in future time periods.

The main problem with instantaneous travel time prediction is that the assumption of stationarity does not hold in congested or transient traffic conditions, and as shown in e.g. (Lindveld & Thijs 1999), (Thijs et al. 1998*a*) and (Thijs et al. 1998*b*), the performance generally deteriorates fast in congested conditions, which is precisely when accurate travel time predictions are most valuable. Since the errors made are not random, aggregation of section level instantaneous travel times - especially on longer routes - may lead to large errors at the route level. To illustrate the difference between instantaneous travel times and 'actual' or dynamic travel times, consider the following examples in which we use first order traffic flow theory[2] to simulate traffic on a three lane freeway stretch of 5000 metres. In both examples, imaginary vehicle trajectories are drawn through space-time by means of the PLSB trajectory method. The first example (Fig 4.4) shows vehicle trajectories as congestion sets in due to oversaturation (demand $>$ capacity at a bottleneck). As congestion sets in, instantaneous travel times (derived from vehicle trajectories based on stationary conditions - dotted trajectories) underestimate the real travel times (the difference between arrival and departure times). As congestion dissolves the effect is opposite: instantaneous travel times overestimate actual travel times. The second example (Fig 4.5) shows the effect of a temporary blockade, due to an accident. Here, the difference between instantaneous travel times and "actual" travel times is even larger. Due to the blockade, vehicles come to a virtual stand still, seriously increasing their travel times. Instantaneous travel times of vehicles departing just before the blockade occurs are much lower than the actual travel times, while instantaneous travel times of vehicles departing during the blockade overestimate the delay significantly.

[2]For details on the LWR first order traffic flow model see for example Appendix D.

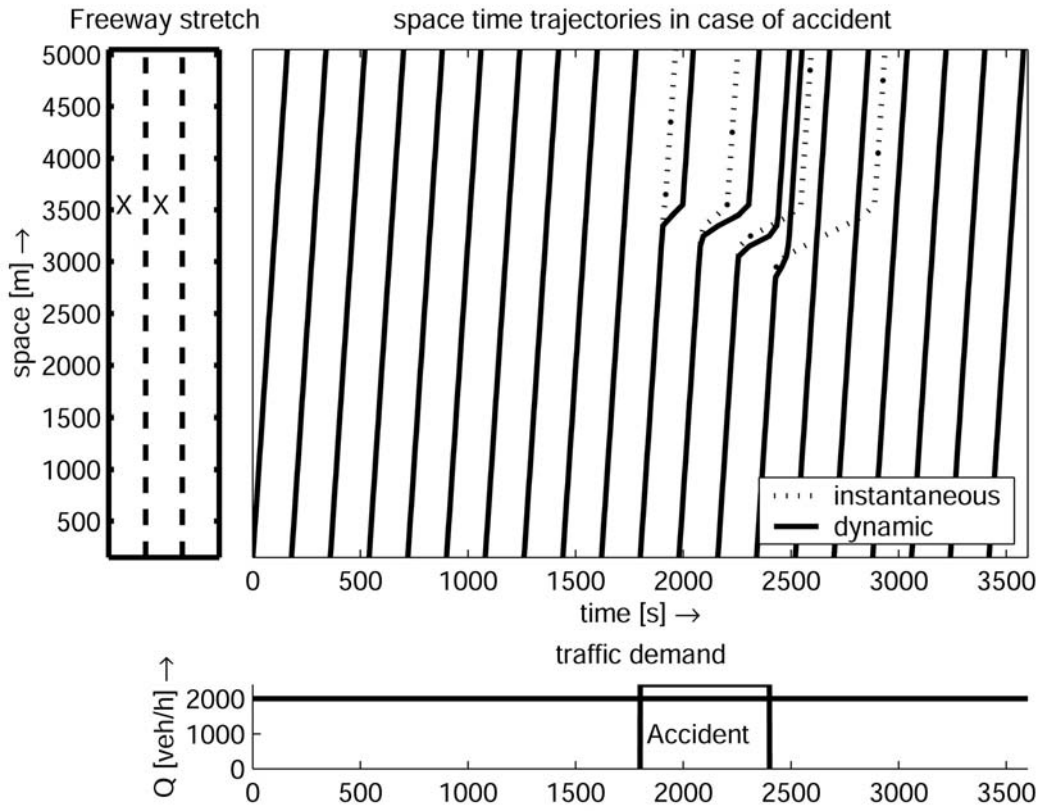


Figure 4.5: Instantaneous travel time versus actual travel time. The figure shows space-time trajectories (right) of vehicles traversing a freeway route on which an accident occurs. Clearly, during the time of the blockade instantaneous trajectories produce large deviations in their resulting travel times. The traffic conditions on the freeway are generated by a first order traffic flow model (Lighthill & Whitham 1955).

4.3.4 Data-driven freeway travel time prediction

The third strand of short term travel time prediction models we classify as the so-called data-driven or inductive models. The principal difference between data driven and model based approaches is that the data-driven approaches consider the traffic processes generating travel times as *black boxes*, and exploit purely inductive techniques to either directly or indirectly predict travel times (see section 2.2 for these definitions). These approaches include ARIMA(X) models (Williams 2001), non-linear time series modeling (Ishak & Al-Deek 2002), (D'Angelo et al. 1999) linear (Rice & Van Zwet 2001), (Zhang & Rice 2003), (Sun et al. 2003), and support vector regression models (Chun-Hsin et al. 2003) feed-forward (Innamaa 2001), (Rilett & Park 2001), (Park & Rilett 1999), (Cheu 1998), (Park & Rilett 1999), (Huisken & Van Berkum 2003) and recurrent (Abdulhai et al. 1999), (Dia 2001), (Ishak et al. 2003), (Van Lint et al. 2002*a*), (Van Lint et al. 2002*b*) neural networks and various hybrid approaches (Chen et al. 2001), (Park & Rilett 1998), (Ishak & Alecsandru 2003), which for example use neuro-fuzzy approaches, or combinations of different ANN topolo-

gies. All these data-driven methods have in common that they correlate mean (observed) travel times or traffic conditions to current and past traffic data, without explicitly addressing the (physical) traffic processes that determine these travel times as model based approaches do.

A distinction must be made between approaches that regress *route* travel times from current or near-past *point- or section* level traffic data (e.g. (Van Lint et al. 2002*b*)) and approaches that regress route travel time from current or past route level traffic data (in most cases travel time, e.g. (Park & Rilett 1999)). The difference lies in the time instant on which "current" data becomes available. Actual travel time measurements become available *only after a vehicle has finished the entire route*, while point measurements (speeds, flows) become available after each aggregation period (of typically a minute). This means that a one-step ahead travel time prediction model trained on time series of actual travel times is in most cases trained to predict *past* travel times and not travel times of vehicles departing at the time of prediction, except in trivial cases where the route travel time is shorter than the time periods after which data becomes available. Thus, counter-intuitively, a travel time measurement system (especially on longer freeway stretches) may not be the most appropriate data collection system for online travel time prediction purposes.

In neural network terms, most approaches (except recurrent) are feed-forward type models (no feedback within the model). ARIMA and multivariate regression approaches can also be classified as feed-forward type of approaches (see e.g. (Kay & Titterington 1999) or (Bishop 1995)). The data used range from flow, speed or occupancy data from local detection systems (e.g. (D'Angelo et al. 1999), (Zwahlen & Russ 2002), (Kwon et al. 2000), (Huisken & Van Berkum 2003)) probe vehicle data (Chen & Chien 2001), travel times from AVI systems (e.g.,(Rilett & Park 2001), (Innamaa 2001), (Park & Rilett 1999))or combinations of data sources (Zhang & Rice 2003). As a general rule, the models presented in these studies outperform other (less sophisticated) models on a particular location or route on a particular test data set. Also, as a general rule, one may conclude that (tailor made) data-driven methods are very efficient in predicting either travel time, speed or flow, since the references here are a small but representative subset of all studies that report successful applications of data-driven methods to travel time prediction[3].

The clear advantages of data-driven approaches are that they do not require extensive expertise on traffic flow modeling, many ready-to use software packages are available for model design and calibration, they are fast and easy to implement, and specifically neural network approaches have proven accurate and reliable traffic predictors (Dougherty 1995). There is, in the light of the objectives of this dissertation thesis, however, one major drawback from which all data-driven approaches, including sophisticated neural networks, suffer. They are all *location specific* solutions, requiring

[3]A search with keywords "*travel* and *time* and *prediction*" on three large-scale literature databases (Web-SPIRS / transport, Web-of-Science, and PATH) yielded several hundred (unique) articles and reports related to data-driven approaches from the last 8 years only.

significant efforts in input- and model selection for each specific application. Location specificity does not reflect the fact that parameters need to be calibrated for each different location. In this sense even a model based approach is location specific (e.g. network specification, fundamental diagram, etc.). Location specificity here pertains to the mathematical structure and the input-output mapping that constitutes the model. To illustrate this, let us assume we have a data driven model for short term freeway travel time prediction, which uses measurements from a subset D_k of detectors of the last T measurement periods (the so-called look-back interval) along route r and can be formulated as follows

$$\begin{aligned} \tau_r(p) &= G(\mathbf{u}(p), \psi, H), t \in [t_0 - T, t_0] \\ \mathbf{u}(t) &= \{..., \mathbf{u}_k(t), ...\}, k \in D_k \end{aligned} \tag{4.1}$$

where ψ depicts a vector of all the parameters in the data driven model model G, and H encompass all other modelling assumptions, such as the model structure (ARIMA, regression model or artificial neural network) and for example distributional aspects of input and output. As explained in section 4.3, we might need very different settings of D_k and T to accurately predict travel times in different circumstances. Then, due to the complex non-linear nature of the travel time prediction problem, an in principle infinite number of different combinations of models G, assumptions H, parameter vectors ψ and input combinations $\mathbf{u}(t)$ might be suitable for the problem. A model designer may use trial and error procedures (e.g.. in (Fallah-Tafti 2001)), cross-correlation analysis (e.g. (Innamaa 2001)), generic ARIMA model design principles (mostly based on (Box & Jenkins 1976)) or genetic algorithms (Abdulhai et al. 1999), (Lingras et al. 2002) to properly set up his model in terms of G, H, ψ and $\mathbf{u}(t)$. What results from such exercises is a model that is tailor-made for the problem at hand, that is, the particular geometry and detector configuration, the data set which was available and other application specific circumstances. These models for one location are (typically) not transferable to the next, due to those location-specific circumstances (geometry, traffic control, etc.). For isolated applications this is not a problem, however, in case of larger scale (network-wide) deployment transferability is required, especially in terms of use, consistency, and maintenance.

4.3.5 Discussion and comparison of approaches

Although traffic flow models provide us with valuable insight into the mechanisms of traffic flow and queue dynamics, this does not automatically imply that these models provide accurate short term travel time predictions. We argue that model based approaches serve mainly as scenario evaluation tools, providing the modeler with invaluable insight into the causes and effects of different traffic demand and supply settings, but also of network alternatives and dynamic traffic management measures. As a travel

time prediction tool, traffic flow models suffer from the requirement of *predicted* input, that is traffic demand at origins and traffic supply at destinations on the route or network of interest, to predict the traffic conditions on that route or network. The accuracy of the model's output, even if the model reproduces traffic patterns very accurately, can only be as good as the predictive accuracy of its inputs. Moreover, setting up real-time traffic flow models for online or short term travel time prediction on many routes on a freeway network would require a lot of modelling effort both in terms of design and maintenance.

Inductive approaches (regression, ARIMA/ARX models, neural networks) generally consider the traffic processes producing travel times as black boxes and aim to correlate either speeds or travel times to traffic measurements (speeds, flows, occupancies) from particular locations and time instants along the route of interest (see eqn 4.1). As was shown above, different but apriori unknown traffic conditions require very different input and model settings to produce good results in all possible situations, yielding a very difficult task for the model designer. Moreover, due to location specific circumstances, a solution that works well on one location may not work at all on the next. Nonetheless, there are many successful applications reported on tailor-made data driven approaches to short term freeway travel time prediction.

In sum, travel time prediction is a complex non-linear spatiotemporal problem for which the dynamics in free-flowing or congested conditions are different. To tackle this complexity, we need either sophisticated traffic flow models and prediction algorithms for the boundary conditions or non-linear data-driven models that are able to learn the non-linear dynamics of travel time from data directly.

4.4 Summary

This chapter gave a taxonomy and overview of travel time prediction models, focussing on short term forecasts on freeways. While for travel time estimation we do not need to bother with the dynamics of traffic flow processes, this is no longer true for travel time prediction. We illustrated that to solve the travel time prediction directly, we have to tackle a "chicken-and-egg" complexity, that is the input required to predict travel time on a route depends on the outcome of that prediction. We could circumvent that difficulty by using traffic flow simulation models to predict traffic conditions for a sufficient number of time steps ahead and predict travel time indirectly by using a travel time estimation (e.g. the PLSB method) on those predicted traffic conditions (speeds). To do this, however, we require predictions of the boundary conditions of the route of interest, meaning that one complexity is substituted for another.

The conclusion is that travel time prediction is a complex spatiotemporal problem, for which we require either sophisticated traffic flow models and prediction algorithms for the boundary conditions, or intelligent inductive models that are able to learn the complex traffic dynamics from data on the route of interest directly. In the next chapter we

propose a new data driven approach to short term freeway travel time prediction that combines insight from traffic flow models with the learning and generalization capabilities of a particular class of inductive models, recurrent artificial neural networks.

Chapter 5

Short Term Freeway Travel Time Prediction with State Space Neural Networks

5.1 Introduction

In the previous chapter we illustrated the complexity of the travel time prediction problem and gave an overview of the various approaches that have been developed to tackle it. We illustrated the benefits and disadvantages of the various approaches and concluded that for travel time prediction either sophisticated traffic flow models combined with accurate forecasts of the boundary conditions (traffic demand and supply) or accurate data driven approaches that are able to learn the complex nonlinear and spatiotemporal dynamics of traffic flow from data are required. In this chapter we propose a novel method within the category of data driven (inductive) models. Based on the taxonomy presented in the previous chapter, the model developed here can be classified as a data driven online travel time prediction model on freeway routes (of adjacent links/sections) for on-trip application (ATIS such as In-Car navigation systems and VMSs). Due to the online (short term) nature, we limit the input factors to network effects on the route of interest. In chapter 9 extensions to this particular limitation will be discussed.

To this end we first address the issue of how to model time-dependent (dynamic) problems with inductive (data driven) methods. Given the non-linear nature of the problem the focus thereby is on artificial neural networks (ANNs), which represent a very general class of data driven models[1]. We argue that the class of spatiotemporal (recurrent)

[1]Many classical statisticial models could also be classified as ANN models, for example linear regression models (special class of multi layered perceptrons (Bishop 1995)), and state-space models (special class of recurrent neural networks (Dorffner 1996)). For a comprehensive treatment we refer to for example (Kay & Titterington 1999).

neural networks is most appropriate for this task. In formulating the travel time prediction problem in state space form (much like a traffic simulation model), a particular class of recurrent neural networks (RNN) is obtained, which is suitable for the short term freeway travel time prediction problem, the so-called state space neural network (SSNN, (Van Lint et al. 2002*a*), (Van Lint et al. 2002*b*)). After the SSNN is derived mathematically and discussed, methods for training (calibrating) the SSNN are presented. The second part of this chapter discusses the predictive power and internal workings of the SSNN in detail, based on synthetic data, obtained from a microscopic traffic simulation tool. It appears that the SSNN is much more accurate than current approaches and well suited for ATIS purposes. Moreover, its internal states are strongly correlated with the actual traffic processes. We close this chapter with a critical discussion and a brief summary of the main findings.

Based on the SSNN model developed in this chapter, we will address the issues of reliability and robustness and real-time application in the ensuing three chapters. Finally, in chapter 9 we will return to the SSNN development and propose possible extensions in terms of application area (e.g. urban road networks), input and output domain (e.g. data from sections elsewhere on the network, other data sources such as weather, roadworks) and discuss implementation issues not covered earlier in this dissertation thesis.

5.2 Modeling Dynamic Processes with Artificial Neural Networks

Since travel time prediction is a complex dynamic problem, it requires a data driven model capable of dealing with dynamic processes. For artificial neural networks (ANNs) roughly two approaches exist.

1. Inputs at different time instant are concatenated in a single vector of fixed length, which the ANN model processes in the same way as it does with spatial input vectors (containing different inputs from the same time instant). Although so-called time-delayed neural networks (TDNN) are presented (schematically) differently than standard FNNs, TDNNs also treat past signals in the same way. In a time series context this approach is considered a Auto Regressive (AR) approach

2. Time is explicitly accounted for in the structure (topology) of the ANN, that is, the ANN has some sort of memory, in which it stores previous outputs or hidden neuron activities. It receives input signals (vectors) from one time instant only and combines these with signals (vectors) from its memory to make predictions. In a time series modelling context this would be considered a Moving Average (MA) approach.

In the previous chapter we showed that many of the ANN solutions (FNN, TDNN) for travel time prediction are based on the first (AR) approach. As (Elman 1990) convincingly argues, this approach has several serious drawbacks. In the next section we will elaborate on the two main problems he stipulates, which we refer to as the input selection problem and the semantic problem respectively. Next, we will briefly introduce the class of ANNs based on the second (MA) approach, the so-called recurrent neural networks (RNNs) or spatiotemporal neural networks, which (partially) solve these problems, but introduce a number of new problems themselves. In the discussion that follows we outline the pros and cons of both approaches and conclude that for the short term freeway travel time prediction problem a RNN solution is most suitable.

5.2.1 Treating time series as a fixed length input vector

This (AR) approach requires that the model interfaces with the real world (in our case a traffic data collection system) by means of a shift register at its inputs (see for example fig. 5.1). A time series is presented as a fixed length input vector. As fig. 5.1 shows, this requires an apriori choice of which inputs at which time lag is needed to capture the dynamics of the problem at hand. A direct consequence of modeling dynamics in such an AR type of fashion is that memory depth (the number of previous time steps the model considers in predicting a new datum) is limited to the size of the input shift register.

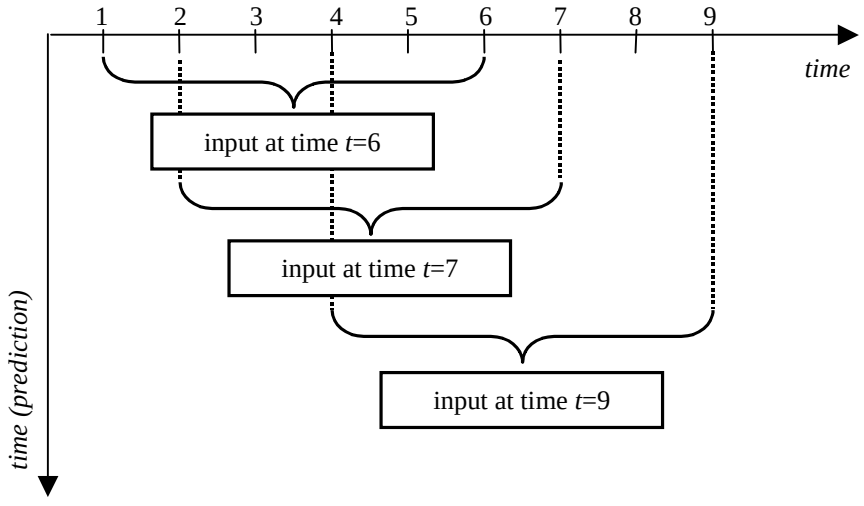


Figure 5.1: Treating time as a spatial pattern by means of a shift register. In this example the input of a model includes a time series of the last six measurements of one particular data source (e.g. an inductive loop detector). For example, at time instant $t = 9$ the input vector consists of measurements from time instants {4, .., 9}.

The semantic problem

The main problem of converting a time sequence into a spatial pattern is that this, regardless of the solution method, may seriously increase the complexity of solving the underlying problem. For example, consider three sequences of binary data [1, 0, 0, ...], [0, 1, 0, ...], and [0, 0, 1, ...], produced by some serial device (e.g. a tape streamer). Clearly, these sequences only differ from each other by one time lag. However, when these patterns are interpreted geometrically, they are in fact very dissimilar vectors occupying very distant corners in input space. In the context of the travel time prediction problem, two in time subsequent measurements on a particular route may be classified by a static data driven travel time predictor as two completely different input vectors, while in fact they represent approximately the same traffic pattern at subsequent time instants. This (artificially) added complexity directly effects the required complexity of the ANN solution: learning an ANN that the three binary sequences above yield very similar outputs may require an unnecessary complex (large) neural network.

The input selection problem

The second problem of treating time as a spatial dimension pertains to the ANN design. By explicitly defining the size of the input time series (the shift register), the total size of the input vector should equal the largest possible input vector that could occur in real life i.e. the largest input vector necessary to solve the problem under all observable conditions. In general, this puts a rigid constraint on the type of input patterns the model can recognize. For the travel time prediction problem in particular, one has to deal with the "chicken-and-egg" complexity, as explained in section 4.3, which states that different input configurations are required in different, but apriori unknown traffic conditions.

Recall eqn 4.1 on page 92 (section 4.3). As explained there, very different settings of D_k (the subset of inputs we choose for the model) and T (the time lags associated with those inputs) might be required in order to accurately predict travel times in different circumstances. After choosing an appropriate model structure (H), a model designer may apply his (traffic) engineering judgement, rely on statistical techniques such as correlation analysis or principal component analysis (PCA) or use for example genetic algorithms to select the appropriate inputs with the appropriate time lags. The principal concern for the model designer in doing so, is to balance the size of the input vector and hence the size of the parameter vector ψ with the predictive power of his model. Larger input vectors imply more model parameters; a larger parameter space yields more difficulty in calibrating the model to not only fit the particular data used for calibration, but also produce a model that generalizes well to "unseen" data. On the other hand a too simple structure (too few parameters) limits the descriptive and predictive power of the model.

As we will show in subsequent sections, there is a fundamental trade off between

model complexity and model generality. Overly complex models are prone to over fit the problem and hence generalize poorly; models that are too simple are simply inadequate to capture all the nonlinearity of the problem at hand[2].

5.2.2 Treating time sequentially: spatiotemporal neural networks

Rather than representing time series as a fixed length input vector, we can also present inputs of consecutive time instants to our model sequentially, and built a feedback (memory) mechanism in the model itself. From a traffic theory point of view, this is the most natural way of treating time. In macroscopic traffic flow simulation models for example, spatiotemporal traffic patterns are captured either by continuum differential equations or discretized difference equations (state space models), in which the state of a particular road section (average speed and vehicular density) is defined completely by its previous state and the inputs in the previous time period (Hoogendoorn & Bovy 2001).

There is a wide range of neural network models that are able to learn temporal sequences of spatial patterns (or scalar inputs for that matter). In general these are known as recurrent or *spatiotemporal neural networks*. A wide variety of application domains exploit these recurrent neural networks such as dynamic system identification and control, speech recognition and grammatical induction (Kremer 2001). Like the (among traffic scientists) more widely known and used feedforward neural networks (FNNs), recurrent neural networks (RNNs) contain an adjustable parameter vector ψ, which determines how signals are propagated through the neural network. These parameters are fixed after training (calibration) and can be viewed as long term memory. RNNs principally differ from FNNs, in that they also incorporate a *short term memory*, allowing them to dynamically deal with input and output patterns that vary over both time and space. For an extensive review and taxonomy of spatiotemporal neural networks we refer to (Kremer 2001). A consequence of a short term (hidden) memory is that the memory depth is no longer restricted to the size of an input time window. In principle, memory depth is infinite, albeit that due to exponential transfer functions, the effect of past information decays also exponentially. In later sections we will return to memory depth and quantitatively analyze how it affects processing.

5.2.3 Motivation of approach

Although the above may indicate that RNN type models are generally more adequate than FNN models in solving dynamic problems, these models also have some limitations and difficulties. First, incorporating time implicitly in a neural network topology

[2]We will address this problem again in section 5.4 when we discuss the training of the SSNN model and also in chapter 7, in which we discuss confidence and prediction intervals around the SSNN predictions.

(e.g. by means of recurrent connections or a so-called memory layer) also increases the number of parameters and may hence also lead to overly complex models. Second, the implicit dynamics in an RNN yield a much more complex training task (see e.g. chapter 6 of (Hecht-Nielsen 1990)), than is the case with FNN type models[3]. Finally, there are many successful examples of static (feed-forward) neural networks for solving dynamic problems in traffic prediction (see previous chapter) but also in many other fields (see e.g. chapter 10 of (Demuth & Beale 1998), or (Kappen & Gielen 1997)).

Since there exist efficient and mathematically sound methods for controlling the complexity of ANN models (see for example section 5.4), a large number of parameters in the ANN solution (either FNN or RNN) does not necessarily pose a problem, although we do advocate also here the principle of parsimony (which favors simpler solutions if they suffice (Box & Jenkins 1976)). The semantic problem, however, is inherent to FNN solutions for dynamic problems and difficult to tackle especially in parameter spaces of high dimensionality. Therefore, as a general rule, we argue that if the number of inputs to a problem is not too large, and either sound theoretical or statistical methods exist to determine the appropriate inputs and input time lags, a feed-forward ANN approach is certainly suitable for dynamic problems. In this case the FNN solution is in fact a very general non-linear time series or dynamic regression model. However, in case the number of inputs is relatively large and the problem is characterized by complex dynamics over both space and time a RNN (spatiotemporal neural network) is the more appropriate modeling choice[4].

Since freeway travel time prediction classifies as a complex spatiotemporal problem for which we apriori do not know which inputs from which locations we need as inputs (recall the "chicken-and-egg-complexity" discussed in section 4.3), a spatiotemporal solution is required. In the next section we show that by formulating the travel time prediction problem in state space form, we are naturally led to a particular type of spatiotemporal neural network, which we will refer to as the state space neural network (SSNN). The advantages of this model are its straightforward design, which is based on the geometry and detector configuration of the route of interest. Since time is modelled implicitly, as a direct consequence, the SSNN does no longer require tedious input selection procedures. Rather, selecting which input from which (past) time instant produces relevant information in a particular traffic situation becomes an intrinsic part of the neural network training procedure.

[3] In some cases (particularly in Elman type recurrent neural networks) it is, however, possible to train the model in a feedforward fashion.

[4] In this thesis we deliberately refrain from quantitative statements on which model is "best" for a particular application, since it is almost allways possible in a particular situation to taylor-make some (parameterized) model such that it outperforms all other models on a particular performance criterion.

5.3 Derivation of the SSNN Model

Since traffic is characterized by complex patterns over both space and time, we seek a neural network topology capable of simulating the evolution of spatiotemporal input-output mappings. Since we also require that the topology of this neural network should be derived from traffic-related considerations rather than from black box approaches such as genetic algorithms or correlation analysis we base our efforts on a so-called discrete state space model (DSSM) (see also (Van Lint et al. 2002*a*), (Van Lint et al. 2002*b*)). A DSSM contains a dynamic equation governing the state dynamics and an output equation that maps the current states to the model output.

This particular choice is in line with macroscopic traffic flow theory[5]. In macroscopic traffic flow models (see for example (Papageorgiou et al. 1989)) the section traffic states (densities and speeds) on a route are a function of previous traffic states and current inputs (traffic demand and supply on the route boundaries) only, where a route consists of several adjacent sections. A section in this context is a stretch of several hundred up to a thousand metres of highway. The output equation is usually a static mapping from densities to traffic flow (called the fundamental diagram). Analogously, the travel time prediction DSSM contains a dynamic equation for section specific travel times (or delays), while the output equation maps these section states to the mean travel time on the entire route for vehicles departing at time t. It has the following general form[6]

$$\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{u}(t), \mathbf{V})\ state\ equation \tag{5.1a}$$

$$y(t) = G(\mathbf{x}(t), \mathbf{w})\ output\ equation \tag{5.1b}$$

with $\mathbf{x}(t) = \{..., x_m(t), ...\}$, $m \in \{1, ..., M\}$, where M depicts the total number of adjacent sections comprising route r. As is shown in (5.1a) and (5.1b), the state $\mathbf{x}(t)$ of the sections on time instant t, is uniquely defined by the previous state-vector $\mathbf{x}(t-1)$ and the section-specific input-vectors $\mathbf{u}(t) = \{..., \mathbf{u}_m(t), ...\}$ of the time period $[t-1, t]$, which contain speeds and flows from measurement locations on the sections, and in- and outflows from on- and off-ramps, connecting to the sections[7]. Note that the state $x_m(t)$ of a single section depends not only on its own previous state but also on the previous state of other sections along the route of interest. This reflects the effect of state information flowing in both up- and downstream directions, dependent on which traffic conditions prevail (free-flowing or congested). The output function, in this case a scalar function $y(t)$, calculates the mean travel time for a vehicle starting the route at time instant t, and takes a vector $\mathbf{x}(t)$ of all section states at time instant t as inputs.

[5] An example (the LWR model (Lighthill & Whitham 1955)) of a macroscopic traffic flow model discretized in state-space form can be found in Appendix D

[6] in this case the output function is a skalar, in more general DSSMs this the output function is a vector $\mathbf{y}(t)$.

[7] Note that the state dynamics are sometimes defined as $\mathbf{x}(t) = F(\mathbf{x}(t-1), \mathbf{u}(t-1), \mathbf{V})$, in which case $\mathbf{u}(t-1)$ depicts the input obtained in time period $[t-1, t]$.

Finally, **V** and **w** denote parameter vectors associated with the dynamic equation (5.1a) and output equation (5.1b) respectively, which both can be adjusted during calibration.

Given an appropriate choice of functions F and G (see below) this DSSM is in fact a One Layer First Order Context (FOC) Memory in the taxonomy of (Kremer 2001) - similar to the Elman RNN proposed in (Elman 1990) (see fig. 5.2). Also (Dorffner 1996) showed that an Elman network in fact is a specific realization of a general DSSM. We will refer to it from hereon as the state space neural network (SSNN), see fig. 5.3. The signals produced by the hidden layer will be referred to here after as the *internal states* of the SSNN. The context layer effectively provides a short term memory for these internal states.

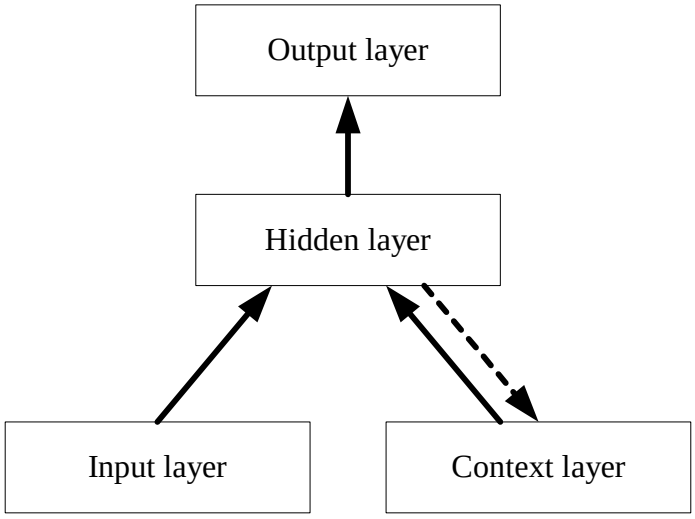


Figure 5.2: The Elman Recurrent Neural Network ((Elman 1990)). This neural network contains a so-called First Order Context Memory (FOC), in the form of a context layer, which stores the hidden neurons states of a previous time step.

A detailed mathematical description of the SSNN is given in appendix B, here we provide the most relevant equations. A neuron j in both hidden and output-layer calculates its output z_j as a weighted sum of its inputs u_{ji} and a bias w_{j0}

$$z_j = w_{j0} + \sum u_{ji} w_{ji} \tag{5.2}$$

Subsequently, the output is transformed by the well-known sigmoid transfer-function

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{5.3}$$

which can be written in matrix form $\Phi(\mathbf{z}) = (\phi(z_1), \phi(z_2), ..., \phi(z_M))^T$. Using matrix notation the SSNN can now be written in the same state space form as the generic DSSM model (eqns 5.1a and 5.1b) and reads

$$\mathbf{x}(t) = \Phi\left(\upsilon_0 + \upsilon \mathbf{u}(t) + \nu \mathbf{x}(t-1)\right) \tag{5.4}$$

$$y(t) = \phi\left(\omega_0 + \omega \mathbf{x}(t)\right) \tag{5.5}$$

where v_0, v, and ν denote the bias and weight vectors associated with the hidden layer (comprising **V**), $\mathbf{u}(t)$ denotes a concatenated vector of all section specific input vectors, and ω_0 and ω the bias and weight vector(comprising **w**) of output layer. The choice of a nonlinear (logistic) output function is arbitrary, a linear function would also suffice. Summing up all inputs (speeds, flows) and internal states implicates that we cannot assign a direct physical quantity to the hidden neuron outputs (the internal states). We can, however, interpret each hidden neuron's output $x_m(t)$ as a metric (scalar) representative for the amount of delay (travel time) on section m at time $(>)t$.

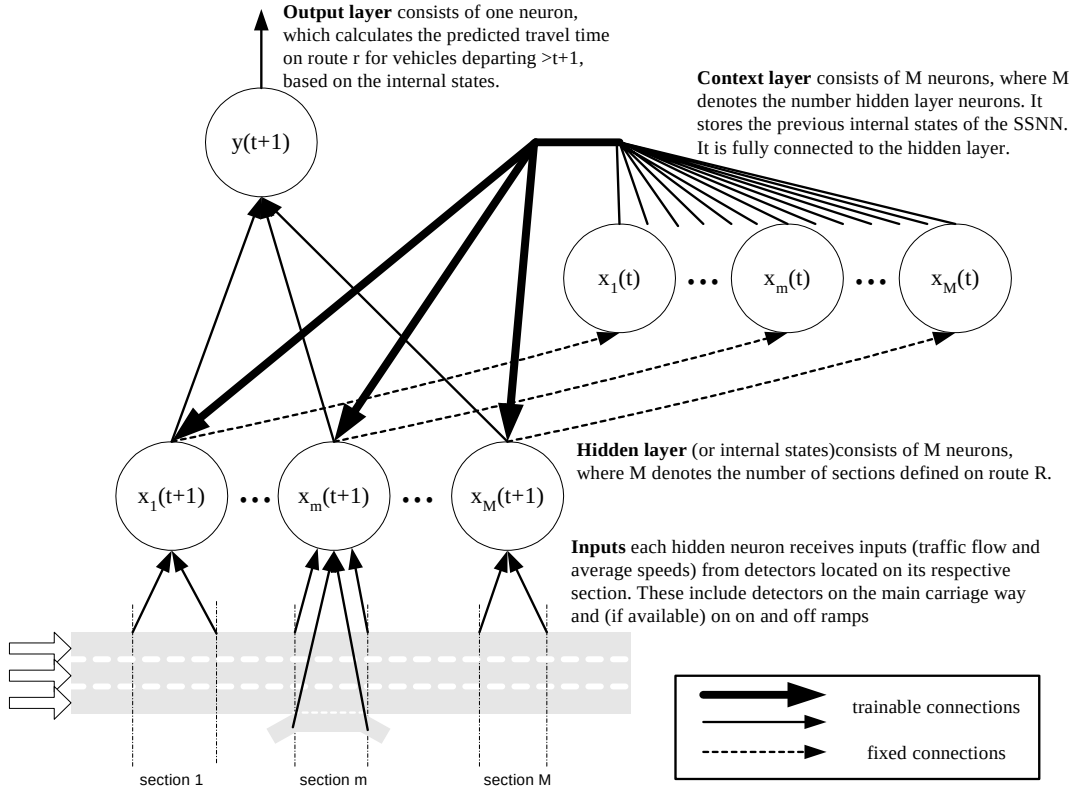


Figure 5.3: State-Space Neural Network (SSNN) topology for short term freeway travel time prediction.

Finally, a note needs to be made about the temporal resolution of the SSNN, that is the unity of one time step t. In macroscopic traffic flow models this resolution is constraint by the minimum amount of time state information requires to traverse over sections (the lower-bound of which is the time the fastest possible vehicle would require to traverse the smallest section). Practically, this results in a temporal resolution in the order of seconds or even smaller. In case of the SSNN model this constraint does not apply, since we feed back previous information from *all* sections to the current state of a particular section. The temporal resolution hence is constraint by the time required to traverse the entire route, which is the minimum travel time. For any real-time application, this is in the order of minutes. We therefore choose a resolution that is practical in terms of its inputs (data from detectors) and equals 1 minute.

In sum, the SSNN predicts travel times based on current inputs in the context of its previous internal states. Both topology and input configuration in the SSNN are completely defined in terms of the lay-out of the route on which we apply it. A route consisting of 10 adjacent sections would result in a SSNN model with 10 hidden neurons each of which are connected to the input signals associated with each section. The temporal dynamics are captured by means of the context layer, which serves as a short term memory for the internal states of the SSNN. Context and hidden layer are fully connected to allow the model to learn the different dynamics during free flow and congested conditions, during which it may have to look at different measurements from different locations. As a consequence, we might apriori expect that the SSNN may not be suitable for routes that are too long, or routes on which travel time is predominantly a function of what happens elsewhere on the network.

5.4 SSNN Training

5.4.1 General concept: regularized training

In general terms, the SSNN can be trained (calibrated) in a supervised[8] manner to approximate a particular parameterized mapping

$$y(t) = G(\mathbf{u}(t), \psi) \tag{5.6}$$

given sufficient and representative data pairs $\{\mathbf{u}(t), o(t)\}$ are available. In (5.6) $o(t)$ denotes the output of the real process (mean travel time for a vehicle starting at t) given input pattern $\mathbf{u}(t)$ (a vector of measurements of detectors on route r). In this context $y(t)$ denotes the model output, and ψ denotes a vector containing all adjustable parameters (weights and biases) in the model. The task of learning then is to find a set of parameters ψ, which gives a mapping that fits the training-set well, and is capable of generalizing well to "new" data.

The SSNN can be trained similar to a FNN by ignoring the functional dependence $x\,(t-1)$ on $x\,(t-2)$ and so on. Although both gradient (derivative of the performance function with respect to the weights) and Jacobian (derivative of model errors with respect to weights) in this case are approximations, (Elman 1990) and (Demuth & Beale 1998) show this approximation yields good results. Consequently, the most widely used technique for supervised training of FNNs, error back-propagation (Hecht-Nielsen 1990), can also be applied also to the SSNN. This algorithm is explained in detail in appendix B. We apply a fast and efficient version of backpropagation to the SSNN, which is known as the Levenberg-Marquardt (LM) Algorithm (Hagan &

[8] pertains to presenting the model with a data set of input, output data $\{\mathbf{u}(t), o(t)\}_{t=1}^{P}$ and adjusting the weight vector such that the error on this data set is minimal (e.g. in the Least Squares sense) .

Menhaj 1994). This algorithm evaluates a neural network performance function after presenting a batch of input/output data pairs to the neural network:

$$F(\psi) = \frac{1}{2}\sum_{t=1}^{P}(y(t) - o(t))^2 \tag{5.7}$$

where P denotes the total number of data pairs in the training data set. Minimizing this sum of squared error function (a mean squared error function can be used just as well), obviously maximizes the predictive performance of the SSNN on the training data set. As noted before, we do not only wish a good performance on the training data, but also a SSNN model that generalizes well to unseen data. This can be achieved by minimizing a cost function with a regularization term (MacKay 1995):

$$F(\psi) = \beta E_D + \alpha E_W = \beta \sum_{t=1}^{P}\frac{1}{2}(y(t) - o(t))^2 + \alpha \sum_{i=1}^{Q}\frac{1}{2}\psi_i^2 \tag{5.8}$$

where Q denotes the total number of weights in parameter vector ψ. The parameters β and α regulate to which extend the output error (the first term in equation 5.8) and the size of the weights (the second term) contribute to the performance function. Regularization takes into account that increasing model complexity (larger or more weights) may lead to better performance on the training data, but also to poorer generalization with respect to unseen data. The regularization parameters can be updated simultaneously with the network parameters ψ with the Levenberg-Marquardt and Bayesian Regularization (LM-BR) algorithm described by (Foresee & Hagan 1997). A more general treatment of Bayesian backpropagation is provided in (Thodberg 1996), which also provides a practical recipe at implementation level and is largely based on earlier work of Mackay (1992).

Central to the LM-BR algorithm is the notion that minimizing $F(\psi)$ is in fact equal to maximizing the posterior probability[9] of a particular weight vector given the training data D, and the regularization parameters β and α. If we assume that both the noise (the randomness) in the training data are Gaussian distributed according to $N(0, 1/\beta)$ and that the prior distribution of the weights is also Gaussian distributed according to

[9] $P(\psi|D, \alpha, \beta)$, that is, the probability of the parameters ψ after observing the data D given the regularization parameters α, β.

$N(0, 1/\alpha)$[10], we can write for the posterior distribution of the weights

$$\begin{aligned} P(\psi|D,\alpha,\beta) &= \frac{P(D|\psi,\beta)P(\psi|\alpha)}{P(D|\alpha,\beta)} \\ &= \frac{1}{P(D|\alpha,\beta)}\left(\frac{1}{Z_D(\beta)}\exp\left(-\beta E_D\right)\frac{1}{Z_W(\alpha)}\exp\left(-\alpha E_W\right)\right) \\ &= \frac{1}{P(D|\alpha,\beta)}\left(\frac{1}{(\pi/\beta)^{P/2}}\exp\left(-\beta E_D\right)\frac{1}{(\pi/\alpha)^{Q/2}}\exp\left(-\alpha E_W\right)\right) \\ &= \frac{1}{P(D|\alpha,\beta)}\left(\frac{1}{Z_F(\alpha,\beta)}\exp\left(-F(\psi)\right)\right) \end{aligned} \tag{5.9}$$

In eqn (5.9) (which is obtained by Bayes rule), $P(D|\psi,\beta)$ denotes the *likelihood* of the data occurring, given our particular setting of ψ and β. The second term in the numerator $P(\psi|\alpha)$ denotes the prior distribution of the weights, representing our prior knowledge of the parameters in our SSNN model before we have presented any data to it. The denominator is a normalization factor, which ensures that the total posterior probability density equals 1 (it is also called the evidence for D).

5.4.2 Algorithm: Levenberg-Marquardt and Bayesian regularization (LM-BR)

From eqn (5.9) it is clear that minimizing $F(\psi)$ is indeed equal to maximizing $P(\psi|D,\alpha,\beta)$. The SSNN training algorithm then becomes an empirical Bayes algorithm (MacKay 1995) and contains two consecutive steps which are executed until the performance or stopping criteria are met.

1. use Levenberg-Marquardt (LM) to minimize $F(\psi)$ given current β and α. In the LM algorithm weight updates are calculated by backpropagating the output errors $\mathbf{e}(\psi) = (e_1(\psi), ..., e_P(\psi))^T$, with $e_t(\psi) = y(t) - o(t)$ on each of the P data pairs into the SSNN. This backpropagation process produces so-called neuron sensitivities or delta's. Based on this delta's we can construct a Jacobian matrix $\mathbf{J}(\psi) = \nabla_{\psi}\mathbf{e}(\psi)$ of output errors with respect to the SSNN weights. This Jacobian is used to approximate the Hessian matrix of the performance function with respect to the SSNN weights:

$$\widehat{\mathbf{H}} = \nabla^2 F(\psi) \approx \beta\mathbf{J}^T(\psi)\mathbf{J}(\psi) + \alpha\mathbf{I} \tag{5.10}$$

   in which $\mathbf{I}$ denotes the identity matrix. With both Jacobian and Hessian, we can apply a variation of Newton's algorithm to move the ψ vector in a direction where performance increases

$$\psi^{new} = \psi^{old} + \left|\widehat{\mathbf{H}}(\psi) + \mu\mathbf{I}\right|^{-1}\mathbf{J}^T(\psi)\mathbf{e}(\psi) \tag{5.11}$$

[10] Off course, different priors for the weights are possible. Suppose a SSNN is already trained on some dataset, then the prior for the weights on a second data set would then equal the posterior obtained after training on the first data set.

where μ is a training parameter updated during training. The update rule (5.11) is approximately Gauss-Newton for small μ , and approximately steepest descent for large μ. If a step with (5.11) leads to an improvement in performance (5.8), μ is decreased and we continue with step 2. If a step with (5.11) deteriorates performance then the weight-update is discarded, μ is increased and a new weight-update is executed (with the same Jacobian) until an improvement is achieved. The algorithm stops of μ exceeds a certain threshold value.

2. update the regularization parameters by maximizing the posterior probability of α and β given the observed data:

$$P(\alpha, \beta|D) = \frac{P(D|\alpha, \beta)P(\alpha, \beta)}{P(D)} \tag{5.12}$$

which (if we assume a uniform prior distribution $P(\alpha, \beta)$ of α and β) is equal to maximizing the likelihood $P(D|\alpha, \beta)$ of the data D in the light of α and β. Note that this likelihood in fact is the normalization factor of eqn (5.9)! In (Thodberg 1996) and (Foresee & Hagan 1997) it is shown that the maximum likelihood estimates for α and β can now be expressed by

$$\alpha^{MP} = \frac{\gamma}{2E_W(\psi^{MP})}, \; and \; \beta^{MP} = \frac{P - \gamma}{2E_D(\psi^{MP})} \tag{5.13}$$

where

$$\gamma = Q - 2\alpha \cdot trace(\widehat{\mathbf{H}}^{-1}) \tag{5.14}$$

is the *effective number of parameters* in the SSNN model. Conveniently, in the LM algorithm the term $\widehat{\mathbf{H}}$ is already calculated as part of the weight update, so optimizing the regularization parameters does not increase the computational expense of the algorithm more than linearly.

Algorithm 2 *SSNN Training procedure*

1. *Initialize regularization parameters (e.g. $\beta = 1$ and $\alpha = 0$), and the SSNN weights ψ (random or with Nguyen-Widrow (Nguyen & Widrow 1990))*
2. *Take a step in the LM algorithm that minimizes performance eqn (5.8) with fixed α and β using eqn (5.11) and adjust weights ψ^{new}*
3. *Optimize α and β with eqn (5.13) given updated weights ψ^{new}*
4. *If convergence criteria met (minimum performance goal, maximum number of steps, maximum value of μ or gradient norm reached) then stop, else continue with step 2*

More details on the algorithm used can be found appendices B and C respectively, which are based largely on (MacKay 1995), (Hagan & Menhaj 1994) and (Foresee & Hagan 1997).

5.4.3 Some notes on SSNN training and regularization

First note that since the SSNN model has a logistic function at the output neuron (eqn 5.5), which has meaning only in the output domain $[0, 1]$, the targets with which the model is trained have to be (linearly) scaled to that interval. We have chosen to scale all input and output to the interval [0.1, 0.9], based on the rule-of-thumb that this leads to faster and more stable learning (Hagan & Menhaj 1994). The specific choice of scaling between 0.1 and 0.9 instead of between 0 and 1, stems from the fact that for $\phi(z) \rightarrow 1$, necessarily $z \rightarrow \infty$, where $\phi(z)$ denotes the output and hidden logistic transfer function (eqn 5.3). In words, training a neuron to produce values close to 1 for a specific input pattern would require the weighted sum z of that input pattern to be very large, yielding large weights and thus a less smooth mapping. Moreover, large weighted input values (z), slow down or even completely halt training procedures.

The second remark relates to the regularization component in the training algorithm. If we assume a Gaussian posterior for the weights $N(\psi^{MP}, \Sigma_\psi)$ and a Gaussian noise model for the targets, we can obtain error bars on the prediction of a new datum at time t^* by local linearization around the output (MacKay 1995):

$$y(t^*) = G(\mathbf{u}(t^*), \psi, H) \simeq G(\mathbf{u}(t^*), \psi^{MP}) + \mathbf{g}(\psi - \psi^{MP}) \tag{5.15}$$

in which $\mathbf{g}$ is the sensitivity of the output to the parameters, that is, the first derivative of the neural network function (eqn 5.6) with respect to its weights $\frac{dG}{d\psi}$, and ψ^{MP} denotes the maximum probable parameter vector after training. The predictive distribution then becomes a Gaussian integral with mean $G(\mathbf{u}(t), \psi^{MP})$ and variance

$$\sigma^2_{y(t^*)|\alpha,\beta} = \mathbf{g}^T \widehat{\mathbf{H}}^{-1} \mathbf{g} + \sigma_o^2 \tag{5.16}$$

in which σ_o^2 denotes the nose level in the target distribution. Note that the Hessian $\widehat{\mathbf{H}}$ (which is the inverse of the variance/covariance matrix of the parameters) is calculated during neural network training (eqn 5.10), and is hence obtained automatically. The output sensitivities can be calculated similarly to the Jacobian matrix of output errors with respect to the neural network weights during training. The only difference is that we do not feed the output error $e(\psi) = y(\psi) - o$ back into the network, but rather just the output $y(\psi)$. We will use this property extensively in chapter 7.

Finally, apart from assigning error bars on its predictions, the output sensitivities $\mathbf{g}$ and the variance/covariance matrix of the neural network weights ($A = \widehat{\mathbf{H}}^{-1}$), give us detailed information on the internal workings of the SSNN. The first, for example, allows us to rationally deduce which neurons (and hence which freeway sections) and which connections are considered most relevant for travel time prediction on this particular route. Parameters that are very sensitive in particular output intervals obviously are more relevant than very insensitive parameters. Since the SSNN is a nonlinear mapping, (linear) correlation analysis of SSNN outputs with the internal states does not provide with that information. Secondly, the off-diagonal elements of A (covariances

of the parameters) give information on for example how redundant the internal model developed during training is. This gives some information on how robust the internal model of the SSNN is, a subject we will return to in chapter 6.

5.5 Experimental Setup

For the experiments and the resulting analysis in the remainder of this chapter, we choose to use synthetic data obtained from a traffic micro-simulation tool, since it enables us (a) to control the types of traffic situations (free-flowing, congested) we wish to feed into the model and (b) to compare the SSNN result with "actual" mean travel times, obtained from the simulation. In chapter 8, we will show that the results also apply to real-time application of the SSNN model.

5.5.1 Research questions

The experiments performed in this chapter will provide us with answers to three main questions:

1. Is the SSNN model *capable of accurately predicting travel times in different traffic conditions*? The most accurate model produces unbiased results, while the variance of the output errors should be in the same order as the squared standard error of the actual mean travel times. To this end SSNN performance is compared to results of an instantaneous travel time predictor (definition on page 27), which is the current model used for VMS panels on the Dutch freeway network that display travel time. In chapter 8 performance on real data of the SSNN is compared to travel time prediction models found in literature.

2. What does the SSNN learn from the data*?*

   (a) in terms of its short term memory (the internal states)

   (b) in terms of its long term memory, that is its parameters (weights and biases)

3. Related to the previous question and with in mind that we use regularization to control the complexity of the SSNN solution, *which parameters after training are considered relevant / effective (see eqn 5.13)*, and which not?

The second and third question are particularly interesting (given the first is answered positively). The total size of the training data set (which is described below) equals 2400 records, while the dimension of parameter space is 228. It is fairly likely there exist many weight vectors in 228 dimensional parameter space that may fit these 2400 input/output patterns reasonably well (for a brief introduction on high-dimensional

parameter spaces see e.g. (Hecht-Nielsen 1990), section 2.4 "N-Dimensional Geometry"). We seek a particular parameter setting that performs well on the test data sets, and hopefully, also makes sense from a traffic engineering perspective. If the latter is true, that is if the SSNN learns to actually capture the underlying traffic processes from the data, then we may expect that the SSNN model may be applied successfully on any freeway route.

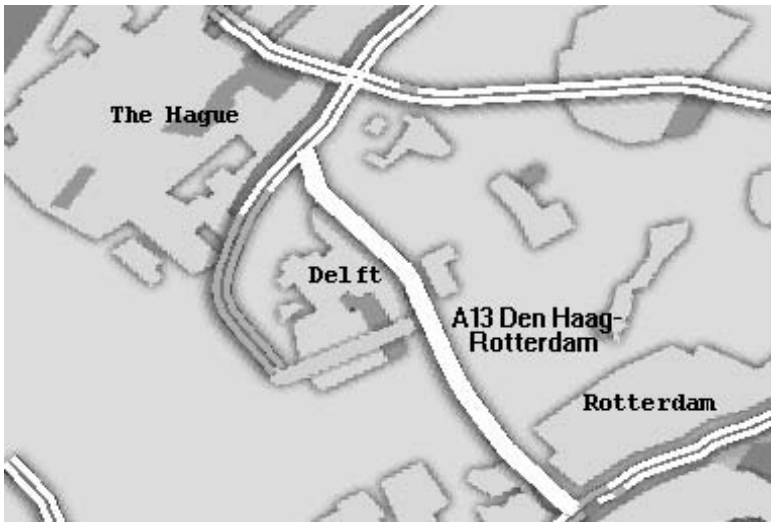


Figure 5.4: The A13 freeway between The Hague and Rotterdam.

5.5.2 Test case description

We have set up a network in FOSIM (Freeway Operations SImulation Model, see e.g. (Vermijs & Schuurman 1994)) that resembles a 8.5 kilometer stretch of the southbound carriage way of the A13 highway between two of the major Dutch cities in the western part of the Netherlands, The Hague and Rotterdam (Fig 5.4).

This three-lane stretch contains four on- and five off ramps, and three weaving sections. We have equipped the FOSIM network with a set of 13 inductive loop detectors, measuring flows and harmonic time mean speeds[11] every 60 seconds. We have divided the freeway stretch into 12 sections, each enclosed between two consecutive detectors up- and downstream. Table 5.1 overviews the 12 sections and their associated detectors, while fig. 5.5 gives a schematic drawing of the lay-out of the route.

[11] In real life, most inductive loop detectors record arithmetic time mean speeds. See chapter 3 for details.

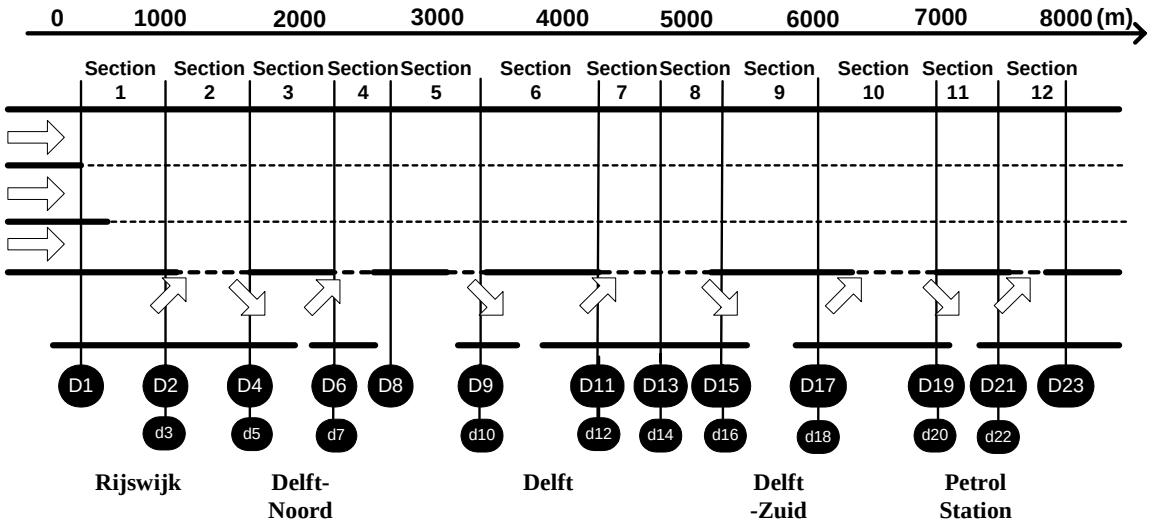


Figure 5.5: Freeway stretch coded in microscopic traffic simulation model FOSIM. The stretch resembles a 8.5 kilometre stretch of the the southbound A13 highway between The Hague and Rotterdam (The Netherlands)

As a consequence of the SSNN topology (see fig. 5.3), we now have sufficient information to set up a SSNN model for this particular freeway stretch. The hidden layer of the SSNN model contains 12 hidden neurons, each representing a section along the route, and each receiving signals from those inputs associated with that section (listed in table 5.1). The context layer also contains 12 units, while the output layer contains 1 neuron. The number of parameters (weights and biases) in the model then initially amounts to 228.

5.5.3 Input and output data

The input to FOSIM are so-called Dynamic OD matrices, which contain time varying demand patterns between each origin (main carriage way and on-ramps) and destination (main carriage way and off ramps). Based on historical data, the average percentage of trucks on this route is between 15 and 20%. We choose to scale the demand to approximately fit with real traffic data collected on the A13, however, no sophisticated OD estimators were used, since our primary interest was to collect detailed data (speeds, flows from detectors and mean travel times) to train and test our models, rather than have the simulation results fit seamless with real-time data. Nonetheless, we have set up three distinct but realistic traffic demand patterns for this particular network.

Traffic demand patterns

Pattern 1: recurrent congestion at Delft-Zuid The first pattern (fig. 5.6) pertains to normal traffic flow operations which occur practically every weekday afternoon on

Table 5.1: Overview of the sections in the FOSIM freeway stretch. Note that $\{u, q\}^{up}$, and $\{u, q\}^{down}$ denote (harmonic time mean) speed and flow from up- and downstreamdetectors respectively, q^{on} and q^{off} denote in- and outflow at on- and offramps respectively. The rightmost column depicts the detectornumbers for each section

	from (m)	to (m)	Nr. Lanes	Input*)	Dets
1	500	1120	3	$\{u, q\}^{up}$,$\{u, q\}^{down}$	1,2
2	1120	1800	3 + weaving section	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{on}, q^{off}	2-5
3	1800	2450	3	$\{u, q\}^{up}$,$\{u, q\}^{down}$	4,6
4	2450	2925	3 + onramp	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{on}	6,7,8
5	2925	3640	3 + offramp	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{off}	8-10
6	3640	4555	3	$\{u, q\}^{up}$,$\{u, q\}^{down}$	9,11
7	4555	5065	3 + weaving section	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{on}	11-14
8	5065	5540	3 + weaving section	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{off}	13-16
9	5540	6305	3	$\{u, q\}^{up}$,$\{u, q\}^{down}$	15,17
10	6305	7245	3 + weaving section	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{on}, q^{off}	17-20
11	7245	7735	3	$\{u, q\}^{up}$,$\{u, q\}^{down}$	19,21
12	7735	8255	3 + onramp	$\{u, q\}^{up}$,$\{u, q\}^{down}$, q^{on}	21-23

this freeway stretch. Congestion sets in after about 100 minutes due to a steep increase in traffic demand at the on-ramp Delft-Zuid (road section 10 in table 5.1). As traffic demand also from upstream locations increases, the resulting queue gradually spills back in the upstream direction (fig. 5.6, bottom graph), eventually blocking the next upstream off- and on-ramps (Delft) from minute 130 up to minute 250. At the same time congestion occurs at the on-ramp Delft-Noord (from minute 140) and spills back to the weaving section between off-ramp Rijswijk and on-ramp Delft-Noord, where it stays stationary until the 210^{th} minute. Between the 130^{th} up to the 180^{th} minute travel times of approximately 17 minutes occur (fig. 5.6, top graph), which is three times the mean free-flow travel time.

Pattern 2: a Sunday afternoon The second pattern (fig. 5.7) pertains to traffic flow operations which typically occur at Sunday afternoons. From 17:00 to 18:20 (minute 120-200) a short period of oversaturation occurs again downstream of on-ramp Delft-Zuid, mainly due to recreational traffic returning home for dinner. Travel times (fig. 5.7, top graph), however, only increase moderately as the queue evolves and dissolves during short intervals (fig. 5.7, bottom graph).

Pattern 3: accident on the on-ramp Delft-Zuid The third pattern (fig. 5.8) pertains to normal traffic flow operations on weekday afternoons during which at minute 140 an

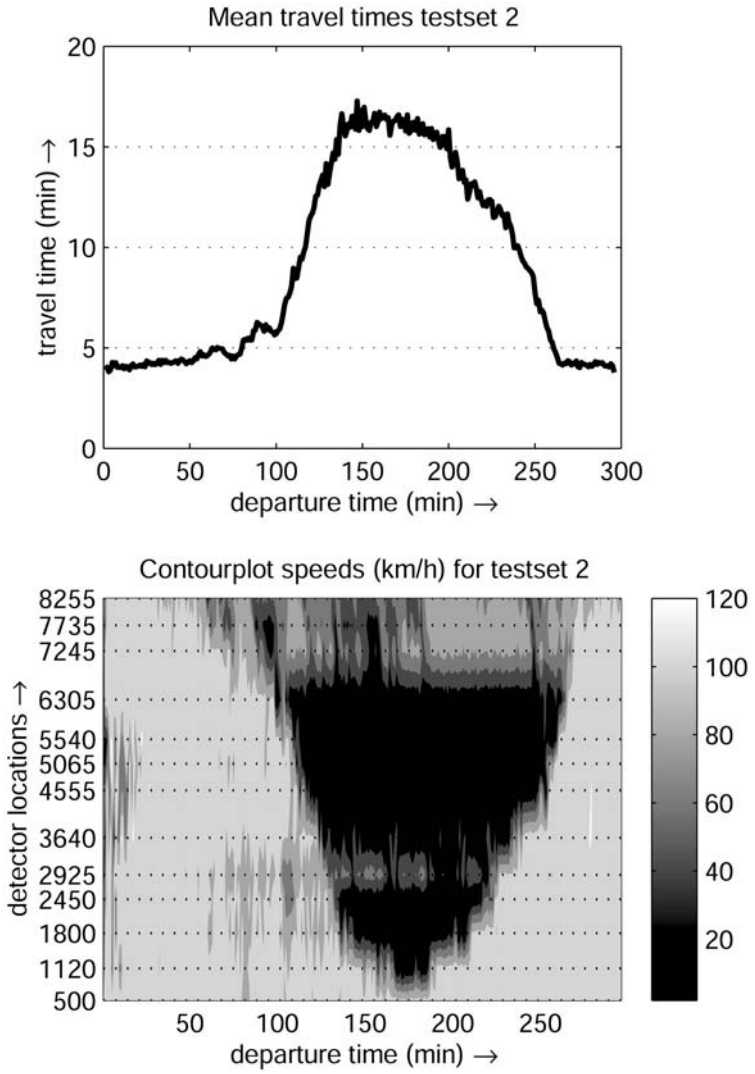


Figure 5.6: Fosim results on testset 2 (Recurrent congestion at Delft-Zuid). The figure shows both mean travel times (top graph) and a contour plot of mean speeds measured at inductive loop detectors (bottom graph).

accident occurs on the on-ramp Delft-Zuid. At this time instant, on-ramp Delft-Zuid is blocked and traffic is rerouted to either the on-ramp at Delft, the one at Delft-Noord or elsewhere. As a result, congestion on the main carriage way downstream of the on-ramp starts to dissolve directly after the accident (fig. 5.8, bottom graph). However, as traffic demand at the upstream on-ramps increases due to rerouting, this process stops as soon as the on-ramp Delft-Zuid is re-opened for traffic at minute 160. The queue rebuilds fairly quickly and spills back to on-ramp Delft-Noord. Due to the temporary blockade at the on-ramp, mean travel times are lower than normal and maximum at 14 minutes (fig. 5.8, top graph). Congestion does, however, take longer to dissolve than normal.

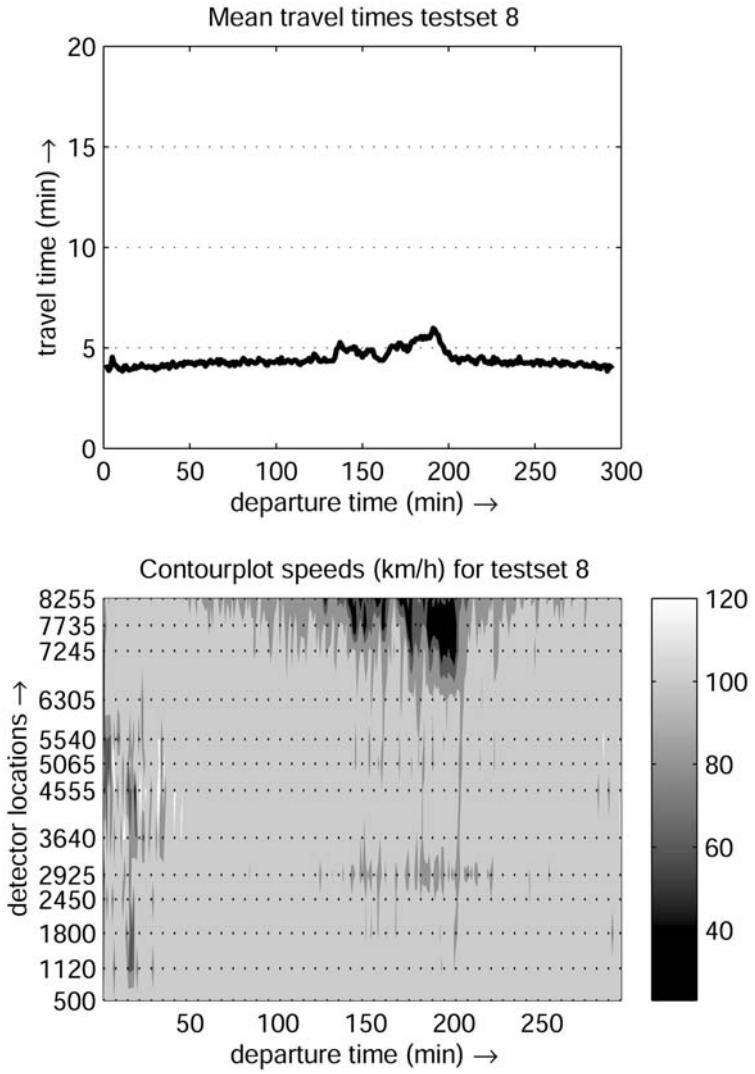


Figure 5.7: Fosim results on testset 8 (a sunday afternoon). The figure shows both mean travel times (top graph) and a contour plot of mean speeds measured at inductive loop detectors (bottom graph).

Output (travel time) patterns

For each traffic pattern presented above, we executed five 6 hour simulation runs with different random seeds, totalling in 15 data sets, each consisting of a sequence of 300 records (minutes) of inputs (mean speed and flows at detectors) and outputs (mean travel times for vehicles departing in that minute). We used the (8) odd sequences for training and the (7) even sequences for testing purposes. Different random seeds per simulation run yields different time varying demand, and also different vehicle behavior per simulation run, even if the input demand pattern is equal (fig. 5.9 gives an example for traffic pattern 1).

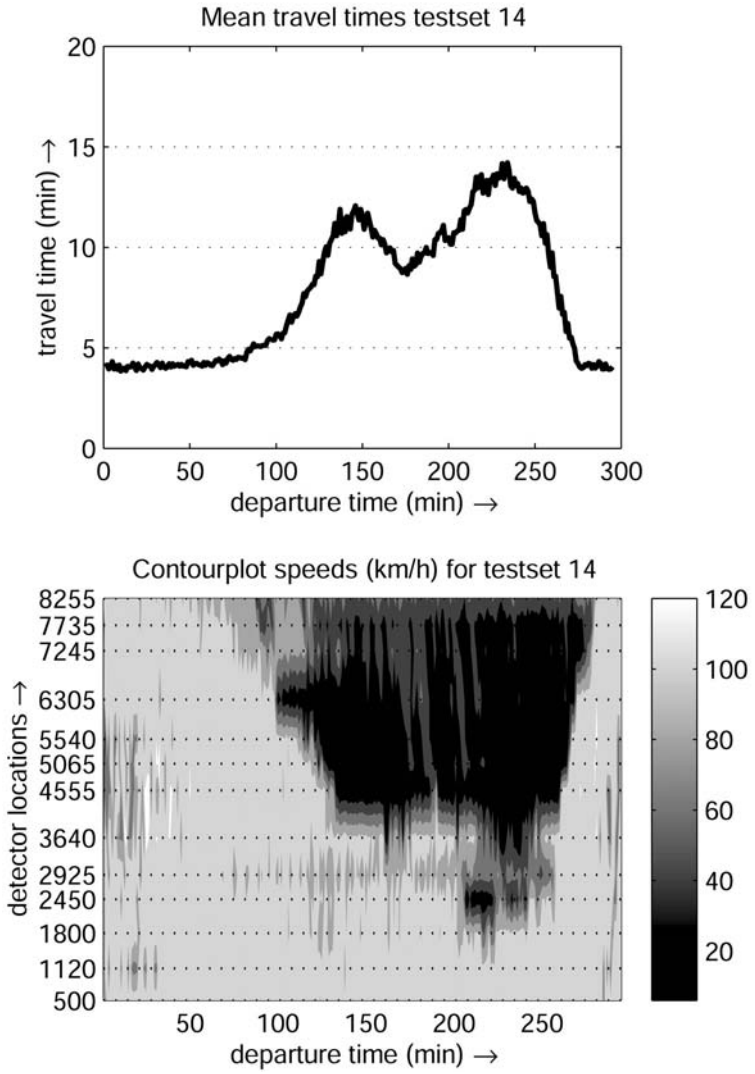


Figure 5.8: Fosim results on testset 14 (Accident at onramp Delft-Zuid). The figure shows both mean travel times (top graph) and a contour plot of mean speeds measured at inductive loop detectors (bottom graph).

5.5.4 Results of the SSNN training procedure

Fig. 5.10 shows the evolution of the sum squared error (SSE), sum of squared weights (SSW) and the effective number of parameters γ for one training trial of 200 epochs. In the first few epochs the Bayesian Levenberg-Marquardt Algorithm quickly decreases both the error on the data (SSE), the number of effective parameters γ and the SSW. The initial decrease in γ (effective parameters) is an artefact of the definition of this number (eqn 5.14). This number γ is proportional to the inverse of the Hessian (variance-covariance matrix of parameters), which at the start is large probably due to weight initialization. After a few epochs γ and SSW slowly increase while the SSE continues decreasing. We stopped training after a maximum of 200 epochs.

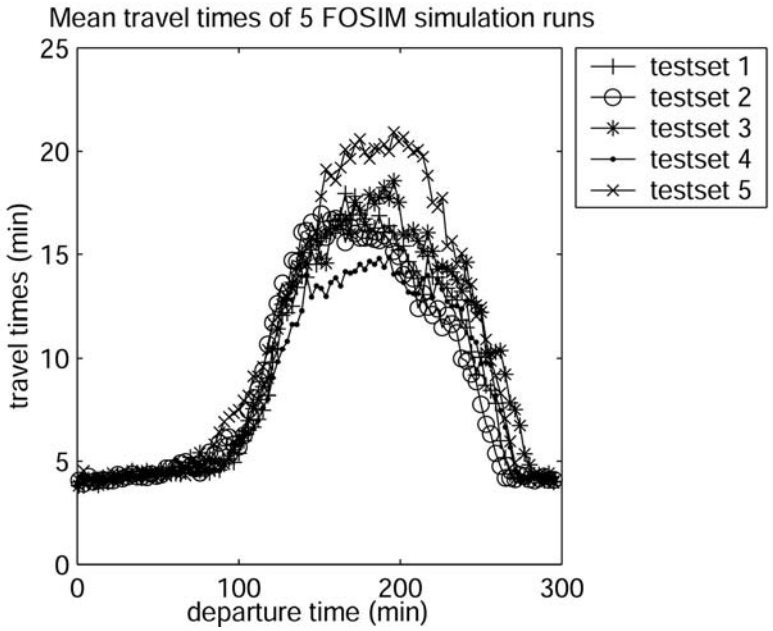


Figure 5.9: Mean travel times for 5 fosim simulation runs generated with traffic demand pattern 1 (recurrent congestion). The five runs were executed with different random seeds, yielding different simulation results.

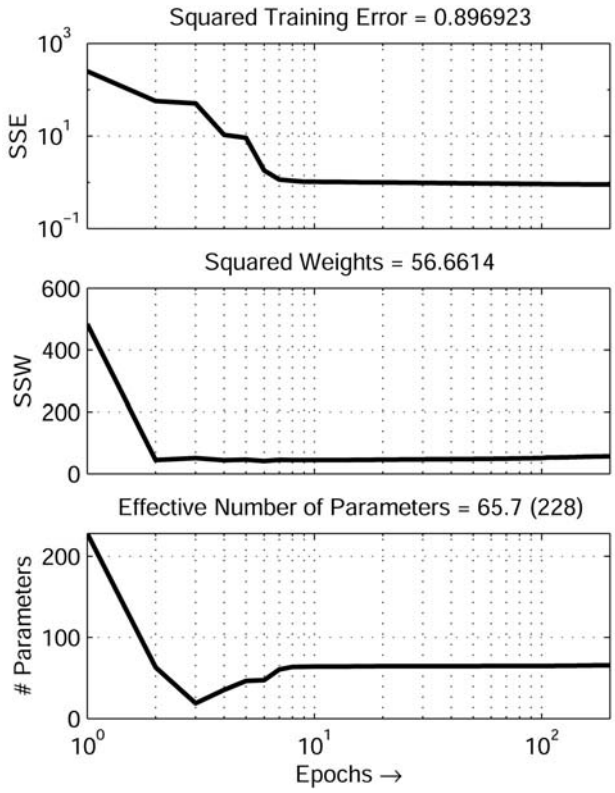


Figure 5.10: Training record of the SSNN model. It shows the Sum Squared Error (top), Sum of Squared Weights (middle) and the Number of effectvie parameters γ (bottom) evolve over 200 training cycles.

An interesting result is that from the initial 228 parameters, after training 66 are dubbed "effective" or "well-determined". Tentatively, one might translate this to "relevant"[12]. What it certainly indicates is that given our particular model settings, the maximum posterior probability of the parameters in the light of the training data yields a parameter vector in which only 66 of the 228 parameters are dubbed "well-determined", or - which is a "loose" interpretation on our part - effectively produce the desired output. Interpreted this way, this yields a reduction in model-complexity of more than 70%!

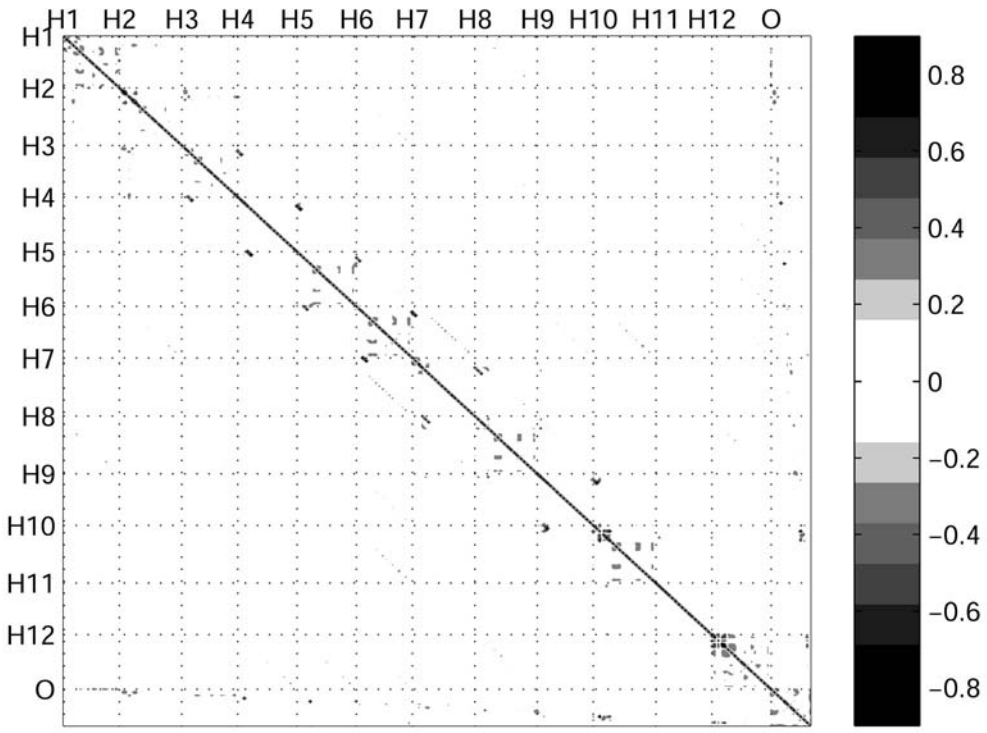


Figure 5.11: Cross-correlation matrix of the SSNN parameters. Dark regions in the plot denote high (positive or negative) correlation. The label H*m* denotes parameters associated with hidden neuron m, while O denotes parameters associated with the output neuron.

$$\Theta_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}, \mathbf{C} = \mathbf{H}^{-1} \text{ (Variance / Covariance matrix of the SSNN parameters)} \quad (5.17)$$

A close look at the cross-correlation matrix (Fig 5.11) with elements Θ_{ij} (eqn 5.17) indicates that the SSNN model has developed a mapping with little redundancy. Strong correlation only exists between parameters of "neighboring" hidden neurons (little dark

[12]We avoid the term "significant", since stricly speaking, we can not calculate the (statistical) significance of the SSNN parameters due to the fact that the SSNN model output relates nonlinearly to its parameters

areas lower/upper of the main diagonal), almost no correlation exists between parameters of "distant" neurons. This makes sense since neighboring neurons often share input signals from the same detectors. The last row (or column) shows the cross correlation of output weights with the hidden neuron weights.

The second (and expected) observation is that five different training runs, with different weight initialization, produce approximately the same number of effective parameters, and moreover, produce approximately the same variance/covariance matrix for the weights. We hypothesize this is due to two phenomena. The first relates to the Levenberg-Marquardt Algorithm, which (for small values of μ - see eqn 5.11) effectively is a Gauss-Newton algorithm, which is robust to local gradient information. It determines its decent not only on local gradients, but also on the curvature (second derivative or Hessian) of the performance function. Since the weight initialization is not entirely random but optimized based on (Nguyen & Widrow 1990), the initial parameter vectors are predominantly located in roughly the same (albeit 228 dimensional) region in parameter space. Despite slightly different starting points, the algorithm (especially the first few steps) consequently steers the parameter vector in the same direction. The second argument is more tentative. There appears to be one particular region in parameter space where the (posterior) probability density of the parameters, given the data and our model, is highest, or - which is the same - where the performance function is lowest. One might think of this region as a deep sea trough in a vast ocean of shallow water. From a traffic flow theory point-of-view, this makes sense, since there exist very specific paths (characteristics - see for example (Helbing 1997)) through space and time along which traffic state information (e.g. mean speeds) travels.

Note that under a different prior weight distribution (for example one yielding very large initial weights), the LM-BR algorithm might nonetheless lead to a different weight-setting. However, keeping initial weights small (around zero) has been a well-established strategy in neural network training that leads to significantly better performing and generalizing models (MacKay 1995), (Nguyen & Widrow 1990), (Thodberg 1991).

5.6 Predictive Performance of the SSNN

Table 5.2 shows the predictive performance in terms of the root mean squared error proportional (RMSEP), mean relative error (MRE) and standard deviation of the relative error (SRE - formulae are given in appendix A) on all of the 7 test data sets obtained from the simulation. As expected the dynamic SSNN outperforms the static instantaneous travel time predictor (see chapter 2 and 3). It predicts almost unbiased travel times (less than 0.5% MRE) with a SRE of 6%. This, in fact, is smaller than the standard deviation of actual travel times obtained from the simulation (the targets), which over *all* (7) test data sets is about 8.7%.

Table 5.2: Predictive performance of SSNN model and instantaneous travel time predictor.

	RMSEP (%)	MRE (%)	SRE (%)
SSNN	7.7	0.49	6.0
Inst. Travel Time	17	3.7	13.3

Figs 5.12, 5.13, and 5.14 show the predictive results for three test data sets, each drawn from the three different traffic patterns described above. As expected, the instantaneous travel time predictor is outperformed by the SSNN predictor in dynamic traffic conditions (patterns 1 and 3), but produces good results for free-flow or near free-flow traffic conditions (pattern 2). Also note that the instantaneous travel time predictor predominantly overestimates travel times, as it fails to track dissolving queues. The SSNN model fairly accurately predicts queue build up and dissolve, but nonetheless produces (as can be expected) the largest errors in congested conditions. In conclusion, the SSNN model proves an accurate predictor of travel times in both free-flow and congested conditions, producing on average approximately zero mean residuals with a more than acceptable 6% standard error.

In sum, the SSNN model outperforms the current models, since these instantaneous predictors are based on the assumption of stationary conditions, which does not hold in congested conditions. In chapter 8 the predictive performance of the SSNN *on real data* is compared against the some freeway travel time prediction models found in literature.

5.7 Analysis of the Internal Workings of the SSNN

An important note on beforehand relates to the initial setting of the short term memory of the SSNN (the context layer). We have chosen an initial activation of this context layer (in fact the value of the internal states $x(t = 0)$) that matches the internal states of the SSNN in free-flow conditions. Setting these context neuron activations to arbitrary initial values (e.g. zeros) causes the SSNN to require some time (generally only 2 to 5 time steps) to stabilize. Below we will now analyze in detail what the SSNN model has learned from the data in terms of its internal states, and its short and long term memory, that is, its context layer and its parameters. We start by analyzing the dynamics of the internal states of the model.

5.7.1 Correlation between internal states and traffic conditions

Recalling eqns (5.4) and (5.5), the weighted internal states (weighted with the output parameters $\mathbf{w}$) of the SSNN can be interpreted as the contribution to the SSNN output,

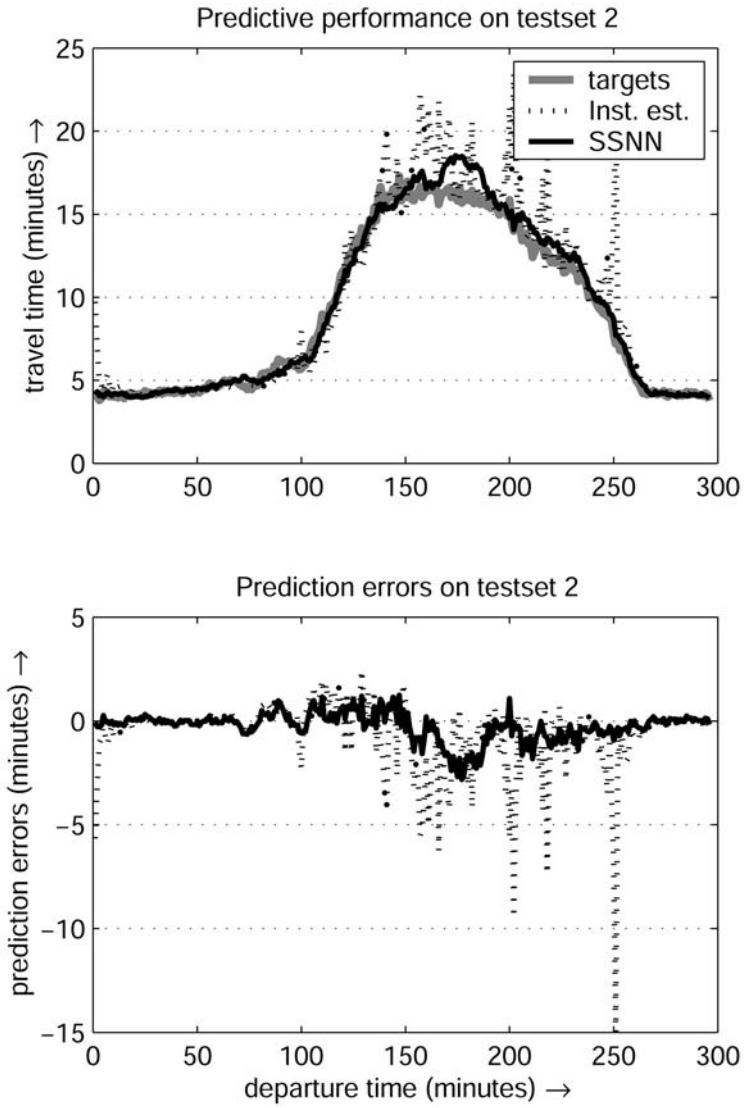


Figure 5.12: Predictive performance of SSNN predictor on test dataset 2 (traffic pattern 1: normal weekday congestion).

that is, the predicted mean travel time on the route of interest. Fig 5.15 shows the evolution of these weighted internal states on test data set 2 (traffic pattern 1). Note that the constant (parts of the) signals are not relevant!

We can observe from Fig 5.15 that obviously not all hidden neurons are active for during the entire 5-hour period. In fact, some neurons only contribute a constant value (4, 9 and 11) regardless of traffic conditions, while other neurons (e.g. 6,7,8, and 10) produce activities which seem very similar to the SSNN output. Qualitatively, this makes perfect sense, since the most active neurons "represent" those freeway sections on which the propagation of congestion (causing delays) is determined (see table 5.1). For example, neuron 10 represents the section downstream of on-ramp Delft-Zuid, which can be classified as the initial bottleneck on which the head of the queue stays stationary for most of the simulation period (see Fig 5.6), while neuron 6 represents

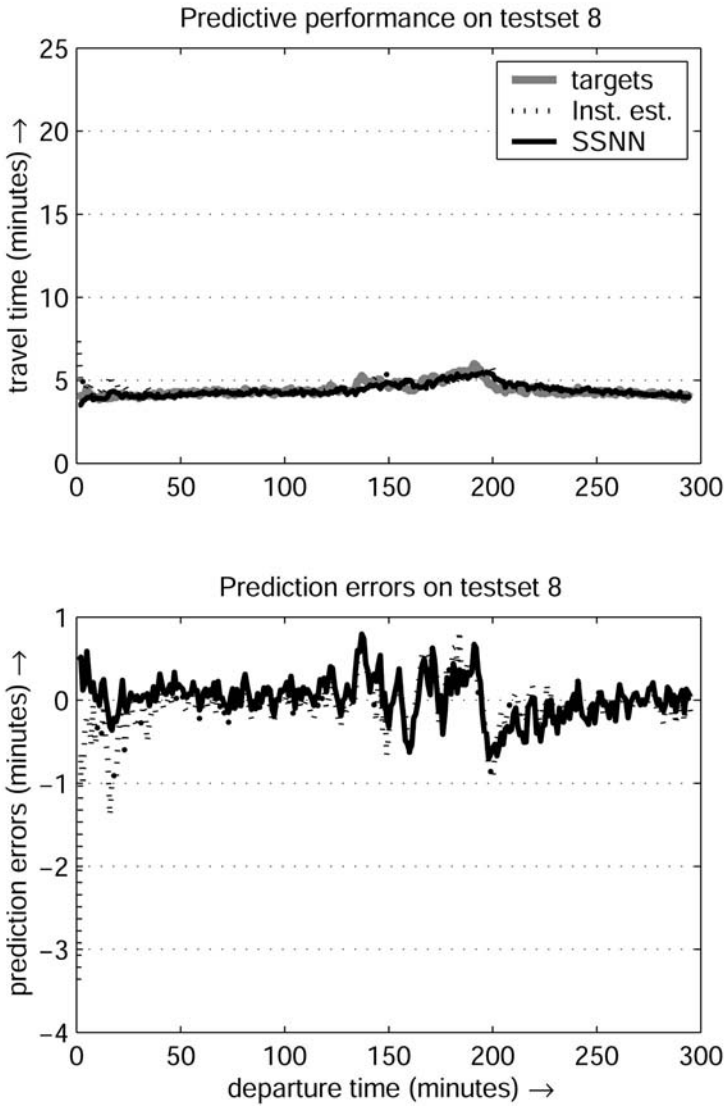


Figure 5.13: Predictive performance of SSNN predictor on test dataset 8 (traffic pattern 2: Sunday afternoon).

the 1.2 km section between on-ramp Delft-Noord and off-ramp Delft on which during this simulation run the highest densities occur. Fig 5.16 (top) shows a contour-plot of the internal states weighed with their output weight for test data set 2. These weighted internal states can be interpreted as the contribution of each hidden neuron to the output. For readability the minimum value of each time series is removed, such that every weighted internal state values in the range $[0, \infty]$. Fig 5.16 (bottom) shows a contour-plot of densities (veh/km) obtained from the simulation. Obviously, areas of high neuron activity (Fig 5.16 top) coincide with regions where densities are high (Fig 5.16 bottom). From Fig 5.16 (top) we can conclude the SSNN model assigns most predictive value to the state of neuron 6, on which associated section (no 6) the tail of the queue is located for most of the congested period.

Fig 5.17 shows a cross-correlation contour-plot of the weighted internal states and the

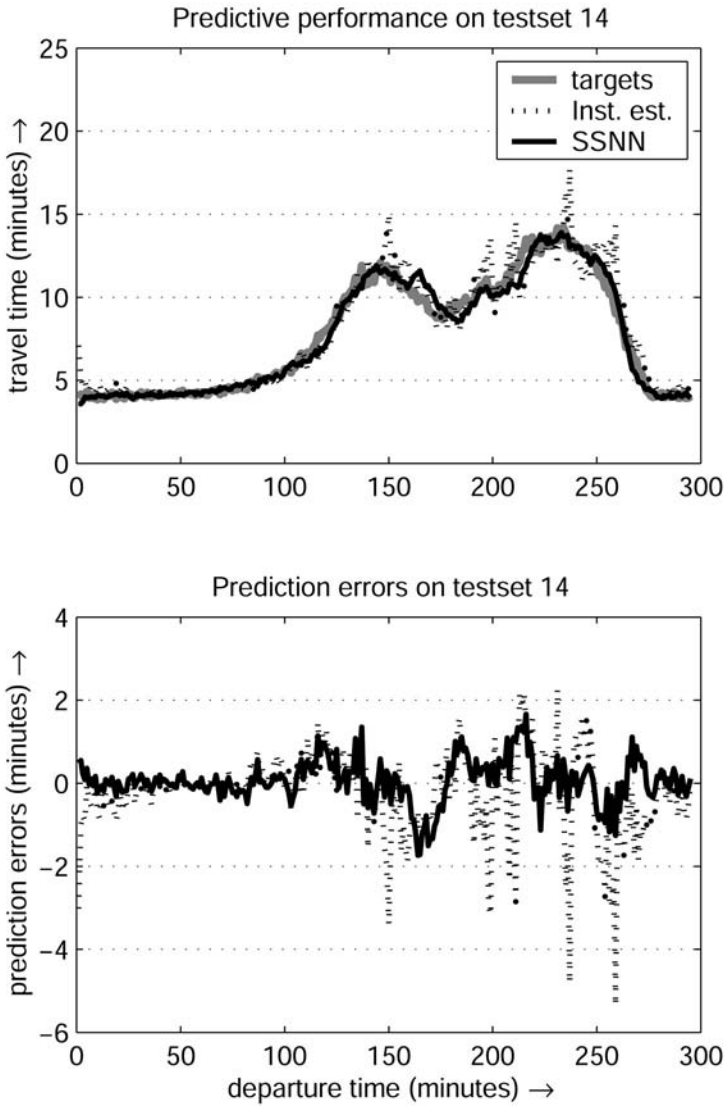


Figure 5.14: Predictive performance of SSNN predictor on test dataset 14 (traffic pattern 3: accident at on ramp Delft-Zuid).

section densities. For readability, the bias of all weighed internal states have been removed. In general, the weighed internal states are most strongly correlated to densities on their associated sections. Nonetheless, they are also highly correlated to densities of "neighboring" sections. For example internal state 6 is correlated to density on sections 5 up to 9. This makes sense from both a neural network as well as a traffic engineering perspective. In the first place neighboring neurons share input signals from the same detectors and secondly, traffic conditions among neighboring sections are strongly interrelated due to the dynamics of traffic processes.

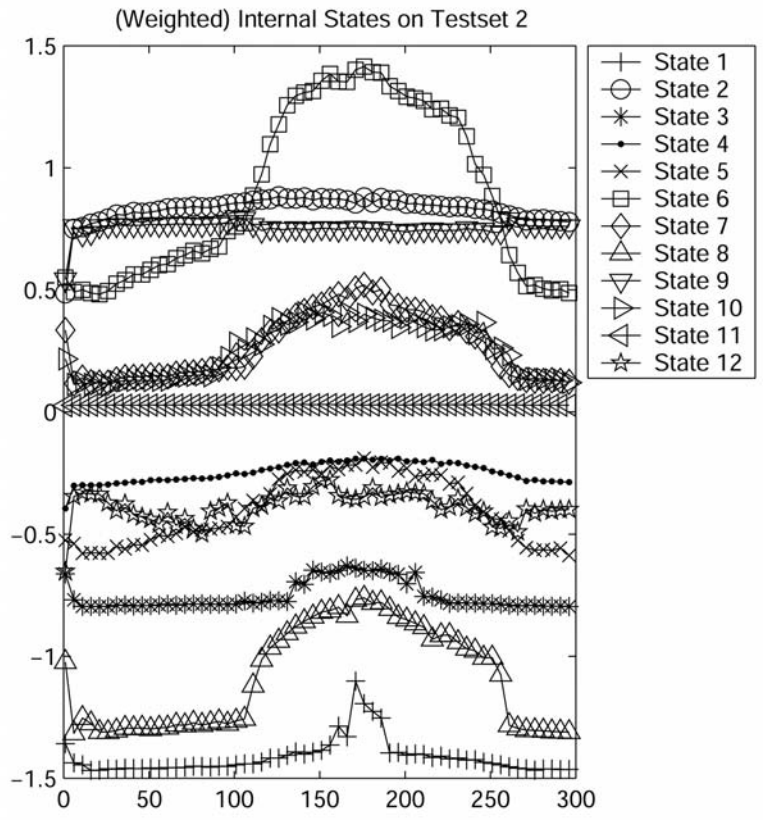


Figure 5.15: Evolution of the (weighted) internal states on test data set 2 (traffic pattern 10). Note that constant states do not contribute to the SSNN output!

5.7.2 Relevance of individual neurons and inputs

From Fig 5.17 we can observe that all weighted internal states are correlated to the SSNN output. However, correlation in itself gives very limited information. Most importantly, correlation does not imply causality ($A \rightarrow B$), since it may very well be that A and B are completely unrelated due to some unobserved variable C causing $C \rightarrow A$ and $C \rightarrow B$.

Relevance of the internal states

Therefore we propose an expression for the *relevance* (sensitivity) of each internal state with respect to the SSNN output. For ANN regression problems such as travel time prediction, various approaches to (neuron or input) relevance detection have been used such as iterative re-training procedures (e.g. (Setiono & Liu 1997), (Setiono & Gaweda 2000)) and (applied to RNNs such as the SSNN) dual extended Kalman Filtering (Leung Chi & Chan Lai 2003). The first requires many retraining sessions for stable relevance indicators. The latter is appealing since it determines relevance online, however, this algorithm does not straightforwardly combine with the (offline) LM-BR algorithm we use. We propose a relevance measure that uses first derivatives of the

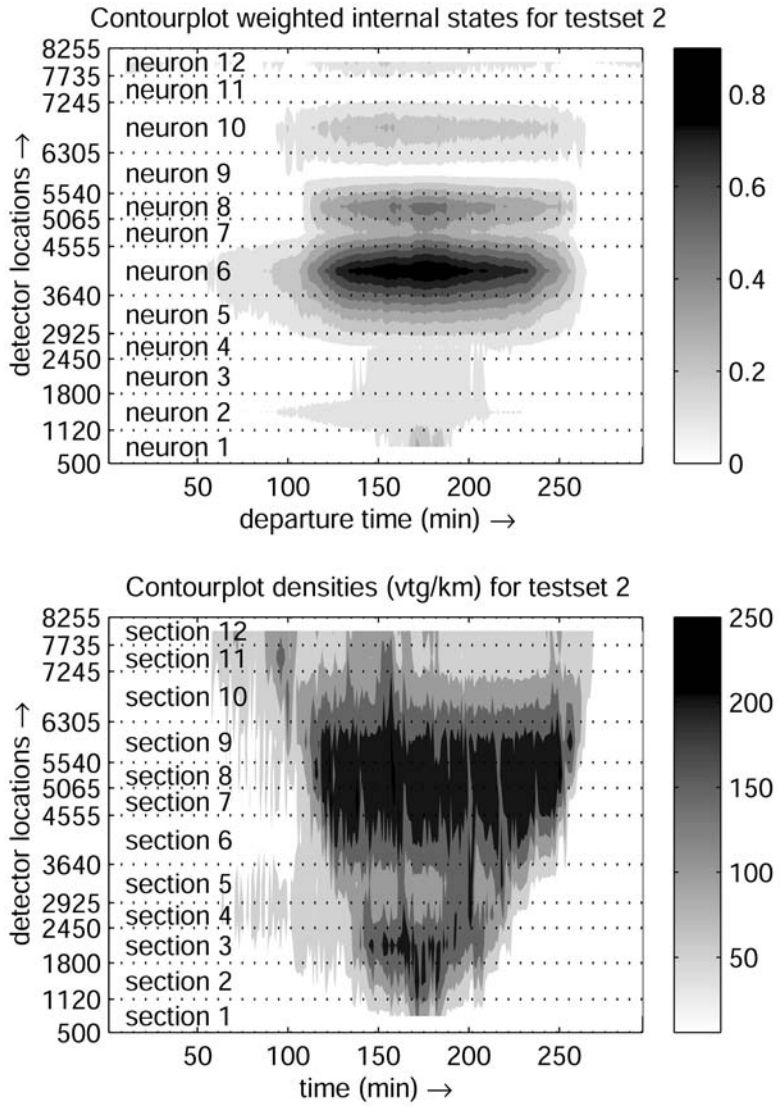


Figure 5.16: Contourplot of the (weighted) internal states on test dataset 2. Dark areas denote high neuron activity.

neural network output with respect to its weights, similar to the weight update procedure in the backpropagation training algorithm. This leads to so-called neuron sensitivities or deltas (for a details see section B.3.1 in appendix B). If the sensitivity for say a particular hidden neuron m_1 is higher than for another neuron m_2, then changes in internal state m_1 invoke potentially larger changes in the SSNN output than changes in m_2. The magnitude of the internal states subsequently governs the absolute effect of these sensitivities. We can use the same recipe for calculating neuron relevance as for calculating deltas during training, except that we do not feed back output errors into the SSNN, but rather just outputs. Consequently, calculating each neurons' sensitivity is analogous to performing one backward pass in the backpropagation learning algorithm. Based on eqns (5.4) and (5.5) and some mathematics analogously to eqn (B.24), (B.25), and (B.26) this yields

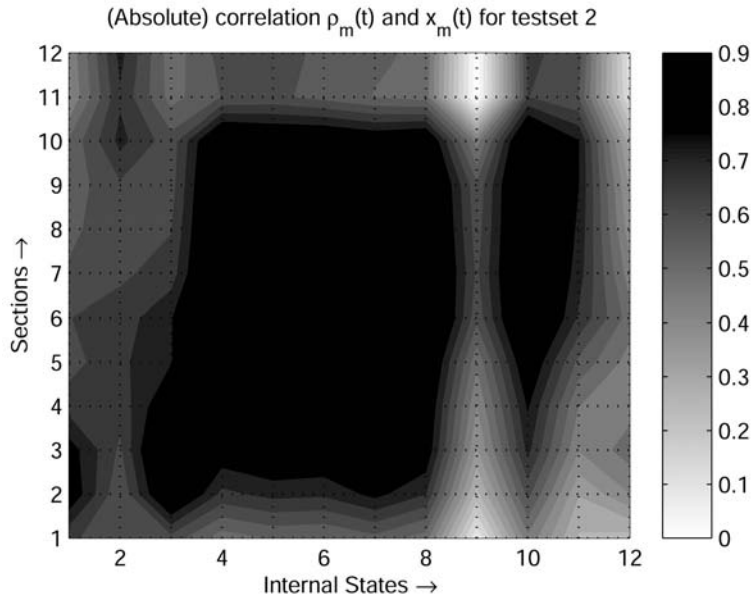


Figure 5.17: Absolute correlation of section densities (vtg/km) and the SSNN internal states. All internal states are moderately to highly correlated to the section densities, except for neuron 9.

$$\begin{aligned} s_m(t) &= \phi'(w_0 + \mathbf{w}\mathbf{x}(t))\mathbf{w}\phi'(I_m(t)), \\ with\ \ I_m(t) &= v_0 + v_{u,m}\mathbf{u}_m(t) + \mathbf{v}_{x,m}\mathbf{x}(t-1) \end{aligned}$$

where ϕ' denotes the derivative of the logistic transfer function (eqn 5.3) with respect to its inputs. Over a series of P input, output data patterns we can then express the relevance of a hidden neuron m (producing internal state m) as the sum of squares of the internal states times the sensitivities $s_m(t)$

$$\begin{aligned} S_m &= \sum_{t=1}^{P} S_m(t)^2 \\ with\ S_m(t) &= s_m(t)\left(x_m(t) - x_m^{freeflow}\right) \end{aligned} \tag{5.18}$$

Note that we subtract the free-flow activity of each hidden neuron's activity. This ensures that the relevance calculated by (5.18) is only large if the product of a neurons sensitivity and its effective contribution (in terms of delay) to the output is large. Hidden neurons that are active during the whole simulation, but only produce a constant value regardless of traffic conditions are therefore considered irrelevant.

Fig 5.18 shows a bar chart of both the (absolute) correlations of the internal states with the SSNN output and the relevance expressed by the heuristic of eqn (5.18). Although all internal states are moderately to highly correlated to the SSNN output (top graph), their relevance (bottom graph) differs largely. In fact, neurons 2, 9 and 11 have no

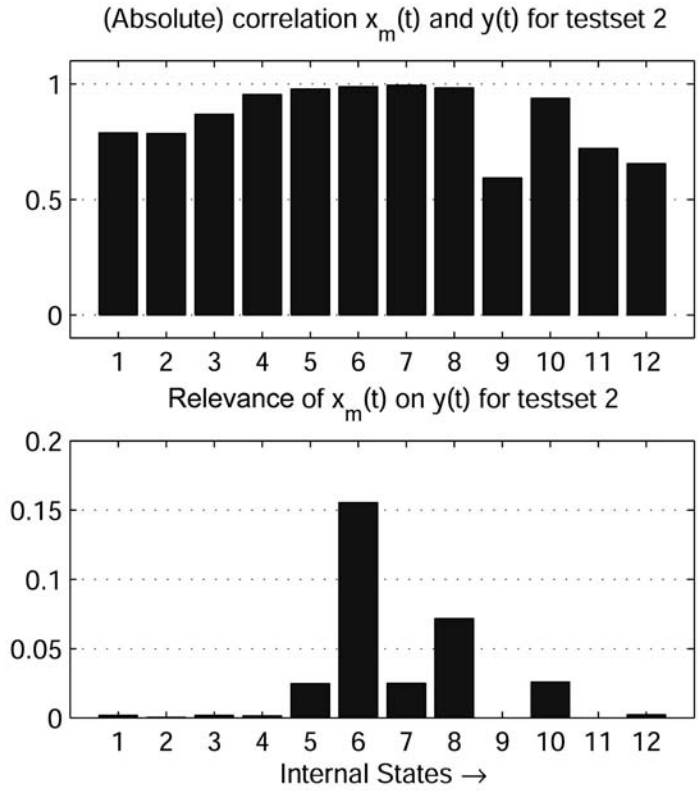


Figure 5.18: Relevance versus (absolute) correlation of the SSNN internal states to the SSNN output.

relevance at all, while neurons 5 to 8 and 10 clearly contribute most to the SSNN travel time prediction (for test data set 2). This makes perfectly sense, since congestion concentrates on the sections with which these relevant neurons are associated. Hence the relevance measure - at least for this problem - produces results that comply with expectations. Fig 5.19 compares the relevance of each internal state on two different test sets, each reflecting one of two different traffic patterns with which we have trained the SSNN, that is, traffic patterns 1 (normal weekday afternoon peak) and 3 (accident on on-ramp Delft-Zuid). For both patterns the same pattern of relevance emerges. For pattern 2 (a Sunday afternoon), which is not shown here, very low relevance is produced (order 10^{-4} times the other patterns), due to the fact that in pattern 2 almost no congestion occurs, resulting in only marginal neuron activities, while during the other two patterns heavy congestion occurs.

Using relevance to analyze memory depth of the SSNN

Using our relevance measure (5.18) we can also investigate the memory depth of the SSNN. For that, we calculate the sample auto correlation function (SACF) of $S_m(t)$, which is the correlation between $S_m(t)$ and $S_m(t-k)$, with $k = \{1, 2, ...\}$(see also eqn 3.35 on page 62). Fig 5.20 shows SACF(k) against k for hidden neuron 6 (top

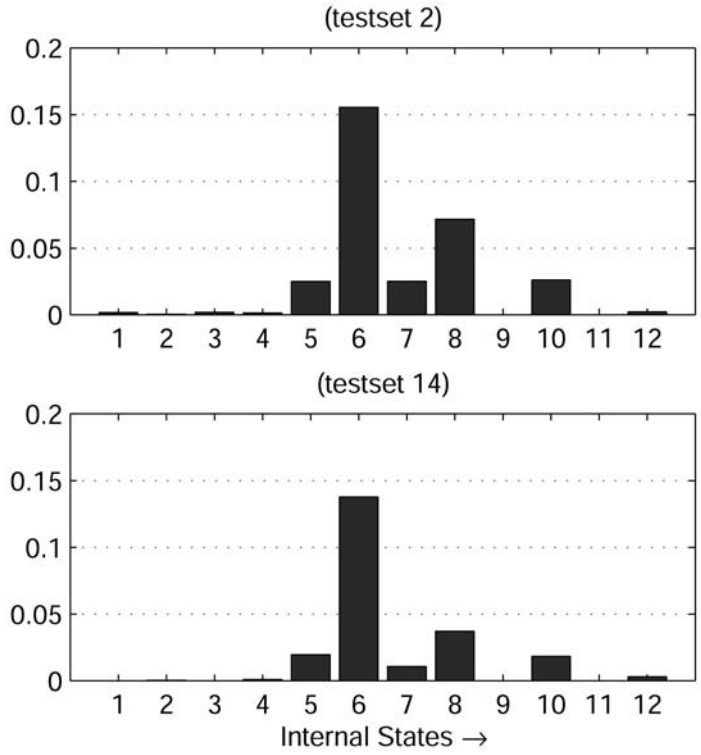


Figure 5.19: Relevance of the internal states (hidden neurons' activities) on SSNN output for two testsets reflecting two different traffic patterns.

graph) and neuron 8 (bottom). In both graphs the dotted lines depict the approximated significance of the autocorrelation found, approximated with $1/(P-k)$. For both neurons a steady decay of SACF is visible. However, in both cases, SACF(k) decays slowly for increasing k, with still a significant amount of autocorrelation left after 30 to 40 time lags. In words, the relevance of an internal state at up to 40 time steps ago, still correlates positively to current relevance. This provides some evidence that past states (up to 40 time steps ago) still influence current predictions.

Relevance of context neurons and individual input signals

In a similar fashion as with the internal state relevance we can also calculate the relevance of each context neuron and each individual input with respect to the SSNN output. The context layer has no transfer function, but rather stores the previous hidden layer activities. This is equivalent to a unity transfer function, which derivative is constant (1) with respect to its inputs $x_m(t-1)$. Let $\mathbf{s}(t) = [..., s_m(t), ...]$, $m = \{1, ..., M\}$ denote the vector of hidden neuron relevance at time step t, and $\nu^T_{x,m}$ the hidden layers' weights associated with context neurons. The context layer sensitivity then reads

$$s^c_m(t) = \nu^T_{x,m}\mathbf{s}(t)$$

while for each context neurons' relevance we can write

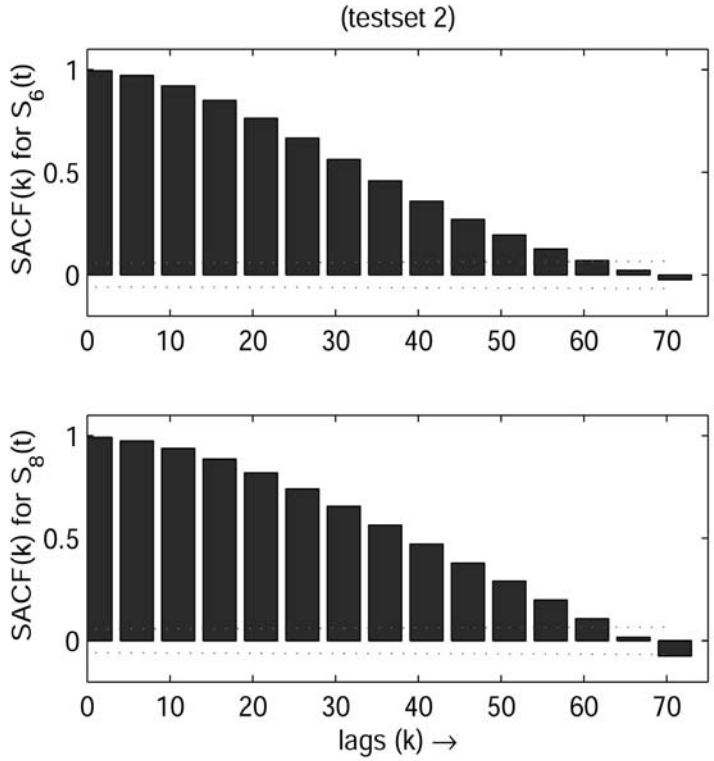


Figure 5.20: Sample Correlation between predicted delay and past sensitivities of hidden neurons 6 (top) and 8 (bottom). Both graphs show this correlation decays for increasing time lags. The dotted lines approximate the significance of this correlation.

$$\begin{aligned} S_m^C &= \sum_{t=2}^{P} \left(S_m^C(t)\right)^2 \\ with\ S_m^C(t) &= s_m^c(t)\left(x_m(t-1) - x_m^{freeflow}\right) \end{aligned} \tag{5.19}$$

At time $t = 1$, we assume the internal states equal the free flow states $x_m^{freeflow}$. Inherently eqn (5.19) yields approximately the same pattern of context relevance as internal state relevance. fig. 5.21 confirms this, showing the relevance for each context neuron on the same two data sets as in Fig 5.19. The implication is that relevant neurons also have very relevant feedback connections.

Finally, we can also calculate the relevance of each detector input datum on the SSNN output. In some ANN textbooks (e.g. (Hecht-Nielsen 1990)) each input datum is represented by an input neuron, which does nothing else but distribute the input signal to the next (hidden) layer. As such the input vector can be viewed as an input layer with unity transfer functions similar to the context layer. Since again we which to express relevance in terms of the contribution of each input datum to extra delay (travel time) in congested conditions, we subtract the free-flow detector signals from the input vector.

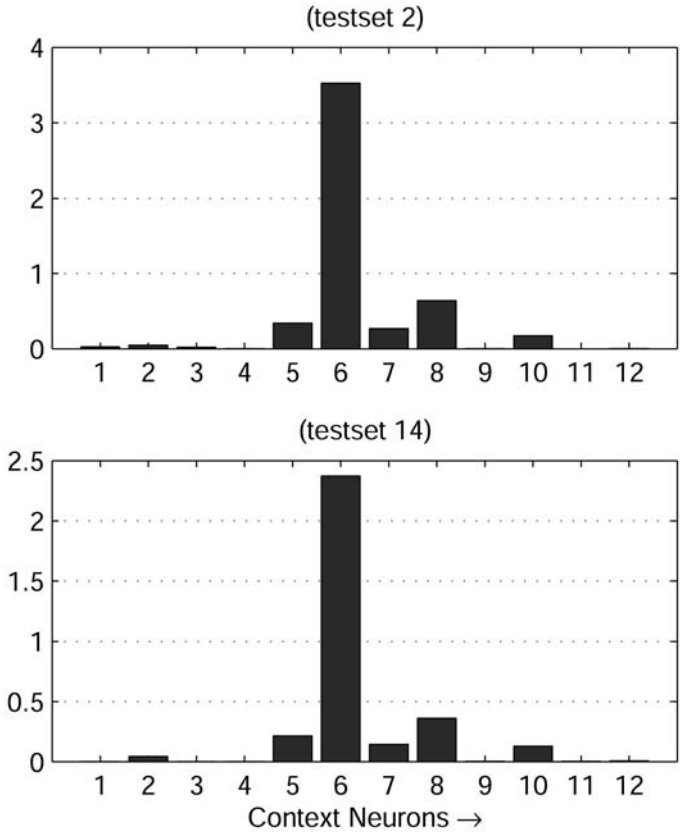


Figure 5.21: Relevance of context neurons to internal states for two testseries reflecting two different traffic patterns.

Let $v_{u,n}^T$ denote the hidden layers' weights associated with the input layer. The input layer sensitivity then reads

$$s_n^u(t) = v_{u,n}^T \mathbf{s}(t)$$

As a consequence, for each single detector input datum we get the following expression for relevance

$$\begin{aligned} S_n^U &= \sum_{t=1}^{P} \left(S_n^U(t)\right)^2 \\ with\ S_n^U(t) &= s_n^u(t)\left(u_n(t) - u_n^{freeflow}\right) \end{aligned} \tag{5.20}$$

Recall table 5.1 and fig. 5.5 earlier in this chapter schematically outlined the freeway stretch, its sections and the detector numbers coded in FOSIM. Fig. 5.22 shows the relevance of each input datum, ordered by their detector number (on the horizontal axis) for the same two test data sets use above. Note that mainstream detectors produce speed and flows, ramp detectors only flows.

The first observation from fig. 5.22 is that from most mainstream detectors predominantly the speed measurements are most relevant, except on the most downstream

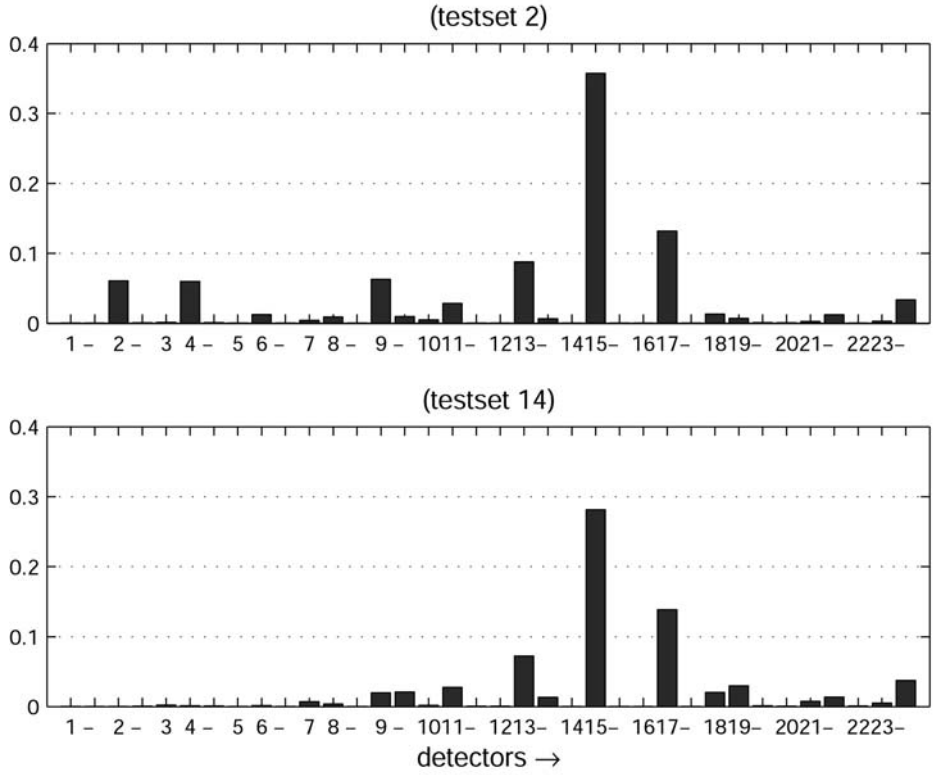


Figure 5.22: Relevance of each detector input datum to the SSNN output. Note that mainstream detectors produce two inputs (speed, flow), while ramp detectors only flow.

sections, and particularly the last, where flows are considered more relevant. The three most relevant inputs are in fact speeds on detector 13, 15 and 17 (which are used by the most relevant sections 6, 8 and 10 respectively). The logic behind it is that the SSNN model monitors mean speeds on the sections most relevant and the outflow at the very downstream end. As either one drops it implies that congestion has set in between section 5 and 12 and travel times will inherently increase. Taking a more detailed look at for example in test data set 2, density increases on sections 5 up to 10 causing a steady increase in the activity of predominantly neuron 6. As traffic first breaks down on section 10, due to high demand at on-ramp Delft-Zuid, a slight increase of the activity in neuron 10 triggers a steep increase in the activities (three feedback connections from the context layer) of neurons 6, 7 and 8 representing those sections where congestion subsequently sets in. Although high traffic demand at Delft-Zuid is ultimately the root of the problem, that particular input signal (detector 20) is barely used by the SSNN. Rather, it infers traffic break on section 10 by looking at speeds upstream (detector 17) and flows downstream (detectors 21 and 23).

5.7.3 Reducing the SSNN model

Based on the order of relevance of the hidden neurons calculated in the previous subsection, this section shows how the SSNN model can be simplified, by eliminating the irrelevant neurons. As said, irrelevant neurons only contribute to the total bias (a constant value regardless of traffic conditions).

There are several advantages and disadvantages of reducing (pruning the weights of) the SSNN. First, there are two reasons why such a model pruning procedure may be beneficial, even for models trained with regularization. One is that a smaller neural network (still warranted by the data!) yields better generalization capabilities than a larger one (Setiono & Liu 1997). The other relates to computational expense. The fewer the number of parameters (weights) the less costly the matrix multiplications (eqns 5.5 and 5.4) involved in using the SSNN are. Secondly, there are also arguments that favor use of a fully connected (regularized) SSNN model. The most important is that a fully connected SSNN model is much easier to interpret and analyze (e.g. as in fig. 5.16). The internal workings of the SSNN give insight in which sections contribute most to delays and which not. Moreover, as will be shown below, irrelevant neurons have a very marginal effect on the SSNN output at least in our test data set. We therefore hypothesize that generalization capabilities are also not more than marginally affected by the presence of these irrelevant neurons. Finally, in the light of current PC processor and memory capabilities, the benefit in computation time of a reduced model will also be marginal (in the order of thousands of seconds). Nonetheless, the elimination of irrelevant neurons allows us to show the usefulness of the relevance heuristic described above.

Fig 5.23 shows the predictive performance of the SSNN on test data set 2 as neurons in order of their "irrelevance" are removed from the model. We do this by setting its respective inputs to 0.5, representing a "mean" signal. The results of in total 13 SSNN models are shown in the graph, ranging from one including all hidden neurons to one in which no hidden neurons are active at all. It appears that removal of neurons 11, 9, 2 and 4 does not affect the SSNN performance more than marginally. By proper setting of model bias, and model retraining, the SSNN could easily do without these 4 neurons. Elimination of neurons of higher relevance then subsequently changes the output more and more. Ultimately, as all neurons have been removed the SSNN model produces a constant value regardless of its inputs, which in fact equals the mean travel time over the entire training data set, which equals 580 seconds. Even as more than 70% of the neurons is removed, the SSNN model still predicts onset and dissolve of congestion fairly accurately. As such, the model is intrinsically robust to "failing" input-signals from (the least relevant) detectors, given that these signals are replaced by a proper *null* value (0.5 in this case). In the next chapter we return to this issue in more depth and discuss ways of increasing robustness, also as input-signals to relevant neurons are failing.

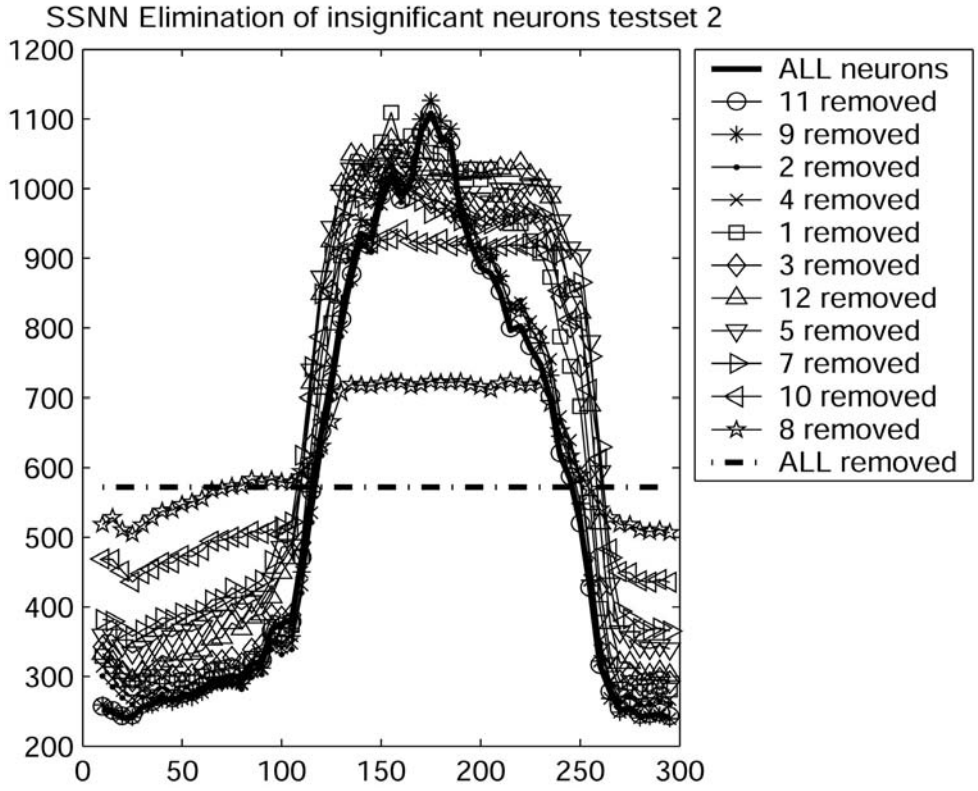


Figure 5.23: Predictive performance of SSNN after successive elimimination of neurons in ascending order of relevance

5.8 Discussion on Hidden Neuron and Input Relevance

As explained in the previous sections, the SSNN model is a Spatiotemporal Neural Network, and more specifically a First Order Context Memory (FOC), which differs from feed-forward neural networks in that it incorporates a short term memory. That short term memory does nothing but store previous values of the internal states.

As shown above, the internal states are strongly correlated to the traffic conditions on the route of interest. This implies that the SSNN uses its short term memory to keep track of previous traffic conditions and interpret the latest input data from detectors in the context of previous traffic conditions on the route of interest. In fact, information in this context memory of more than 30 time steps ago still correlates to the SSNN output, indicating a significant memory depth of the SSNN. As discussed in previous section a recurrent neural network (such as the SSNN) processes temporal dependencies of input and output in an MA (Moving Average) type of fashion, in contrast to feed-forward (FNN) or time-delayed neural networks (TDNN) which use an input shift register (fixed time-window). Given the findings here, one might hypothesize that to capture all dynamics a fairly large shift register would have been required to solve the freeway problem with a FNN or TDNN model.

Secondly, not all neurons are relevant in terms of their contribution in predicting extra

travel time. It appears that those neurons associated with sections on which congestion occurs are considered most relevant. This is not unexpected, since the SSNN is deliberately structured such that each neuron receives only inputs associated with a particular freeway section. However, since the SSNN has a fully connected recurrent layer feeding back past signals from the hidden neurons, it could have easily adopted a weight setting that would not correlate to traffic conditions at all. In 228 dimensional parameter space an in principle infinite number of weight configurations might be able to reproduce the traffic patterns in our training data set. Still, the training procedure consistently produced a weight setting that made sense from a traffic engineering perspective.

We hypothesize that this very particular weight setting is firstly due to the nature of the travel time prediction problem. Predicting travel times is to an extent equivalent to predicting traffic conditions (see chapter 4), and particularly mean speeds. From traffic flow theory it is well-known that traffic state information (mean speeds, densities) travels along very specific paths (characteristics - see e.g. (Hoogendoorn 1999), or (Helbing 1997)) through space and time. Consequently, given the state space structure of our SSNN model it is conceivable there exists also a very particular region in its parameter space that can reproduce these traffic patterns most efficiently. Secondly, the weight initialization routine and Bayesian regularized Levenberg-Marquardt training algorithm prevents the SSNN parameter vector from gliding into other regions of parameter space that may also be able to produce the traffic patterns, but at the cost of much larger weights. As already indicated above, after the regularized training procedure only 66 of the 228 parameters are dubbed "effective".

Finally, in a sense, the SSNN operates much like a macroscopic traffic flow model. This similarity is not coincidental but a direct consequence of the state space structure of the SSNN (eqns 5.1a and 5.1b). There are, however, two clear distinctions:

- The internal states of the SSNN have no direct physical meaning in terms of observable traffic variables, although they correlate positively to link densities. Nonetheless, they can be considered indicators of which freeway sections contribute most to extra travel time (delay) in particular traffic conditions.
- Since the SSNN uses the internal states $\mathbf{x}(t)$ to predict say a travel time of τ seconds on route r for vehicles departing at time instant $t = t_0$, the internal states should probably be regarded indicators for traffic conditions for time instants $t_0 \leq t \leq t_0 + \tau$.

5.9 Summary

In this chapter a novel data driven method for the short term prediction of travel times on freeways was introduced, the so-called state space neural network (SSNN). The

SSNN is in fact a first order context (FOC) memory, which is a specific class of spatiotemporal neural networks. It differs from the well known Elman recurrent neural network (also a FOC) in that it has partial connections between input and hidden layer, that is, each freeway section is represented by one hidden-layer neuron receiving only information from detectors present on that section. The temporal dynamics are dealt with by means of a short term memory, which allows the SSNN to predict travel time based on current measurements in the context of its previous internal states. In many respects, the SSNN operates like a macroscopic traffic flow model. For its design, only the geometry and detector configuration of the freeway route of interest are required, alleviating the model designer of tedious input selection procedures common to neural network design.

On the basis of synthetic data it is shown that the SSNN not only is capable of accurate travel time prediction, but also that its parameter setting is closely related to the actual traffic processes that generate these travel times. This is a direct result of its state space structure (a la traffic flow models) and the Bayesian regularization procedure applied during training. We proposed a heuristic based on backpropagation to calculate the relevance of each neuron and input. Using this relevance heuristic, the SSNN can be pruned of the least relevant neurons, yielding a reduced model with fewer weights, without loss of predictive performance. Although we argue SSNN reduction is not necessary, the procedure clearly shows the usefulness and validity of the relevance heuristic (at least for our application).

In chapter 8 we will show these results are also obtained in case of real data. In the next chapter, however, we will re use the synthetic data from this chapter and analyze the behavior of the SSNN in case of missing data and discuss methods to ensure that also in these conditions the SSNN model still produces accurate output.

Chapter 6

Predicting Travel Time with Unreliable or Missing Data

6.1 Introduction

In the previous chapter we derived a so-called state space neural network (SSNN) capable of accurate (online) freeway travel time prediction, that is, on the basis of simulated data[1]. The results supporting the (objective) validity and applicability of the SSNN, however, were obtained by feeding the models with 100% accurate and reliable (simulated) data. The input data in a real-time situation, collected by a real-time traffic monitoring system, will often consist of corrupted or missing values (e.g. on average 15% of the inductive loops of the Dutch freeway monitoring system (MONICA) may be out of operation or producing unreliable measurements[2]). Before the SSNN model can be applied in a real-time environment, we need to study its behavior under missing or unreliable input data, for which the synthetic data sets from the previous chapter provide a useful test bed.

The effect of missing data on the performance and applicability of the SSNN framework (Fig 6.1) is (potentially) twofold. First, it affects the real-time operation of the SSNN, and second it affects the training procedure. In the latter case both input and target data (PLSB estimated travel times) are affected by missing data. In this case, a "do nothing" strategy results in learning the SSNN model "the wrong thing". The aim of the strategies discussed in this chapter is to provide for a "graceful degradation" of performance (of both the PLSB method as well as the SSNN model) in case of increasing amounts of missing or corrupted data.

[1] In chapter 8 we will show that the SSNN model is also accurate and valid with data obtained from a real traffic data collection system.

[2] Statistic from a week of 1 minute aggregate measurements of inductive loop detectors on the A13 highway between Den Haag and Rotterdam, januari 2002; 9746 of 65536 measurements were classified unreliable (missing or faulty) =14.9%

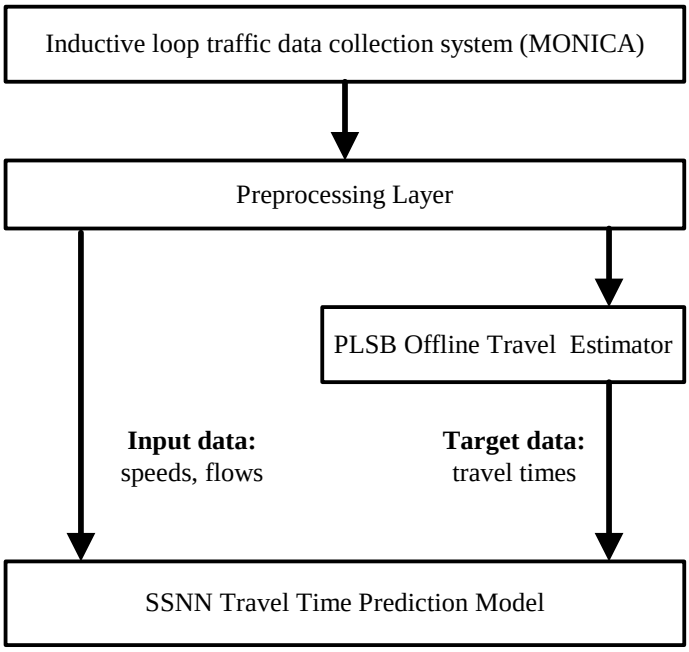


Figure 6.1: Travel time prediction framework: functional dependence of preprocessing layer, offline travel time estimator and SSNN travel time predictor

In principle, the data cleaning / preprocessing layer (Fig 6.1) performs the following tasks (see also chapter 2)

1. **Data Checking**: before possible problems (e.g. missing data) can be adequately tackled, they need to be detected first.
2. **Data Completion**: filling the possible gaps in the data with reasonable replacements
3. **Data Correction**: recheck the now complete data set for validity and consistency and replace / adjust data if required

This chapter presents methods that deal with the last two points, that is, we assume it is known that a certain datum is either missing (second point) or unreliable / inconsistent (third point). In this chapter, we will use the same synthetic data sets as used in the previous chapter (for details, refer to section 5.5). The work in this chapter is largely based on (Van Lint et al. 2003).

6.2 Classification and Representation of Input Failure

In this chapter we use the following definition for input failure:

Definition 12 *Input failure is the occurrence of unreliable or missing data in the input vector. This happens when a measurement device produces data that is (either by the modeler or the device itself) dubbed unreliable, or when it produces no data at all.*

In the ensuing we interchangeably use the terms input failure and detection (or detector) failure, both adhering to the definition above. For experimental purposes we propose a classification of input failure as presented in fig.6.2. Note that in practice all three types may occur simultaneously. The first type of detection failure (fig.6.2a – Incidental (random) failures) occurs due to, for example, temporary power or communication failures in the freeway monitoring system. The second type (fig.6.2b – Structural failures) occurs mainly due to physical damage or maintenance backlogs to the inductive loops or roadside equipment. Although the distinction presented here may not be as crisp in practice, the proposed distinction expresses two extreme configurations of input failure that can be expected in practice, and is hence useful in the investigation of the robustness of travel time prediction models to input failure. A third type of failure (fig.6.2c – Intrinsic failure), measurement noise and bias, is inherent to detection devices and averaging measurements over time in general. An example of the latter type of input failure, which is extensively discussed in chapter 3 is the fact that in MONICA (an inductive loop based traffic data collection system) the arithmetic time mean speed per measurement period is calculated, yielding a biased estimate of the space mean speed (the mean speed on a particular section), which is in fact the quantity of interest. Other known sources of intrinsic data corruption are miscounts, double counts or false counts of vehicles, device calibration errors, round off errors, etc.

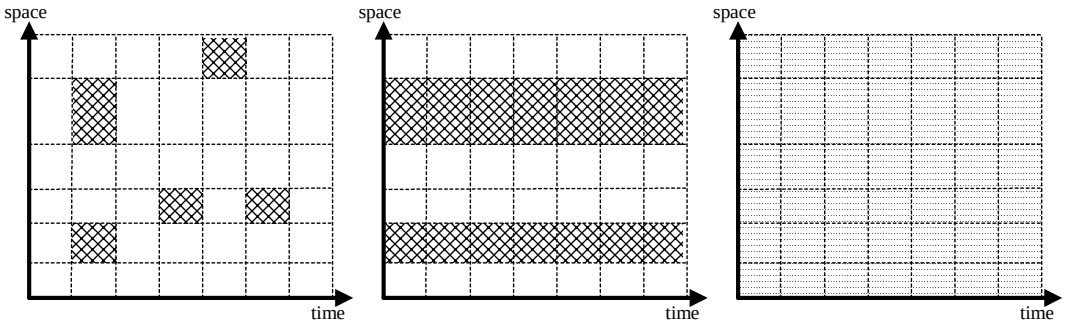


Figure 6.2: Classification of possible input failure (i.e. missing or unreliable data from traffic detectors). In practice a mixture of all types of failure will occur.

Note that in practice a mix of these input failure types will occur. In some cases incidental failure of some (loop) detector may be a prelude to structural failure of that particular loop detector. In this chapter we will address structural and incidental input failure.

To represent an unreliable or missing input datum we set these values to a small negative value close to zero (e.g. -10^{-6}), which is a number that has no physical meaning

for any of the quantities (speeds, flows) that constitute the input to either PLSB method and SSNN model. In this way the pre-processing layer can easily detect these values as “missing”. As a convention these values are represented by the term $null$. In case of a "do nothing" strategy, these $null$ values are omitted in the input to the PLSB method (see section 6.4.1), while replaced by 0.5 in case they are input to the SSNN, since this model requires a fixed sized input vector. The choice for 0.5 as null-value for the SSNN is motivated by the results of the previous chapter. In section 6.5.1 we will show in more detail that this particular choice in fact leads to the desired "graceful degradation" of the SSNN performance. This "do nothing" or "null" strategy is used as a baseline performance indicator for the other strategies proposed

Throughout this chapter, if a measurement from detector d at time period p is dubbed corrupt then all values (i.e. both speed and flow) measured at $\{d, p\}$ are replaced with $null$. Incidental (random) input failure is generated with a random generator J, producing numbers from a uniform distribution on $[0, 1]$, such that each measurement $\{d, p\}$ has an equal probability to be labeled corrupt. If the required level of corruption is set to 20% then a measurement is labeled corrupt if $J(d, p) \leq 0.2$. The maximum amount of incidental input failure considered is set at 40%. In case of structural detection failure ALL measurements $\{d_k, p\}$ from a specific detector d_k are labeled corrupt. Considering all possible combinations of failing detectors leads to a very large amount of test data, which is why we will only consider cases where 1, 3 or 5 detectors are structurally down on the relevant part of main carriage way, particularly the downstream detector on sections 4 to 9 (see table 5.1 on page 112). This implies that the total amount of cases we consider add up to 16 test data sets:

- 1 detector down = 5 test data sets
- 3 detectors down = $\frac{(5)!}{(5-3)!(3)!}$ = 10 test data sets
- 5 detectors down = 1 test set

We hypothesize that analyzing the effects of these failing detectors on the main carriage way gives enough insight into the effects and the possible solutions of structural input failure

In the next section we will first outline four general strategies of dealing with missing data. In the section thereafter we will explore the effect of missing data (caused by both incidental and structural input failure) on the (PLSB) travel time estimation algorithm proposed in section 3.4, while subsequently we will investigate and propose strategies for dealing with missing data for the SSNN travel time predictor. We will return to (and account for) the uncertainty inherent to missing data in the next chapter.

6.3 General Strategies for Dealing with Missing Traffic Data

In this section we first identify various approaches to tackle the missing data problem in traffic prediction tasks. In general, we identify four families of approaches

1. *Null replacement*, that is, leave the data as is (i.e. incomplete), or (if the receiving model requires so) replace missing data with some default value (0.5, zero, one, -99), and let the model receiving the data (in our case the PLSB method and SSNN model) handle the missing data problem. As noted before these default replacements are called $null$ values.

2. *Simple imputation*, that is, replacing missing values by ad-hoc (statistical) procedures. These could include: the sample mean, median or other descriptive statistic, the last known value, a forecasted value by means of a time series or a regression model (even a neural network) or a spatial interpolate.

3. *Model based Imputation*, which in essence is a special case of simple imputation. In this case missing values are replaced by procedures related to knowledge of the (physical) process generating the data, rather than statistical methods. Examples include for instance traffic flow simulation models in combination with Kalman Filters.

4. *Multiple imputation*, in which case the corrupted data set is replicated a number of times, say $N > 1$ times, each in which the missing data are replaced through some simple or model based imputation method. Then, with the N "complete" data sets, N predictions or inferences can be made, which can be statistically summarized. The key notion is that with multiple imputation, the statistical properties of the source data and the inherent uncertainty related to missing data are preserved.

We will discuss some general properties of these methods below.

6.3.1 Null replacement

A clear advantage of using a neural network model such as the SSNN is that it is generally not very sensitive to small disturbances (noise) and some bias in its inputs, given that it is designed and trained properly (Bishop 1995). In this sense, a neural network approach intrinsically satisfies a degree of robustness to noisy and biased input data (fig. 6.2c). This, however, does not hold for incidental and / or structural failure.

Leaving the data as is implies we leave it to the SSNN to handle the missing data problem. This means *we specifically must include corrupted data patterns in the training*

data sets, such that we allow the SSNN to develop an internal model, which takes data corruption into account automatically. Practically this means we train the SSNN with a certain amount of *null* values (in this case each *nulled* input signal is replaced with 0.5). We might expect that the SSNN will be able to learn from corrupted data, but that an increase in robustness is at the expense of its predictive accuracy, since we have deliberately increased the complexity and non-linearity of the problem at hand.

Note, however, that we can only train the SSNN to predict valid travel times with corrupted input if the target data (travel times) are valid. Since the targets are estimated from the same traffic data collection system as the inputs, this implies we still have employ some kind of data cleaning procedure, albeit offline and only for the purpose of generating training data sets.

6.3.2 Simple versus multiple imputation

In practice the most commonly used approach to remedy input-failure is "simple" imputation. Schafer (Schafer 1997) shows that simple imputation schemes tend to change the covariance structure of the input-data and may induce bias. Replacement of missing values with for example the sample mean taken for that particular value biases the estimates of the variance and covariance for that value and other variables to zero, while replacement with regression forecasts may conversely inflate observed correlations. Therefore, Schafer proposes EM and Markov Chain Monte Carlo based approaches that account for the missing values, and the uncertainty they inherently introduce. Examples of these and other approaches to remedy the missing data problem can be found in many fields, including neuro-computing (Armitage & Lo 1994) and (Meert 1996), pattern recognition (Gabrys 2002), climatology (Jeffrey et al. 2001), and medical statistics (Faris et al. 2002), to name a few.

Despite the clear theoretical shortcomings of simple imputation schemes (treated from different perspectives in (Schafer 1997) and (Armitage & Lo 1994)), the results in (Chen et al. 1998), and (Chen et al. 2001) indicate that such simple imputation schemes combined with a neural network based traffic predictor, do produce accurate traffic predictions even when up to 30% of the input data to the model is missing. One (tentative) hypothesis may be that a (properly trained) neural network is robust to the "damage" caused by the imputation scheme applied. Slight changes in the statistical properties of the input data do not cause the neural network to produce inaccurate results. A possible explanation for this phenomenon is that the patterns formed by traffic measurements along one particular route have very different statistical properties than the multivariate data sets used throughout (Schafer 1997), of which all records are assumed to be identically, independently drawn from some multivariate probability distribution. The latter is certainly not the case for data obtained at consecutive time instants from a traffic data collection system, which exhibit strong correlations through both time and space. Another apriori objection to the necessity of advanced multiple imputation schemes for our travel time prediction problem, is that input failure in a

traffic data collection scheme may not be classified as "Missing At Random" (MAR), which Schafer informally defines as *"the probability that an observation is missing may depend on the observed data, but not on the unobserved (missing) data"*. MAR thus means that whether or not a datum is missing does not depend on the fact that other data are missing. In a traffic data collection system detector failure is likely to be correlated over both space (neighboring detectors) and time. The odds on some (inductive loop) detection device producing erratic or no values at some time instant $t + 1$ are likely to be higher if that device produced erratic or no data on time instants $\leq t$. Note that the MAR assumption underlies the EM and Monte Carlo Markov Chain methods for multiple imputation that are proposed by Schafer.

We hypothesize that in operation the SSNN travel time predictor may still yield valid results given a data cleaning procedure based on simple imputation. This may not hold for training and evaluating the SSNN, since central in the short term freeway travel time prediction framework implemented here is the fact that we calibrate the SSNN travel time prediction model with *estimated* travel times, rather than real travel times (Fig 6.1). As outlined above, these estimated travel times *do* depend on both mean speed and speed variance (see section 3.3.3), and are thus potentially affected by (simple) imputation mechanisms that change the statistical properties of the measurements (speeds and flows). Nonetheless, the basic assumptions (statistical independence of observations and Missing At Random) underlying the multiple imputation approaches proposed by (Schafer 1997) are typically violated by data from a traffic data collection system regardless whether or not these data are used offline or online. Moreover, there are more appropriate model based approaches for reconstructing traffic data available.

6.3.3 Model based Imputation

For the missing data problem with respect to traffic forecasting particularly, model based imputation schemes have been investigated in various contexts. In the DACCORD project for example (e.g. (Thijs et al. 1999), (Thijs et al. 1998*a*), (Thijs et al. 1998*b*)), two imputation methods were implemented and extensively tested on three different test sites (Amsterdam, Paris and Padua-Venice), the first based on a Kalman filter, the second on a cross-correlation algorithm. One could argue both methods would qualify as simple imputation methods, nonetheless, the logic behind the methods is based (to a degree) on the spatiotemporal characteristics of the propagation of traffic flow (especially in case of the Kalman Filter).

In essence, both methods induce missing data at detectors from available data at neighboring detectors. As a general rule the former was found to be more reliable than the latter. The strategy deployed then was to apply the Kalman filter if enough measurements were available and to resort to the correlation algorithm otherwise. Most interestingly, in (Thijs et al. 1998*a*) significant gains in the performance (RMSEP) of a number of travel time estimators (amongst others a Piece Wise Constant Speed Based (PCSB) trajectory algorithm) and predictors are reported after applying these

data cleaning strategies. In (Haj-Salem & Lebacque 2002) a data cleaning strategy based on a first order traffic flow model is reported, producing even better results then the methods applied in DACCORD, due to the linear nature of both methods in contrast to the nonlinear nature of traffic flow. The approach in (Haj-Salem & Lebacque 2002) explicitly accounts for the non-homogeneity of traffic processes. However, the increase in performance is expressed in terms of densities and flows, and not in terms of estimated or predicted travel times, although we may expect also gains in the latter quantities.

Tentatively, model based imputation schemes seem the most appropriate data cleaning tool for traffic prediction purposes, because they address the spatiotemporal characteristics of traffic processes. On the down side, however, they require much more modelling effort than statistical methods or simple non-parameterized methods such as interpolation or exponential smoothing. A traffic network needs to be set up to run the model, which means specifying sections, nodes, off- and on-ramps. This network also needs to be maintained during operation. Secondly, a traffic flow simulation model is parameterized, for example, the LWR model requires the (detector-specific!) choice and calibration of the Fundamental Diagram (FD)

$$q(t) = Q^e(\rho)$$

which (statistically) relates traffic flow (or speed) to traffic density. Usually $Q^e(\rho)$ is a concave function which has a maximum (capacity flow) q_C at the so called critical density ρ_C. This critical density separates free flowing from congested conditions. Furthermore, one or more parameters (which also need calibration) determine the shape of the FD. In (Logghe 2003) an overview is presented of different types (triangular, concave, non-concave) of FDs. A third objection to the use of model based imputation is computational expense: filling in missing data on some route, requires the model to run at least one measurement period (of typically a minute) in real-time. Computational expense is particularly relevant in real-time operation of the models.

6.3.4 Brief summary of strategies for missing data

In the travel time prediction framework we identified two distinct effects of missing data on the predictive quality of the SSNN model. The first pertains to the SSNN training, the second to its real-time operation. During training, missing data affects the input and target data (generated with the PLSB trajectory method) for the SSNN. In this case, the accuracy of data cleaning methods is crucial, since we do not want the SSNN to learn 'the wrong thing'. Based on the analysis of the various approaches above, we will therefore investigate the effect of null replacement, simple and model based imputation methods on the performance of the offline PLSB travel time estimator.

During SSNN operation algorithmic performance (computational expense) is of critical importance. Therefore, for online use we will investigate null replacement and

(non-parameterized) simple imputation schemes *only*. The results in (Chen et al. 2001) indicate that such simple imputation schemes combined with a neural network based traffic predictor, do still produce accurate traffic predictions, given that the SSNN is trained properly.

6.4 The Effect of Missing Data on the PLSB Travel Time Estimator

In the piece-wise linear speed based (PLSB) trajectory algorithm (see section 3.4) imaginary vehicles traverse through a grid in space and time (fig. 6.3). This grid is constituted of sections k enclosed by up- and downstream detectors and periods p, during which each of these detectors produces harmonic time averaged speeds of all passing vehicles. In principle, neither sections nor periods need to be of a constant size, as long as for each region $\{k, p\}$ up- and downstream speeds are available. The PLSB algorithm then calculates the exit location and time of an (imaginary vehicle) entering a cell $\{k, p\}$ in space time as a convex combination of the harmonic time mean speeds measured at the up- and downstream end of section k during time period p (equation 3.51 on page 73). Given a fixed departure time t_0 at the most upstream detector, each vehicle trajectory is determined fully and provides an estimate for the mean travel time for vehicles departing at t_0.

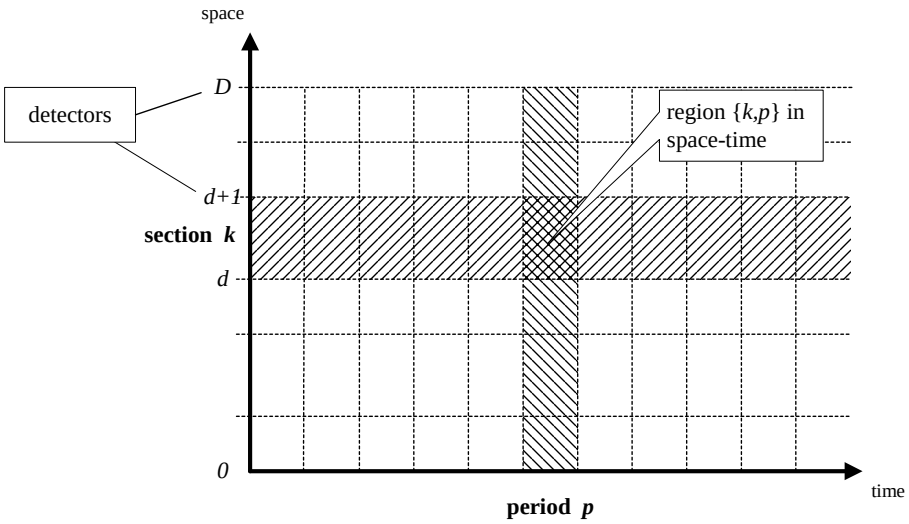


Figure 6.3: Grid of space time regions for the PLSB trajectory method

6.4.1 Data cleaning strategies

The space time grid described above thus yields a rectangular data set of size $D \times P$ (No of Detectors $\times$ No of Periods). If data are missing, obviously, in some cells

$\{k, p\}$ no exit location can be calculated. We will employ the three strategies described earlier to deal with these gaps and test these against data in which we artificially added incidental detector failure.

Null replacement

For the strategy "Null replacement" we do not fill in the gap with a default value, rather, we omit the particular detector output at that period and calculate exit locations and times at the first available measurement (see fig. 6.4). There are two exceptions: (a) at the (spatial!) boundaries, if some measurement is missing, it is substituted with the last known value, and (b) if some imaginary enters a region where the *upstream* detector value is missing it is assumed to enter with its last known speed. In case of structural failure at the most up- or downstream detector on the route, obviously, no replacement is made: the resulting travel times will be biased in the negative direction. Note that in the methodological sense, the PLSB trajectory method is robust since it does not require the input data set to be complete, as long as an exit location downstream can be calculated.

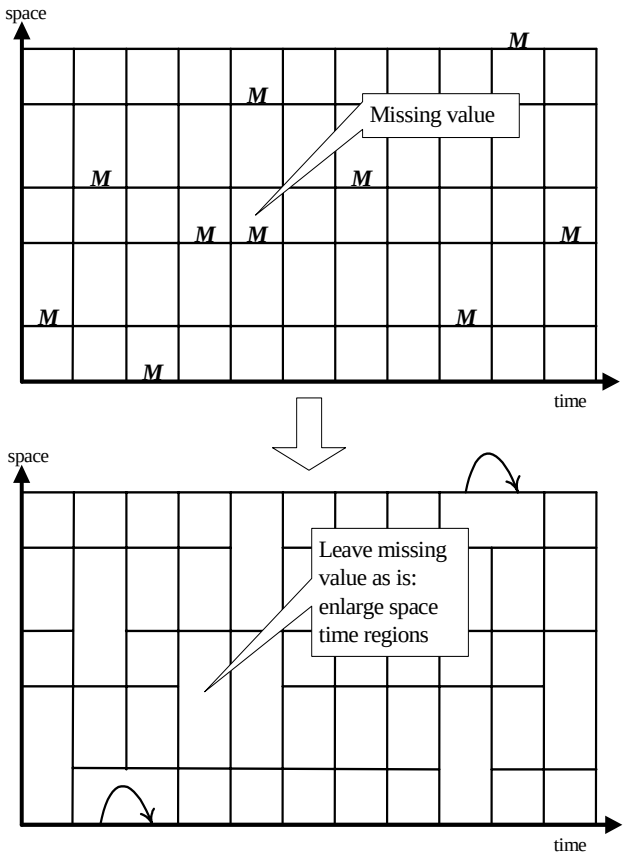


Figure 6.4: Null replacement strategy for the PLSB trajectory method.

Simple imputation through interpolation

The second strategy to be applied is simple imputation as described above. Since the PLSB method is an offline method, we can employ interpolation in both the spatial and temporal direction, given the route is equipped with detectors $d \in \{1, .., D\}$ and a database of measurements U from these detectors periods $p \in \{1, .., P\}$ is available The location of each detector is denoted by x_d. Suppose at some detector d during time period p no data are available, the spatial interpolation procedure we fill in this gap according to

$$U^{space}(d, p) = \begin{cases} U(d + d_a, p) & d + d_a \leq D \\ U(d-1, p) + \frac{x_d}{x_{d+n} - x_{d-1}} U(d + d_a, p) & 1 < d < D; \\ U(d, p-1) & otherwise \end{cases} \quad d + d_a \leq D \tag{6.1}$$

in which $U(d + d_a, p)$ is the first available measurement in the spatial direction. Similarly, in the time direction we can repair the gap with

$$U^{time}(d, p) = \begin{cases} U(d, p + p_a) & p + p_a \leq P \\ U(d, p-1) + \frac{1}{k+1} U(d, p + p_a) & 1 < p < P; \\ U(d, p-1) & otherwise \end{cases} \quad p + p_a \leq P \tag{6.2}$$

in which $U(d, p + p_a)$ is the first available measurement in the time direction. We fill in the gap with the minimum of both interpolates (implying the maximum constraint on traffic throughput (flows) and travel time (speeds)), that is

$$U^*(d, p) = \min \left(U^{space}(d, p), U^{time}(d, p) \right) \tag{6.3}$$

Note that in case of structural detector failure, interpolation in the time direction cannot be used. In that case the value to be imputed comes from eqn 6.1 only. Schematically, the interpolation scheme is outlined in fig. 6.5.

Model based imputation

In this case we try (as in e.g. (Haj-Salem & Lebacque 2002)) a first order Lighthill, Witham and Richards (LWR) model in combination with an extended Kalman Filter. Details on the (discretized) LWR Model are listed in appendix D. The model based imputation then works as follows. We now consider a route of adjacent sections, each centered on a single detector. That means only one detector per section. Again suppose at some detector d during time period p no data are available (Fig 6.6 (a)). Suppose that during period $p - 1$ all data are available from all detectors and (if present) on- and off ramps connected to the sections. These available measurements then constitute the initial conditions (Fig 6.6 (b)).

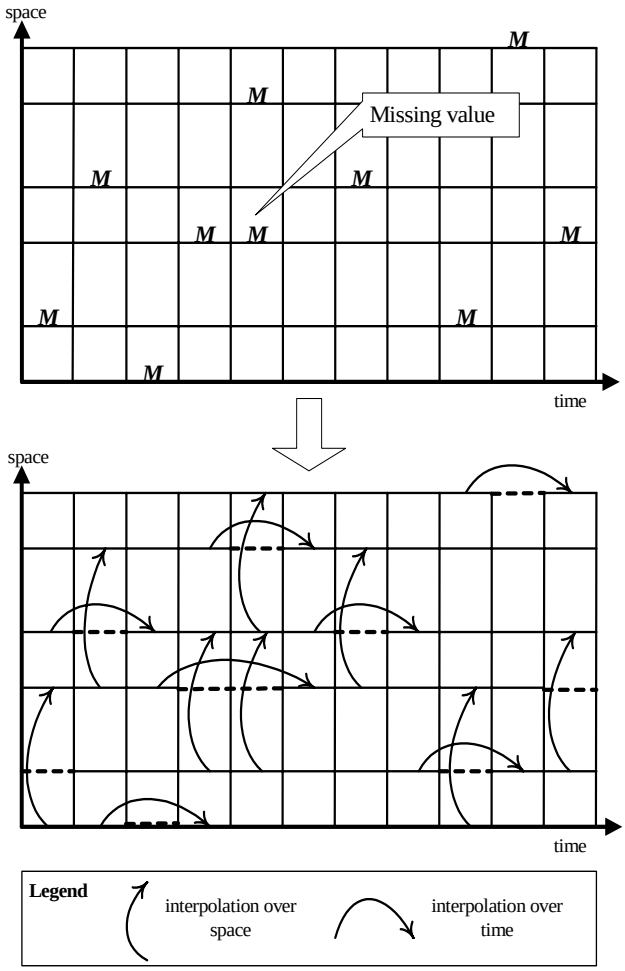


Figure 6.5: Simple Imputation for the PLSB trajectory method: interpolation over space and time

The LWR model is run for one measurement period p (typically 60 seconds) which yields predictions of density, flow (and speed) on each section. After each predictive step (Fig 6.6 (c)), each prediction of flows (and speeds) is corrected by means of an extended Kalman filter (Fig 6.6 (d)). The Kalman Filter combines the model prediction with measurements and weighs these two components by their (assumed) uncertainty. If the model is "far off" (which is tracked by prediction errors) it puts more weight on the measurements, while for example in case of missing data, the Kalman filter puts all weight on the model predictions. A step-by-step explanation of the algorithm is given in appendix D

Let us finally note that a correct calibration of the fundamental diagram, for which we adopt a piece-wise linear (triangular) function, is crucial. For the experiments in this section (on the same simulated data as in the previous chapter), the estimated parameters are listed in table 6.1.

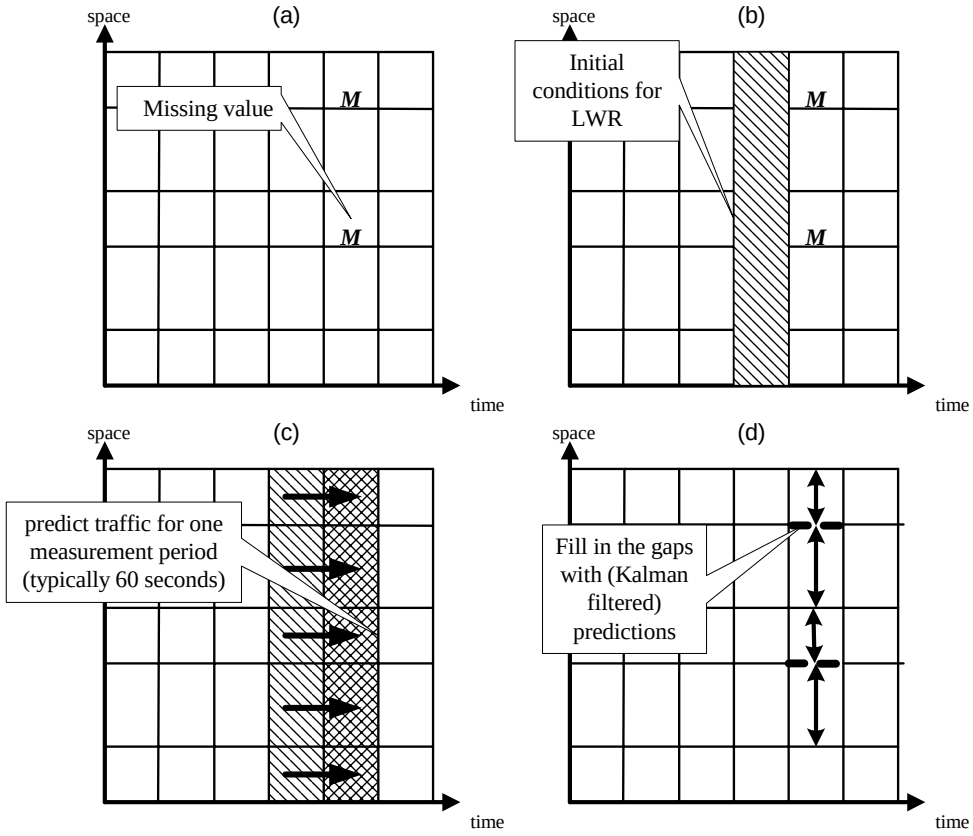


Figure 6.6: Model-based Imputation for the PLSB trajectory method.

6.4.2 Results

Incidental failure

As a baseline indicator, table 6.2 gives the performance of the PLSB method on all 100% clean data sets.

Despite the theoretical advantages of (and apriori bias to) the model based imputation scheme, the results in tables 6.3 to 6.6 clearly indicate the superiority of the interpolation scheme and even the null imputation scheme over the more advanced LWR /

Table 6.1: Estimated parameters of triangular fundamental diagram for the experiiments on synthetic data (obtained with FOSIM)

	value	SI	description
v_f	30.5	m/s	free speed
ρ_c	0.02	veh/m/lane	critical density
$\rho_{\max}$	0.2	veh/m/lane	jam density

Table 6.2: Performance of PLSB travel time estimator under 100% clean data.

	Baseline performance
MRE (%)	1
SRE (%)	7
RMSEP (%)	8

Table 6.3: Performance of PLSB travel time estimator under 10% random missing data. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	none	Interpolation	LWR/Kalman
MRE (%)	1	1	3
SRE (%)	7	6	8
RMSEP (%)	11	8	56

Kalman Filtering technique in terms of the PLSB performance. At all levels of incidental failure the simple methods outperform the more complicated model based method on all performance indicators. It is remarkable that a straight forward and easy-to-implement method such as the interpolation method produces such good results. Even at 40% incidental (random) failure, it is able to reconstruct the data such that the PLSB method is able to produce a RMSEP of 7%, which is equally good as on 100% clean data. Remarkably, in some cases the interpolation method even leads to estimation results slightly better than the baseline results on clean data. Since the PLSB method estimates travel time of an imaginary vehicle using the difference between its exit time out of the last section and its entry time into the first section of a route, a vehicle could experience many different trajectories which approximately lead to the same travel time. We hypothesize this explains the excellent results of the interpolation method. As long as the general traffic pattern remains in tact, PLSB estimated vehicle trajectories most probably lead to very similar travel times, regardless of small deviations in speeds (due to the interpolation method) along the route.

There are several plausible reasons why the model based method does not perform that well. The first is in the fact that as data are considered missing, the (LWR) model predictions on those locations are considered 100% reliable. During non-stationary periods (as congestion sets in or dissolves) the model often "misses" the sometimes abrupt non-linear transitions between free-flow and congested conditions if during these time periods increasing amounts of detector information is missing. As data becomes available, the difference between measurements (congested) and model predictions (still free flow) is large and the Kalman Filter requires some time to "push the model back". This is amongst other things due to the fact that the LWR model has no dynamic equation for speeds, which are considered a non-linear but static function of

Table 6.4: Performance of PLSB travel time estimator under 20% random missing data. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	None	Interpolation	LWR/Kalman
MRE (%)	2	0	4
SRE (%)	9	6	10
RMSEP (%)	13	7	19

Table 6.5: Performance of PLSB travel time estimator under 30% random missing data. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	None	Interpolation	LWR/Kalman
MRE (%)	2	0	7
SRE (%)	10	6	13
RMSEP (%)	14	7	25

density (the fundamental diagram). As a result, the differences between free-flow and congested conditions in terms of speeds are unrealistically high. In the former vehicles drive with their desired speed (30.5 m/s, see table 6.1), even through near critical densities, and instantaneously break as they enter in above critical densities. Again, as the LWR/Kalman Filter fails to track the onset of congestion due to missing data, the difference in travel time (the reciprocal of speed!) will also be high in those periods.

A second and related reason is that we have estimated one fundamental diagram for all detector locations. Since the fundamental diagram should not be regarded as a causal model, but rather a statistical equilibrium relation for a given location (geometry), the results here are certainly biased. It would most likely be much better to maintain a different fundamental diagram for each detector location. Moreover, the parameters of this fundamental diagram should be kept adaptive and part of the state equation. Finally, we hypothesize that a so-called second order traffic flow model (e.g. Payne-type models (Hoogendoorn & Bovy 2001)) might be better suited for this particular data reconstruction task, since it contains a more realistic (dynamic) equation for speeds, which accounts for the dynamics (anticipation) of driver's responses and the (convective) processes which cause speeds over both time and space to be related. Nonetheless, it is uncertain whether such a more complex model based procedure would yield results similarly good as the much simpler interpolation method.

Note that the purpose of the data reconstruction task here is to provide valid input matrices of speeds to the PLSB travel time estimator and that we assess the performance of each repair method with the PLSB results, not with the errors on each replacement

Table 6.6: Performance of PLSB travel time estimator under 40% random missing data. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	None	Interpolation	LWR/Kalman
MRE (%)	2	-1	9
SRE (%)	11	6	17
RMSEP (%)	17	7	35

individually, which was done in for example (Haj-Salem & Lebacque 2002). In the latter case, different results might be obtained.

Structural failure

As outlined in the introduction of this chapter, we test the robustness of the PLSB method in case when 1, 3 and 5 detectors on the main carriage way are down. As a baseline indicator, again refer to table 6.2 for the performance of the PLSB method on all 100% clean data sets.

Table 6.7: Performance of PLSB travel time estimator when 1 out of 5 loop detectors on the (congested part of the) main carriage way is structurally failing. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	none	Interpolation	LWR/Kalman
MRE (%)	0	2	1
SRE (%)	10	7	15
RMSEP (%)	10	9	18

Table 6.8: Performance of PLSB travel time estimator when 3 out of 5 loop detectors on the (congested part of the) main carriage way are structurally failing. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	none	Interpolation	LWR/Kalman
MRE (%)	2	2	4
SRE (%)	9	9	25
RMSEP (%)	13	13	35

Table 6.9: Performance of PLSB travel time estimator when 5 out of 5 loop detectors on the (congested part of the) main carriage way are structurally failing. The columns depict null imputation, interpolation and LWR/Kalman Filtering as repair strategies.

	none	Interpolation	LWR/Kalman
MRE (%)	3	3	-9
SRE (%)	12	12	65
RMSEP (%)	19	19	82

In case of structural failure a similar picture emerges as with incidental failure. Null and simple imputation outperform the LWR/Kalman filter approach by far (Tables 6.7 to 6.9). When 5 detectors fail, the LWR/Kalman method produces a RMSEP of over 80%. In terms of both bias and variance the null imputation method performs equally well as the interpolation scheme. Clearly, structural failure is the more serious type detector failure. When 5 out of 13 detectors fail at once (yielding about 39% input failure of detectors on the main carriage way) larger travel time estimation errors occur than when failure is random. At 40% incidental failure the interpolation method still produces a RMSEP of 7% (table 6.6), in case of structural failure the RMSEP adds to 19% (table 6.9).

6.5 The Effect of Missing Data on the SSNN Travel Time Predictor

6.5.1 Data cleaning strategies

As discussed earlier, handling the missing data problem in SSNN operation requires procedures that are computationally efficient. Hence, we adopt null and non-parameterized imputation strategies only. The prerequisite is that the SSNN model is designed and trained properly. We will test the data cleaning strategies as shown in table 6.10. Strategies S0 depict null replacement strategies in which we train the SSNN model with increasing amounts of corrupted data. The other strategies (S1 to S3) use a SSNN model trained with 100% clean data and simple imputation strategies to clean the input data on beforehand. In the next two subsections these methods are briefly explained.

Null replacement: training the SSNN with missing data

The first strategy (S0 in table 6.10) is to leave missing values as is. Since the SSNN has a (apriori) fixed topology, it can only accept input in this (apriori) defined format. This

Table 6.10: Strategies to enhance robustness of the SSNN travel time prediction framework in real time operation.

Strategy	Corruption Traindata	Corruption Testdata	Imputation Scheme
S0	0-40%	0-40%	null
S1	0%	0-40%	Interp
S2	0%	0-40%	MA
S3	0%	0-40%	MA/Interp.

implies that missing data values should be replaced by some default *null* value, which allows the SSNN to detect it as such and distinguish it from valid measurements, while still produce reasonable output, such that "graceful degradation" occurs for increasing amounts of missing data.

As mentioned earlier, the choice here is made to replace all missing values with 0.5. To illustrate this particular *null* value does lead to a "graceful degradation" of performance, recall the sigmoid transfer function of the previous chapter, which every hidden neuron in the SSNN uses to transform its input vector $\mathbf{z} = [z_1, ..., z_N]$, $z_j \in \langle 0.1, 0.9 \rangle$, $\forall j$ to the scalar value $h \in \langle 0, 1 \rangle$

$$h = \frac{1}{1 + \exp\left(w_0 + \sum_{i=1}^{N} w_i z_i \right)}$$

Fig. 6.7 shows the neuron output h as a function of the number of missing values in the input vector (which we arbitrarily set to length 9) for four different *null* values, -100, -1, -10^{-6}, and 0.5. Each of the four graphs shows ten different neuron responses for increasing numbers of *null* values, based on 10 different weight settings, chosen randomly from a zero mean normal distribution with unity variance. Clearly, fig. 6.7 (top-left) shows that a large *null* value (e.g. -100 or $+100$) leads to oscillatory behavior even at small percentages of missing data. A small but significant (of the same order as $\mathbf{z}$) negative value of -1 (fig. 6.7 (top-right)) does not lead to oscillation, but certainly to an unstable response, since it effectively steers the sigmoid function in the wrong direction (either positive or negative depending on the weight setting). A small value (around zero) does improve stability, but still leads to an unwanted neuron response. In case of the SSNN particularly, near zero values represent either very low (or even slightly negative) speeds or very low or zero flows. Since both flow and speed from one detector are replaced with *null* in case the detector fails, the resulting *null* value represents a very unusual and extreme input signal[3]. Choosing a *null* value in

[3] e.g. closure of all lanes (zero flow + zero speed) or a complete empty road (no flow and no mean speed available)

the neurons input domain (0.5, which is the mean of the minimum and maximum input values) as in fig. 6.7 (bottom-right) yields by far the most stable result. As explained in the previous chapter, 0.5 represents a "mean" input-signal (moderate flows, moderate speeds), that allows for a graceful deterioration of neuron output. In the extreme case when all input-signals are missing, the neurons' response will equal the mean response it has learned from the training data.

We will train the SSNN with data sets corrupted with a "mix" of both incidental and structural failure. First, five training data sets are generated with an increasing percentage of incidental (random) failure (0%, 10%, 20%, 30% and 40% respectively). Note that each training data set consist of the same 8 six-hour sequences as outlined in section 5.5.3 on page 114, and that the 0% corruption training data set is in fact the one used in the previous chapter. Next, in each of the four corrupted training data sets each detector is "shut down" for longer periods of time. Arbitrarily, we choose structural failures of 30 consecutive periods (30 minutes). At any time instant no more than two detectors are structurally failing. Effectively, this causes an increase of the total percentage of input failure in each training data set of 1 to 2%. The hypothesis is that the combined effect of incidental (random) and structural failure should enable the SSNN model to develop a parameter setting that implicitly incorporates both types of detector failure.

Simple imputation: interpolation and exponentially moving average.

Secondly, we propose three strategies (S1-S3 in table 6.10) to tackle input failure by means of three simple non-parameterized imputation schemes to be applied in the preprocessing layer. In general traffic patterns are strongly correlated over both space and time. Arguably, in densely meshed networks (in urban areas) spatial correlation may be considered the predominant factor, certainly if traffic control is applied. In less densely meshed networks or freeway corridors as discussed here both temporal and spatial correlation should apriori be considered.

The first simple imputation method (strategy S1 in table 6.10) considers spatial traffic patterns on the route of interest. If measurements on intermediate locations on the route are fraud we can use spatial interpolation to fill in the gaps. Details on this method can be found in section 6.4.1. The second simple imputation method (S2) is to consider time series of individual inputs only. Clearly, traffic measurements on fixed locations exhibit strong autocorrelation over time. Missing or corrupt measurements $U_d\,(t+1)$ from detector d at time instant $t+1$ can be replaced by a forecast $f_d\,(t+1)$ of a simple time series model, in our case by an exponentially moving average:

$$f_d(t+1) = f_d\,(t) + \alpha\,(U_d\,(t) - f_d\,(t))\,, \alpha \in [0,1] \tag{6.4}$$

with α typically set to 0.3. The quantity $f_d\,(t)$ denotes the exponential forecast for input d at time instant t, while $U_d\,(t)$ denotes the d^{th} input value at time instant t.

Obviously, more complicated AR(I)MA models, multivariate regression models or non-linear forecasting techniques can be used, but these require offline design and calibration. Note that in case of structural detector failure, a time series forecast, such as (6.4), is not feasible, since it requires at least one valid measurement per input time series.

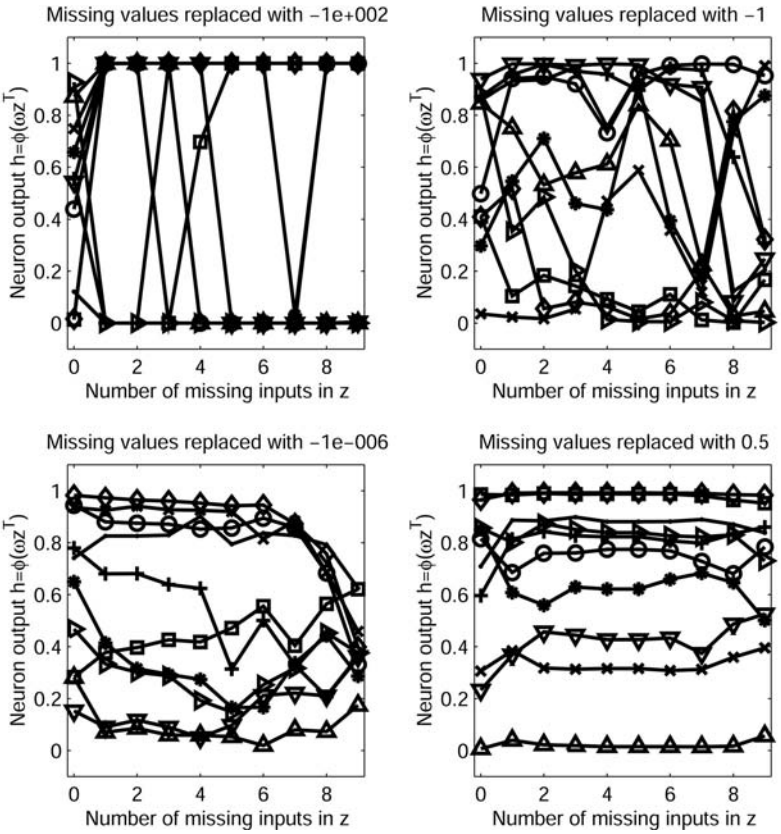


Figure 6.7: Effect of setting missing data values to different *null* values on a single neuron with a logistic transfer function. Each graph shows ten responses representing ten random weightsettings.

Finally (strategy S3 in table 6.10), we can combine both methods similarly to the interpolation method over space and time described in section 6.4.1. In this case the imputed value is the minimum of the values obtained from the Moving Average (MA) and Interpolation (Interp) method.

6.5.2 Results

Incidental input failure

Table 6.11 presents the RMSEP performance (which is ratio between RMSE and mean travel time in percentages) for all proposed strategies on all available test data sets. The

second row first shows that a SSNN model trained with 100% correct data is not able to handle missing data, which is to be expected. The *null* values replacing missing data are outside its input domain and its performance deteriorates steadily as the percentage of random failure in the test data sets increases. This strategy basically is provided as a baseline comparison for the other strategies.

Table 6.11: RMSEP (%) performance of all strategies applied to datasets with incidental input failure. The columns depict the test datasets with increasing incidental input failure, the rows the different strategies S0 (null imputation), S1 to S3 (simple imputation).

Test Datasets / Strategies	Incidental Failure				
	0%	10%	20%	30%	40%
S0 (0%)	8	19	33	52	77
S0 (10%)	11	10	14	18	26
S0 (20%)	13	12	11	12	15
S0 (30%)	18	15	13	12	12
S0 (40%)	23	20	17	14	12
S1 (Int)	8	8	9	10	12
S2 (MA)	8	8	8	8	9
S3 (MA/Int)	8	8	8	9	10

The next four rows in table 6.11 show that training the SSNN with increasing amounts of corrupted data does increase the robustness with respect to missing data but at a price in terms of performance. SSNN models trained with X percent of incidental input failure are capable of producing RMSEP values of 11-12% when tested against test data sets with similar incidental input failure levels (X percent). Obviously, to make the SSNN model robust, the amount of examples of input failure should be comparable to the amount expected in practice.

The last three rows in table 6.11 show the performance of an SSNN model trained with 100% correct data on increasingly corrupt test data sets in conjunction with three simple imputation strategies (Interpolation, Moving Average and a combination of these two respectively). Remarkably, all three outperform each of the S0 strategies, with the MA procedure (strategy S2) performing best. Even at 40% incidental input failure, the MA procedure reconstructs the data such that the SSNN model can still predict travel times as accurately as with 100% clean data (increase of RMSEP from 8 to 9% only). These results support the hypothesis that a (properly trained) neural network is robust to the "damage" caused by simple imputation schemes applied, that is, slight changes in the statistical properties of the input data do not cause the neural network to produce inaccurate output.

Structural input failure

Tables 6.12, 6.13 and 6.14 show mean and standard deviation of the relative error and the RMSEP performance of the SSNN model against data sets in which respectively 1, 3 and 5 detectors on the congested part of main carriage way are structurally failing. Typically, null imputation (S0) strategies lead to underestimation, since the imputed signal (0.5) triggers the SSNN to a mean response. Again, the simple imputation strategy (spatial interpolation) outperforms the (null imputation - S0) strategies in which the SSNN is trained with structural detector failure. In the worst case scenario (5 failing detectors) the spatial interpolation procedure allows the SSNN to still produce reasonable predictions with a relative error of 1% and a RMSEP of 14% (against 8% on clean data), as opposed to 13% and 20% respectively for the "best" S0 strategy (in which the SSNN is trained with 40% input failure).

Table 6.12: Performance of SSNN travel time predictor when 1 out of 5 detectors on the (congested part of the) main carrigaway is structurally failing.

	Interp	0%	10%	20%	30%	40%
MRE (%)	1	12	0	-4	-5	-7
SRE (%)	6	10	9	10	14	17
RMSEP(%)	9	17	11	13	17	21

Table 6.13: Performance of SSNN travel time predictor when 3 out of 5 detectors on the (congested part of the) main carrigaway is structurally failing.

	Interp	0%	10%	20%	30%	40%
MRE (%)	1	39	20	7	1	0
SRE (%)	8	20	19	14	13	14
RMSEP(%)	10	33	20	15	15	17

Table 6.14: Performance of SSNN travel time predictor when 5 out of 5 detectors on the (congested part of the) main carrigaway is structurally failing.

	Interp	0%	10%	20%	30%	40%
MRE (%)	1	93	62	31	15	13
SRE (%)	10	47	37	25	18	17
RMSEP(%)	14	72	50	30	22	20

6.6 Discussion on Imputation Strategies for PLSB and SSNN

In the analysis presented above, simple imputation schemes produce surprisingly good results both for the PLSB offline travel time estimator as well as the SSNN travel time predictor.

Overall, the interpolation method is the best imputation scheme to tackle the missing data problem for the PLSB offline travel time estimator. In case of both structural and incidental detector failure it is able to reconstruct the data such that plausible travel times can still be estimated. The performance of the more advanced LWR/Kalman filter is disappointing, albeit that several plausible causes and possible improvements can be identified. Nonetheless, in its current setting, at moderate to high percentages ($>10\%$) of incidental failure and as more then one detector is structurally down, this method does not produce valid replacements in terms of the PLSB performance. Given the extra effort required in terms of modelling and parameter estimation (fundamental diagram) required for an improved version of the LWR/Kalman method, we strongly prefer the simple non-parameterized interpolation procedure.

A similar conclusion can be drawn for handling the missing data problem in real-time. Due to the generality of the S0 strategies (SSNN models are trained with a mix of structural and incidental input failure), these more robust SSNN models offer a trade off between robustness and predictive accuracy. One could argue that it is possible to construct and train SSNN models for each particular structural detector failure problem and use these specifically trained SSNN models in each of those particular situations. This would, however, require a large number of SSNN models to be trained. In our case there are 8191 possible combinations[4] of structurally failing detectors on the main carriage way alone, yielding the same number of specialized SSNN models. On a route equipped with 25 detectors, this number increases to over three million combinations. Off course, this number could be significantly reduced by engineering judgement, for example by pre selection of likely combinations of failing detectors (based on history). This, however, does not guarantee a "waterproof" and hence robust framework. Furthermore, it inherently means that location specific design is required for the SSNN framework to be successfully applied.

Although it is clear there are many alternative imputation strategies that could be applied for both the offline (PLSB) and real-time (SSNN) data reconstruction task such as ARIMA models, neural networks, higher order model based / Kalman filtering approaches and multiple imputation schemes, in both cases the principle of parsimony

[4]There are 13 detectors on the main carriage way, yielding

$$\sum_{k=1}^{13} \frac{13!}{k!\,(13-k)\,!} = 8191$$

possible combinations of failing detectors

(which favors simple solutions as long as they perform well) leads us to conclude that simple non-parameterized imputation by means of spatial interpolation and a moving average offer a robust and easy to implement method for handling the missing data problem in the SSNN framework. These findings provide strong evidence that the SSNN model is robust to the 'statistical damage' done by the simple imputation procedures. There is enough redundancy in its parameter setting to allow partial distortion of its inputs without serious deterioration of its output. In this sense the proposed combination of SSNN and simple imputation provides for a framework that is robust with respect to both incidental and structural input failure, certainly at degrees of input failure that occur in real life.

6.7 Summary

In this chapter we analyzed the behavior of the SSNN based short term freeway travel time prediction framework under missing data (input failure). This effect is twofold. First, input failure comprises the quality of the offline travel time estimates (with the PLSB trajectory method) we need for training the SSNN model, and secondly, missing data deteriorates the performance of the SSNN in real-time. We categorized input failure into incidental (random) failure, structural failure (data from a detector is missing for longer time periods) and intrinsic failure due to peculiarities of detection equipment. Methods to handle intrinsic failure, which predominantly affect the PLSB estimator and not the SSNN model, were discussed extensively in chapter 3.

From literature we derived four strands of approaches to tackle the missing data problem, that is, null imputation (do nothing or fill gaps with some default value), simple imputation (fill gaps with sensible replacements such as regression forecasts), model based imputation (in combination with for example Kalman filters) and multiple imputation (creating multiple plausible input data sets). We assessed the first three on both incidental and structural failure in terms of the PLSB performance and the first two on the SSNN performance. In both cases simple non-parameterized imputation methods (spatial interpolation and exponential moving average (MA)) outperformed all other methods. For example, even at 40% incidental input failure, the MA procedure reconstructs the data such that the SSNN model can still predict travel times as accurately as with 100% clean data (increase of RMSEP from 8 to 9% only).Although several plausible causes and possible improvements can be identified for the other methods, we argue these possible improvements do not justify the extra effort required in terms of modelling, parameter estimation and SSNN training. The combination of SSNN model and simple imputation methods provide for robust framework for online freeway travel time prediction

However robust the proposed framework, missing data deteriorates performance and hence reduces the reliability of the travel time prediction framework. In the next chapter we will revisit some of the concepts discussed here and propose methods to quantify

that reliability in a statistical sense. As we will show, without actually measuring predictive performance, the SSNN model enables us to quantify the uncertainty inherent to missing data and provide a "warning message" in cases something is wrong with the data or the model.

Chapter 7

Quantifying Uncertainty in Travel Time Prediction

7.1 Introduction

In chapter 5 we presented an accurate prediction system of travel times on freeways based on the so-called state space neural network (SSNN) and in the previous chapter (6) we developed robust methods of handling missing or corrupt (input)data. In this chapter we assess the reliability of this SSNN model. As noted in the introduction of this thesis, in the Oxford Dictionary reliable is defined as *"consistently good in quality or performance, and able to be trusted"*, and reliability as *"the quality of being reliable"*. Although according to this definition reliability encompasses a broad range of qualities (in fact all the qualities we assess in this thesis), we will restrict ourselves in this chapter to quantify reliability in a statistical sense by means of confidence and prediction intervals.

Confidence intervals reflect the fact that the output (travel time) of the SSNN is uncertain, due to its (estimated) parameters. As such, the SSNN output should be regarded as a distribution determining the maximum probable predicted value and the spread around that value given the SSNN parameters and the input data. Taking a Gaussian distribution for the SSNN output (which we will do in the ensuing) yields that the maximum probable value in fact equals mean travel time and that the spread around the mean can be straightforward expressed using the variance / covariance matrix of the parameters. It should be emphasized that confidence intervals have nothing to do with the (statistical) distribution of travel times (per departure time period). They express the uncertainty in the SSNNs' prediction based on the uncertainty in its parameters (Van Lint 2003).

Prediction intervals around the prediction entail the *possible (or most plausible) distribution of travel times around that prediction*, that is, the most plausible (possible) range of travel times a vehicle is likely to encounter, given the mean prediction. As

we will demonstrate below, this so-called predictive distribution is obtained through analysis of historical travel time data (the same with which the SSNN is trained). The parameters of this distribution (or some of them) can be expressed in terms of mean travel time. A consequence of the fact that in operation mean travel time is not observed but *predicted,* is that as predicted mean travel time is wrong (biased), the predictive distribution is also wrong (at least in terms of its central moment) and can not be used to obtain statistical upper or lower bounds for the prediction. Secondly, the prediction intervals here do not encompass temporal effects influencing travel times (e.g. day-to-day variability) introduced in section 2.3 (recall fig. 2.4 on page 28), but reflect the distribution of travel times for vehicles departing in the same period p on some freeway route r , given the prevailing traffic conditions on r during their time en route.

As we will show in this chapter the total predictive distribution is due to (a) the distribution of actual travel times itself (not all vehicles departing in some time period drive with the same speed); (b) the fact we train our model with estimated travel times (which causes estimation errors we have to account for) and (c) the parameters of the short term travel time prediction model (the SSNN). The last source of uncertainty by itself yields confidence intervals, while all sources together yield prediction intervals that necessarily enclose these confidence intervals. Quantifying this total uncertainty in a statistical sense serves two main purposes

1. It allows us to put error bars on (mean) travel time predictions

   As noted above *these error bars (prediction intervals) strongly depend on distributional assumptions on the various sources of uncertainty and moreover, do not account for structural prediction errors but only indicate the possible spread around predictions.*

2. It provides for a warning mechanism in cases we have low confidence in the predictive quality of our model

   Arguably this is the more important purpose. Large confidence intervals coincide with large errors. Consequently, large confidence intervals provide a signal that something might be wrong with either the model or the data with which it is fed.

This chapter is organized as follows. In the first section the distribution of travel time is investigated under different traffic circumstances. In the next section we review the travel time framework presented earlier, and subsequently identify the three sources of uncertainty mentioned above. In the final sections, some qualitative and quantitative results are shown and conclusions are offered. The next chapter will combine the findings of this and the previous two chapters and apply these into a real-time traffic data collection framework.

7.2 Distribution of travel time

As in the previous two chapters, we base our efforts on the synthetic data set as presented in section 5.5. As will be shown below, the distributional characteristics of these simulated travel times are not very realistic, as compared to the distribution of real travel times in the next chapter. The main difference is that the distribution based on simulated data becomes very flat and skewed in congested conditions, which in real life doesn't occur (Van Lint 2004). Nonetheless, in this chapter we will provide the key concepts that will also be applied (successfully) to real travel times.

Recall that mean travel time on a route r for vehicles departing in period p is defined as the average time it takes these vehicles to traverse that particular route, that is

$$\mu_\tau(p) = \int \tau g(\tau)\, d\tau = \frac{1}{N_p} \sum \tau_i(t),\ t \in p \tag{7.1}$$

in which it is assumed individual travel times are drawn from some probability density function (pdf) $g(\tau)$ and that $\tau(t)$ denotes the travel time of an individual driver departing on route r at time instant t and a total of N_p vehicles depart in period p. The size of aggregation period p off course largely influences the shape of this distribution. In this dissertation thesis this period equals 1 minute. The questions to be answered then are (a) which pdf represents the travel time distribution best, and (b) what are the (possibly time dependent) parameters of this distribution. Since travel times are by definition positive, $g(\tau)$ should be nonzero for positive values only. We therefore consider the *gamma*[1] and *lognormal*[2] distributions as reasonable candidates.

Fig. 7.1 shows for test data set 2 (normal weekday traffic pattern with recurrent congestion) a histogram and estimated gamma and lognormal probability density functions under free flow conditions (in which the mean speed on the main carriage way is above 90 km/h). The figure shows the histogram is slightly skewed to the left, and indicates the lognormal distribution fits the data best. This is confirmed with a Kolmogorov-Smirnov (KS) test[3], which produces the highest probability for a lognormal distribution. Nonetheless the KS test rejects both distributional hypotheses (gamma and lognormal) at a significance of 95%. This is predominantly due to the size of the sample

[1] the gamma pdf is

$$g(\tau|a,b) = \frac{1}{b^a \Gamma(a)} \tau^{a-1} e^{-\tau/b}$$

[2] the lognormal pdf is

$$g(\tau|\mu,\sigma) = \frac{1}{\tau\sigma\sqrt{2\pi}} e^{-(\ln\tau-\mu)^2/2\sigma^2}$$

[3] The Kolmogorov-Smirnov test is a robust and distribution-free statistical test which calculates the maximum distance (D-statistic) between the empirical cumulative distribution of a dataset and a theoretical cumulative distribution (e.g. gamma, lognormal) with known parameters, based on the desired level of significance and the number of observations.

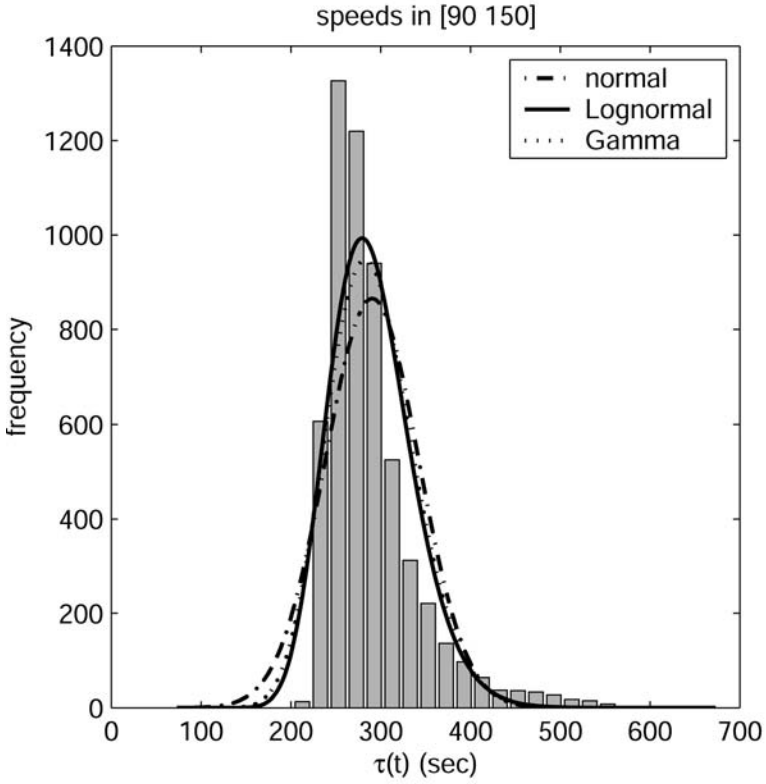


Figure 7.1: Histogram and estimated normal, lognormal and gamma probability density functions of travel times in free-flow conditions (average speed on main carriageway $>$ 90 km/h).

(over 5000 observations). For large samples the KS statistic (also referred to as *D*-statistic), which measures the distance between the empirical cumulative distribution function (*cdf*) and some known *cdf* (e.g. the lognormal), must be very small in order *not* to reject a distributional hypothesis. In illustration, Fig 7.2 shows the empirical *cdf* of travel times from test set 2 in congested conditions (mean speed on main carriage way $<$ 50 km/h) and an estimated lognormal *cdf*, which are in fact a fairly close match.

When we view $g(\tau)$ for heavily congested periods a slightly different picture emerges (fig. 7.3) than in case of free-flow conditions. The distribution now is more strongly skewed to the left and has a long right-tale reflecting a significant number of vehicles departing in period p that actually experience much longer travel times than the mean travel time for vehicles departing in period p. Furthermore, travel time variance in these congested conditions is five times larger than in free flow conditions. Although the KS test rejects a lognormal hypothesis in both congested and free-flow conditions, we argue it is reasonable (and convenient) to assume lognormally distributed travel times. Most importantly, the actual distribution of travel times from the micro-simulation is not very realistic compared to real travel times (see next chapter), so that a very accurate distributional assumption on these simulated travel times is not of primary interest. Moreover, from the candidate distributions, the lognormal distrib-

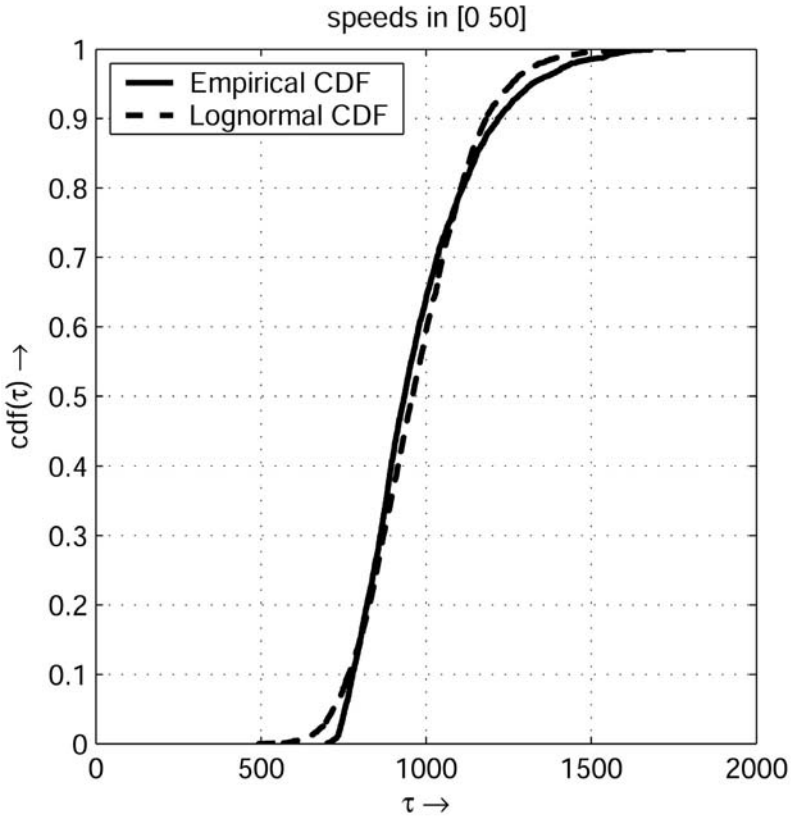


Figure 7.2: Empirical versus estimated lognormal cumulative distribution function (cdf) of travel times in congested conditions (mean speed on main carriageway < 50 km/h)

ution produces the smallest KS-statistic in both free-flow and congested conditions. In the next section we will use the lognormality assumption to quantify the uncertainty associated with the SSNN prediction.

7.3 Three Sources of Uncertainty

In the previous section the distribution of travel times of vehicles departing on some route r in period p is discussed. We identify three components that constitute to error bars around the predictions made by the SSNN travel time predictor. The first component is the distribution of travel time itself, which as shown above, we assume is lognormal. The second component relates to the fact that the SSNN model is trained with offline estimated travel times rather than real travel times and the third component is due to error bars on the parameters of the SSNN model. The latter component also captures the uncertainty due to the input data, since the SSNN maps these (non-linearly) to the output data through its parameters. Below we discuss each of these components

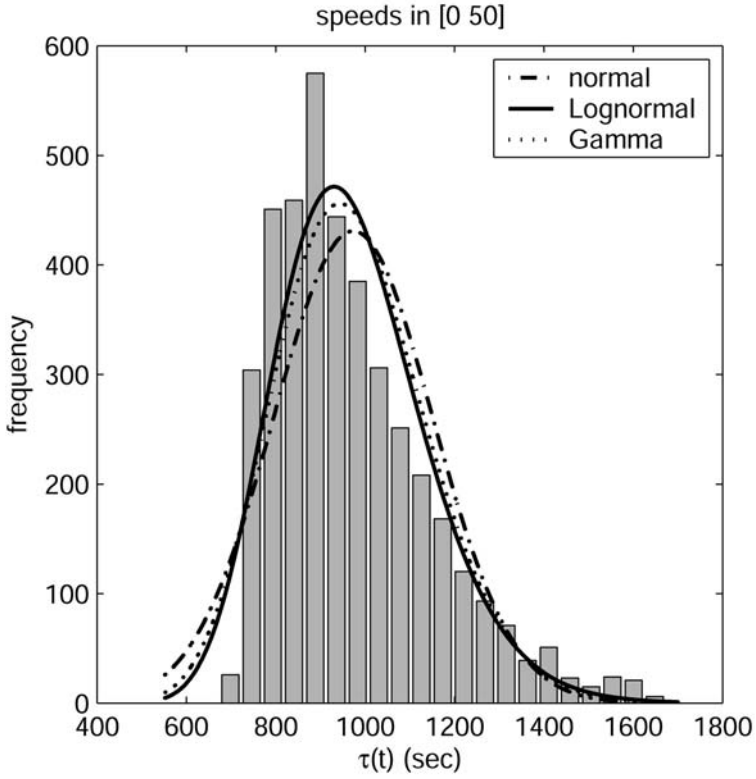


Figure 7.3: Histogram and estimated normal, lognormal and gamma probability density functions of travel times in congested conditions (average speed on main carriageway < 50 km/h).

7.3.1 Uncertainty inherent to the distribution of travel time

Using the SSNN travel time prediction model introduced in chapter 5, the travel time prediction problem can be cast as a regression problem as follows

$$\tau(p) = G(\mathbf{u}(p-1), \psi) + e_\tau(p) \tag{7.2}$$

in which $\tau(p)$ denotes the expected travel time for a vehicle departing during period p, G denotes the SSNN model , $\mathbf{u}(p-1)$ the SSNN input (data measured on route r during time interval $p-1$), ψ denotes an adjustable vector of all SSNN parameters and $e_\tau(p)$ denotes a random error term. In words, the SSNN model is trained to reproduce the systematic processes that generate mean travel times, while stochastic (random) fluctuations in travel times are represented by $e_\tau(p)$. Since we assume travel times are lognormal distributed, we define this error term as follows

$$e_\tau(p) + \widehat{\mu}_\tau(p) \sim LN(m_\tau(p), s_\tau(p)) \tag{7.3}$$

in which (MathWorld 2004)

$$\begin{aligned} m_\tau(p) &= \ln\left(\widehat{\mu}_\tau(p)\right) - s_\tau^2(p)/2 \\ \widehat{\mu}_\tau(p) &= G\left(\mathbf{u}(p-1), \psi\right) \end{aligned}$$

Note that the maximum (the mode) of the lognormal distribution is in fact smaller then the mean $\widehat{\mu}_\tau(p)$. Furthermore, we assume he second parameter $s_\tau(p)$ of the lognormal distribution (determining spread and skew) is a function of mean travel time, that is

$$s_\tau(p) = h\left(\widehat{\mu}_\tau(p)\right)$$

Since the distribution is wider and more skewed in congested conditions compared to free-flow conditions, we expect $h\left(\widehat{\mu}_\tau(p)\right)$ to be an increasing function of $\widehat{\mu}_\tau(p)$. On the 8 data sets reserved for SSNN training (see section 5.5) we find the following LS fit for a linear model:

$$s_\tau(p) = 0.12 + 10^{-4}\widehat{\mu}_\tau(p) \tag{7.4}$$

7.3.2 Uncertainty due to offline travel time estimation procedure

Since the SSNN model is trained with offline estimated travel times (with the PLSB trajectory method) for targets rather than real travel times, the errors in its predictions are also related to the errors the estimation procedure makes. Let

$$\tau_{est}(p) = PLSB\left(\mathbf{u}(p), \ldots, \mathbf{u}\left(p + \text{CEIL}\left[\tau_{est}(p)\right]\right)\right) \tag{7.5}$$

denote the PLSB procedure, which takes as inputs vectors of measurements along r (in this case just speeds) during the estimated travel time. The time period[4] $\left[p + \text{CEIL}\tau_{est}(p)\right]$ is the period in which the PLSB estimated vehicle trajectory exits the last section along the route. We can also cast this procedure as a regression problem by letting

$$\tau(p) = \tau_{est}(p) + e_P(p) + e_\tau(p) \Leftrightarrow \tag{7.6}$$

$$\widehat{\mu}_\tau(p) = \tau_{est}(p) + e_P(p) \tag{7.7}$$

The residuals $e_P(p)$ reflect the estimation errors made by the PLSB procedure. Eqn 7.6 states that there exist extra error bars around the mean travel time which the SSNN model predicts due to errors in the PLSB procedure. Similar to above we investigate the nature of this extra error term. Since these errors can be negative, the normal distribution is an appropriate candidate. The error term then is denoted as

$$e_P(p) \sim N\left(\mu_P(p), \sigma_P(p)\right) \tag{7.8}$$

[4]The operator CEIL rounds off in the direction of (minus) infinity, e.g. $\text{CEIL}(1.2) = 2$.

In which $\mu_P(p)$ denotes the mean estimation error (which does not necessarily equal zero) and $\sigma_P(p)$ the standard deviation of the estimation errors. First, fig. 7.4 and 7.5 show histograms and estimated normal distributions for the PLSB estimation errors under free-flow and congested traffic conditions respectively during test data set 2 (recurrent congestion). Again it appears that $\sigma_P(p)$ increases with mean (estimated) travel time, but also bias ($\mu_P(p)$, i.e. mean estimation error) is a function of mean (estimated) travel time. In severely congested conditions (mean speed on the main carriage way < 50 km/h), the PLSB estimates are biased in the negative direction, that is, mean travel time is underestimated almost one minute (equals on the average 5%), while in free-flow conditions the residuals are almost zero mean. Based on 7.4 and 7.5 we argue normality is a reasonable (and convenient) assumption, although for both cases, a KS test rejects normality. Although we might find a closer fit to the histograms using different distributions under different traffic conditions, we argue as above the main purpose here is to demonstrate a method for obtaining error bars around the SSNN prediction rather than accurately reproduce the distribution of PLSB estimation errors.

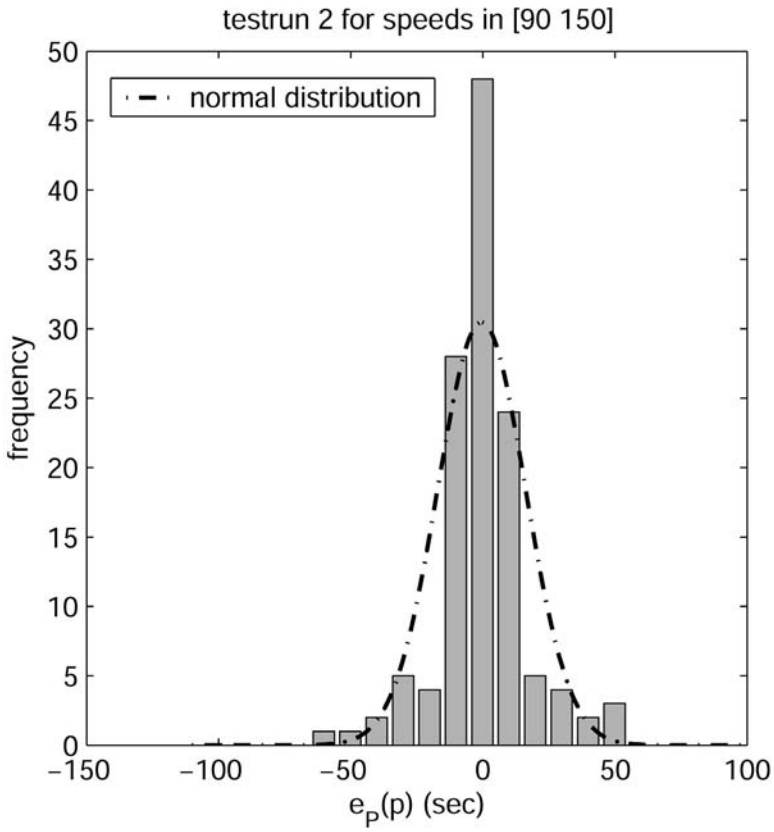


Figure 7.4: Histogram and estimated normal distribution of PLSB travel time estimation errors under freeflow traffic conditions (mean speed on main carriageway >90 km/h).

Finally, in fig. 7.6 it is shown that both PLSB estimation bias and standard error can be

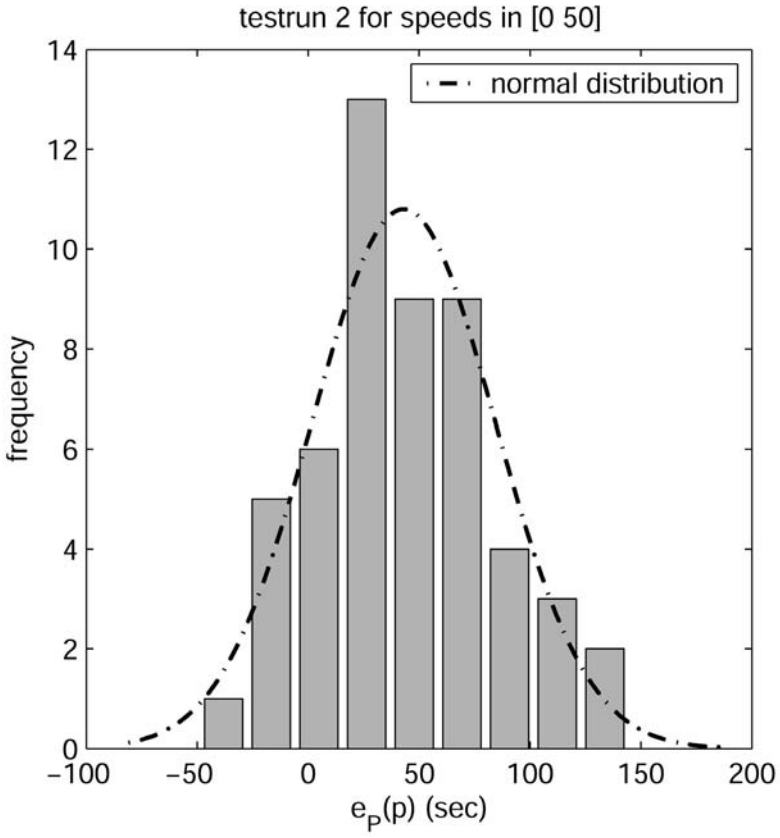


Figure 7.5: Histogram and estimated normal distribution of PLSB travel time estimation errors under congested traffic conditions (mean speed on main carriageway < 50 km/h).

considered a function of mean (estimated) travel time. As mean travel time increases so does the estimation error, both structurally (a bias toward underestimation) as well as randomly. On the basis of our data we find the following two linear relationships:

$$\widehat{\mu}_P(p) = -12 + 0.07\,\tau_{est}(p) \tag{7.9}$$

$$\widehat{\sigma}_P(p) = 7.5 + 0.05\,\tau_{est}(p) \tag{7.10}$$

As a result, the travel time prediction problem can now be rewritten as

$$\tau(p) = G(\mathbf{u}(p-1), \psi) + e_\tau(p) + e_P(p) \tag{7.11}$$

where the two identified disturbance terms (due to the travel time distribution and the PLSB estimation procedure respectively) are considered independent. Note that the bias due to the PLSB method does not contribute to the uncertainty, rather it systematically corrects the SSNN prediction in congested conditions, where it is been trained (with PLSB estimated travel times) to underestimate travel time.

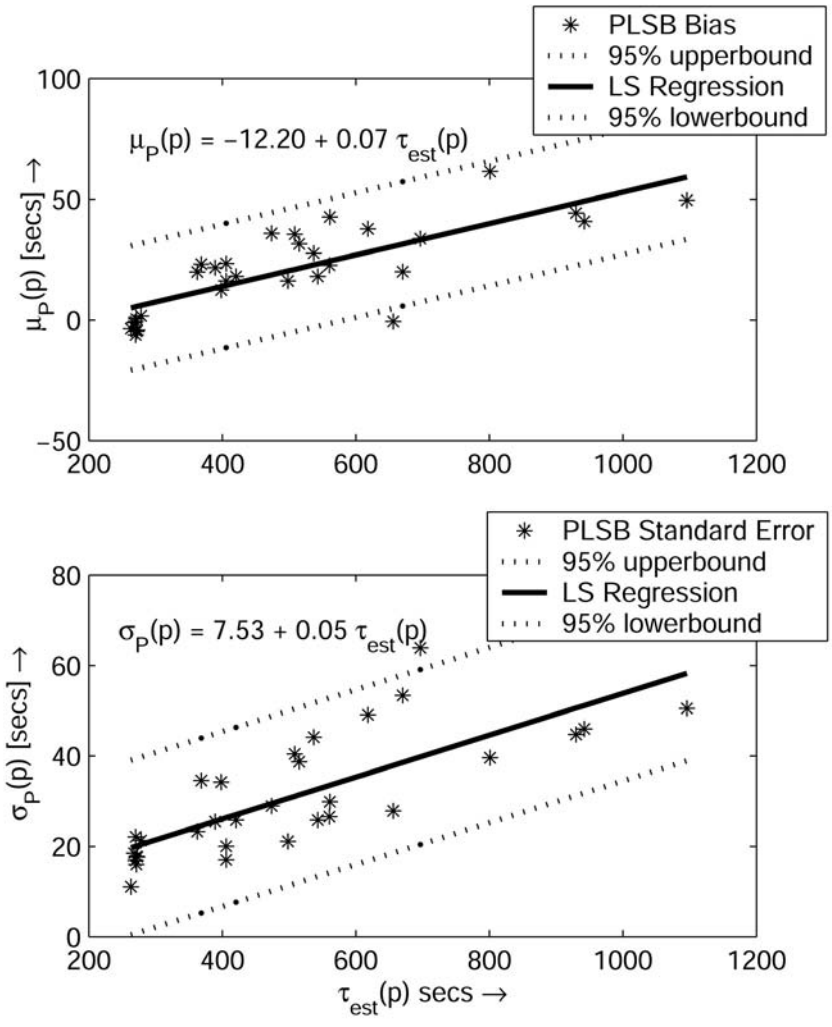


Figure 7.6: Bias and standard error of PLSB travel time estimation procedure. Both can be conceived a (linear) function of mean estimated travel time.

7.3.3 Uncertainty due to the parameters of the SSNN

As (MacKay 1995) convincingly argues a neural network (in fact any non-linear parameterized regression function) should be considered a probabilistic model and its calibration (training) as probabilistic inference[5]. The result of training then is not one most likely parameter vector that fits the training data best, but a region of high probability density in parameter space, that gives rise to the training data best. Practically this means putting error bars on the parameters, which in turn result in error bars on each neural network prediction. If all sources of uncertainty (travel time (data) and model) are assumed independent (in all time periods), equation (7.11) becomes

$$\tau(p) = G(\mathbf{u}(p-1), \psi) + e_{\psi}(p) + e_{\tau}(p) + e_{P}(p) \tag{7.12}$$

[5]More details and background, which underly the concepts presented here can be found in Appendix C.

in which the extra noise term $e_\psi(p)$ depicts the uncertainty due to the SSNN parameters. Error bars on each prediction are hence the sum of noise due to model + noise in data (due to the distribution of travel time and the PLSB estimation method). In (Heskes 1997) and in the next section it is assumed that within a period the model uncertainty $e_\psi(p)$ is zero mean normally distributed with variance $\sigma^2_\psi(p)$. This component is derived in the next section (7.4) by exploiting the Hessian matrix of the performance function with respect to the SSNN parameters. In the next chapter (section 8.2.4) we will use an alternative and computationally more efficient method to do the same thing in the case of real-time data. Taking into account model uncertainty $\sigma^2_\psi(p)$ only yields confidence intervals, that is

$$G(\mathbf{u}(p-1), \psi) \pm c \times \sigma_\psi(p) \tag{7.13}$$

in which c denotes the appropriate value from a Student-t distribution or - since the SSNN has so many degrees of freedom - the nominal coverage of the normal distribution at the desired level of significance. Note that these confidence intervals reflect "the difficulty" the SSNN model has in predicting mean (!) travel times in a particular situation. It literally has nothing to do with the distribution of travel time, since the SSNN has been trained to reproduce mean travel times only. If an input pattern (measured speeds and flows) is considered ambiguous (in that it may be associated with more than one travel time) or if it comes from a region in input space with which the SSNN is not so familiar (because it was underrepresented in its training data, or because it is severely corrupted due to input failure), confidence intervals will be larger, which means the model prediction is more uncertain.

7.3.4 The total predictive distribution

Assuming independent sources of uncertainty we adopt an additive model for the total predictive distribution (eqn 7.12). This distribution has a mean value of $G(\mathbf{u}(p-1), \psi)$ and variance equal to the sum of variances from the error components $e_\psi(p) + e_\tau(p) + e_P(p)$, that is

$$\sigma^2(p) = \sigma^2_\psi(p) + \sigma^2_\tau(p) + \sigma^2_P(p) \tag{7.14}$$

We assume that the total predictive distribution is lognormal[6] with parameters M and S. Mean μ and variance σ^2 of a lognormal distribution relate to these parameters as follows (MathWorld 2004)

$$\begin{aligned} \mu &= e^{M+S^2/2} \\ \sigma^2 &= e^{2M+S^2}\left(e^{S^2}-1\right) \end{aligned}$$

[6]This assumption is based on the fact that the largest part of the predictive distribution is due to the lognormal (very flat and skewed) distribution of travel time. Again, the width of this distribution is not very realistic compared to real travel time distributions.

Given eqn (7.14) and $\mu(p) = G(\mathbf{u}(p-1), \psi)$ this yields

$$\begin{aligned} M(p) &= \ln(\mu(p)) - S^2/2 \\ S^2(p) &= \ln\left(\frac{\sigma^2(p)}{\mu^2(p)} + 1\right) \end{aligned}$$

Note that inherently, these parameters are time dependent. Taking into account all uncertainty components yields prediction intervals given by

$$\left[G(\mathbf{u}(p-1), \psi) - c^- \times \sigma(p), G(\mathbf{u}(p-1), \psi) + c^+ \times \sigma(p)\right] \tag{7.15}$$

in which $\sigma(p)$ is the standard deviation of the total predictive distribution. Since this distribution is not symmetrical, the prediction interval is also not symmetrical. The statistics c^+ and c^- could be derived from a student t distribution, but we will (for each prediction) derive them directly from the estimated lognormal distribution. The prediction intervals hence reflect the nominal coverage of that distribution, which is given the large amounts of individual observations (travel times) per period p a reasonable approach. Note that as mean travel time (produced by the SSNN) is far of, the predictive distribution clearly is also far off from the actual travel time distribution.

7.4 Confidence Estimation for Neural Networks (I)

In the past decade, a number of methods to estimate confidence measures for neural networks have been developed. These include Maximum Likelihood techniques, Bayesian techniques and techniques based on bootstrapping (see (Papadopoulos et al. 2001) for an overview and comparison). In this chapter we will exploit the Bayesian Approach (based on (Papadopoulos et al. 2001)), in which we explicitly use the information on the posterior distribution of weights we obtain during training (see section 5.4.3 and appendix C). In the next chapter (section 8.2.4) we will use random subsampling as an alternative method for obtaining model confidence. The reasons for this are the much more efficient way in which this methods allow us to deal with large amounts of training data, then the Bayesian method applied here.

Recall equation (5.15) in section 5.4.3, where we assume a Gaussian posterior for the weights $N(\psi^{MP}, \Sigma_\psi)$ after training. Confidence levels on the prediction of a new datum can then be obtained by local linearization around the output (MacKay 1995):

$$y(p) = G(\mathbf{u}(p-1), \psi) \simeq G(\mathbf{u}(p-1), \psi^{MP}) + \mathbf{g}(\psi - \psi^{MP}) \tag{7.16}$$

in which $\mathbf{g}$ is the sensitivity of the output to the parameters, that is, the first derivative of the neural network output with respect to its weights $\frac{dG}{d\psi}$, and ψ^{MP} denotes the

maximum probable parameter vector after training. The predictive distribution (disregarding noise in the target data for the moment) then becomes a Gaussian integral with mean $G(\mathbf{u}(p-1), \psi^{MP})$ and variance

$$\sigma_{\psi}^{2}(p) = \mathbf{g}^{T}(p)\widehat{\mathbf{H}}^{-1}\mathbf{g}(p) \quad (7.17)$$

$$\mathbf{g}(p) = \frac{\partial y(p)}{\partial \psi}|\mathbf{u}(p-1), \psi, \alpha, \beta \quad (7.18)$$

in which the Hessian matrix $\widehat{\mathbf{H}}$ (second derivative of the SSNN performance function with respect to its parameters) is calculated during SSNN training, and is hence obtained automatically (see eqn 5.10). The output sensitivities $\mathbf{g}(p)$ can be calculated similarly to the Jacobian matrix (first derivative) of output errors with respect to the neural network weights during training. The only difference is that we do not feed the output error $e(p) = y(p) - o(p)$ back into the network, but rather just the output $y(p)$. For a detailed derivation of this backpropagation algorithm refer to appendix B section B.3.1.

7.5 Experimental setup

In the previous sections we have derived all the ingredients for predicting not only mean travel time but also the uncertainty (in terms of variance) associated with that prediction. These include model confidence, variance due to the PLSB estimation procedure and the inherent distribution of travel time itself.

7.5.1 Data

We will use the same 7 data sets reserved for SSNN testing (see section 5.5) in the ensuing sections and 5 extra data sets generated with the same traffic demand pattern as test data sets 1 to 5 (recurrent congestion at Delft-Zuid) but in which we have simulated major roadworks on sections 2 up to 6 (see fig. 5.5 and table 5.1). As is done in the Dutch high situation VMS portals on those sections dynamically reduce (mandatory) speed-limits of 70 km/h. Obviously, this restrictive measure will yield different and for the SSNN model partially unknown traffic patterns.

As qualitative indicators of how well these statistical measures of uncertainty reflect the difficulty the SSNN framework has in predicting travel times on all available test data in different conditions we use the Confidence Interval and Prediction Interval Coverage Percentage index as defined in appendix A on page 257, which reflect the percentage of observations that fall in the confidence intervals (eqn 7.13) and prediction intervals respectively (eqn 7.15).

7.5.2 Scenarios

We study confidence and prediction intervals of the SSNN model under three different scenarios.

Scenario 1 Base case: 100% clean test data and an SSNN model trained with clean data.

Scenario 2 Missing data case, in which we calculate confidence and prediction intervals for three missing data problems from the previous chapter.

- **2a** The first incorporates test data with 20% incidental input failure, a preprocessing strategy (MA procedure) and an SSNN model trained with clean data.
- **2b** The second incorporates test data with 20% incidental input failure, no preprocessing strategy and an SSNN model trained with clean data.
- **2c** This scenario also incorporates test data with 20% incidental input failure, no preprocessing strategy but a "robust" SSNN model trained with 20% corrupted data.

Scenario 3 Similar to base case, but now we test the SSNN model on a data set in which dynamic speed-limits are applied due to large scale roadworks on sections 2 to 6.

We apriori expect the second and third scenario to yield larger confidence and prediction intervals then the base case scenario. In the second since we have tampered with the SSNN input data, and in the third because we present the SSNN model with data (traffic conditions) which may be (partially) outside its input domain.

7.6 Results

In this section we present both qualitative and quantitative results. The latter by means of CI_CP and PI_CP percentages on all test data sets, the former by means of graphs in which we construct confidence and prediction intervals that reflect a nominal coverage of 95% of the assumed distributions (Gaussian and Lognormal respectively).

7.6.1 Scenario 1: base case

In fig. 7.7 confidence and prediction intervals are calculated for test data set 2 (normal weekday congestion). Details on this traffic pattern can be found in section 5.5. The first observation to be made fig. 7.7 (bottom) is that the predictive distribution

(95% prediction intervals) indeed widens as mean travel time increases, which is in line with the simulated travel times. The second observation is that the estimated log-normal predictive distribution is wider than the actual travel time distribution. This off coarse makes sense, since it is constituted not only by assumptions on the actual travel time distribution, but also on the PLSB estimation errors and SSNN confidence. fig. 7.7 shows that the uncertainty due to the travel time distribution is the largest of the three components, followed by the PLSB estimation errors. Both are an order larger than the model confidence component during all traffic conditions. Nonetheless, all components increase as mean travel time increases as shown in fig. 7.8, in which the contribution of each of the three components (standard deviations) is plotted against mean predicted travel time.

Finally, from fig. 7.7 (left of both top and bottom graph) observe that the SSNN model requires some time steps to adjust. This is due to the recurrent connections of the internal states and the context layer, which we initialized with zeros in this particular case. A better option is to initialize the context layer with the mean internal states during free-flow conditions.

7.6.2 Scenario 2: missing data

In this subsection we return to the missing data problem. The previous chapter showed that the SSNN can be trained to deal with missing data itself by including these in the training data set or be fed with data cleaned by simple preprocessing algorithms.

Scenario 2a: missing data and (MA) preprocessing method

In this case input failure is tackled by means of an MA filter (eqn 6.4 on page 153). The results are shown in fig. 7.9. It appears that not only the mean results are still very good (recall table 6.11 on page 155 of the previous chapter), also confidence intervals are very similar to the ones for scenario 1 (clean data). The "damage done" by the MA procedure does not cause the SSNN model to either produce inaccurate results nor does it significantly increase the associated confidence levels.

Scenario 2b: missing data and a "do nothing" strategy

In this case input failure is not tackled at all, instead, each missing datum is replaced with a null value (0.5 - see previous chapter). The results are shown in fig. 7.11. Clearly, the SSNN behavior is erratic both in terms of its mean predictions as well as the model confidence component. Fig. 7.12 shows how model confidence behaves erratic as a function of mean (predicted) travel time, with peaks that are due to missing data. In other words, as missing data percentages are high, the SSNN confidence is low (its confidence intervals are large), which is exactly what we expected. Although we

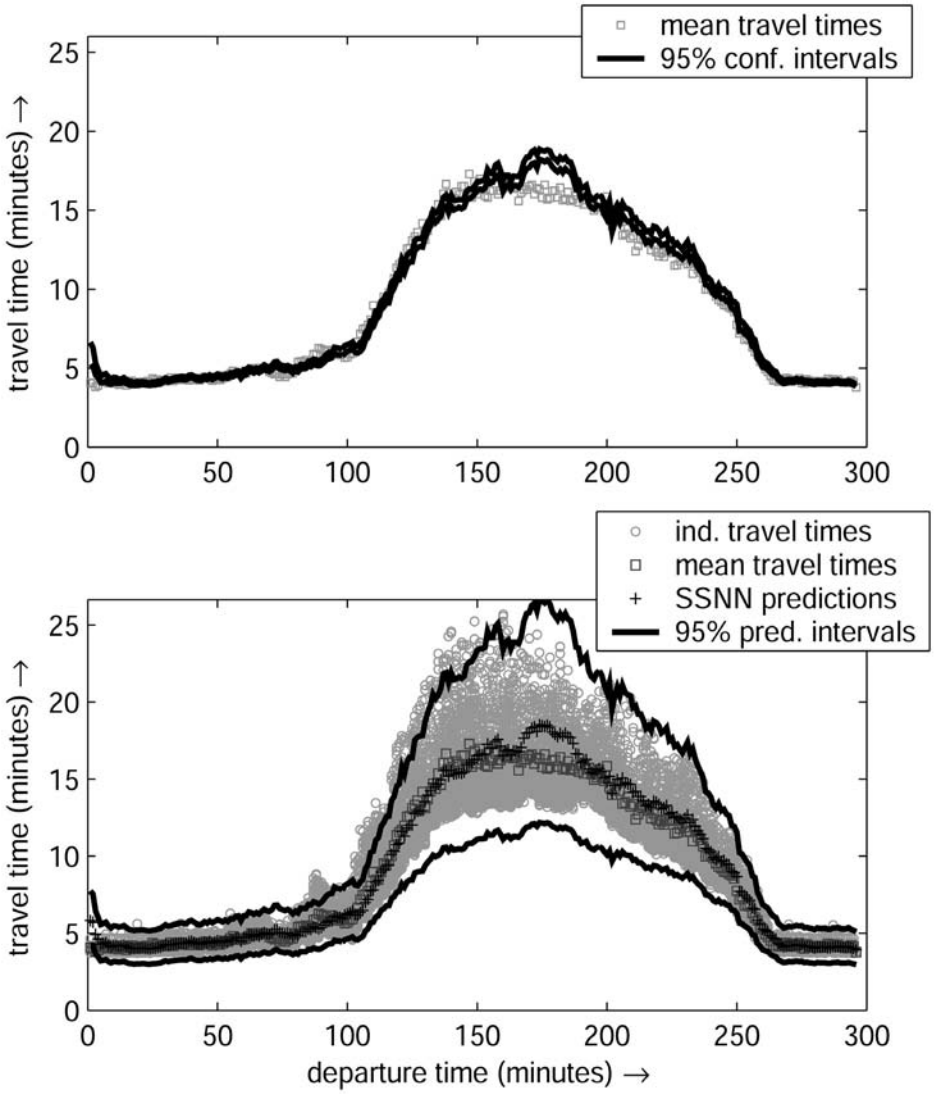


Figure 7.7: Confidence (top) and prediction intervals (bottom) for Scenario 1.

already concluded that this strategy for dealing with missing data does not suffice, the results underline that model confidence is closely related to whether or not the input data are in the input domain of the SSNN.

Scenario 2c: missing data and a "robust" SSNN model

In this case input failure is tackled by means of an SSNN model trained with (20%) missing data, a strategy also discussed in the previous chapter. The results are shown in fig. 7.13. Note that in this case the SSNN tends to overestimate travel times in congested conditions. An artefact of this structural overestimation is that the coverage percentage of the prediction intervals (fig. 7.13 (bottom)) is very high. Furthermore, a slight increase in the model confidence component can be observed from fig. 7.14, in comparison to Scenarios 1 and 2a. Also, the variation in the model confidence

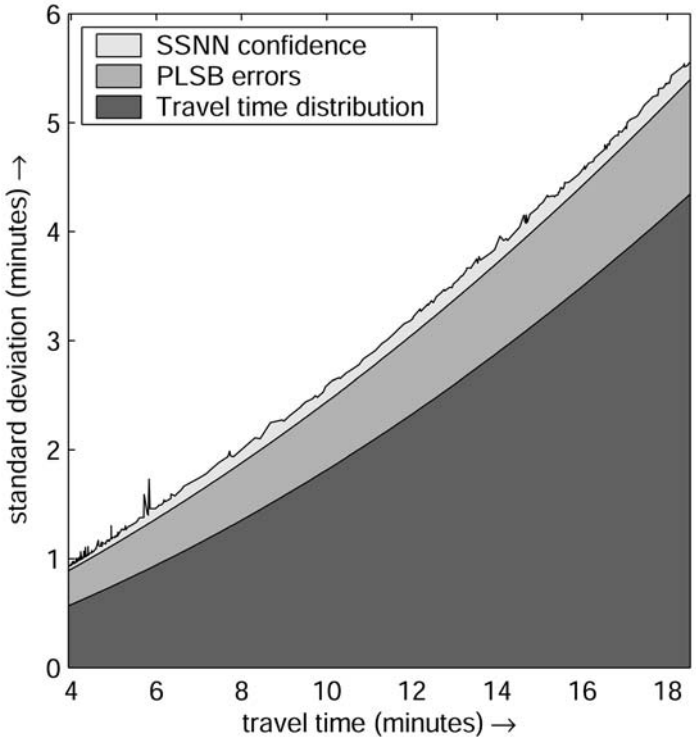


Figure 7.8: Contribution of each of the uncertainty components as a function of mean (predicted) travel time in Scenario 1.

component is slightly higher as can be observed from fig. 7.14. As was concluded in the previous chapter, this strategy does increase robustness to missing data but at the cost of its predictive performance and also - shown here - of model confidence.

7.6.3 Scenario 3: unknown traffic conditions

Finally, in scenario 3 we present the SSNN model with a traffic situation it - at least partially - is unfamiliar with. Due to roadworks, a dynamic speed-limit (of 70 km/h) is applied on sections 2 to 6. Fig. 7.15 shows this not only deteriorates the predictive performance, but also significantly increases the associated confidence levels. As the SSNN model has difficulty predicting the onset of congestion (fig. 7.15 time instants 100-130) its confidence intervals significantly increase, which indicates increased uncertainty. Also fig. 7.16 shows the model confidence component is significantly larger in this scenario as compared to scenarios 1 and 2a. The conclusion is that also in cases where data are not missing but for other reasons outside the input domain, confidence levels go up, providing a clear mechanism for detecting potential problems.

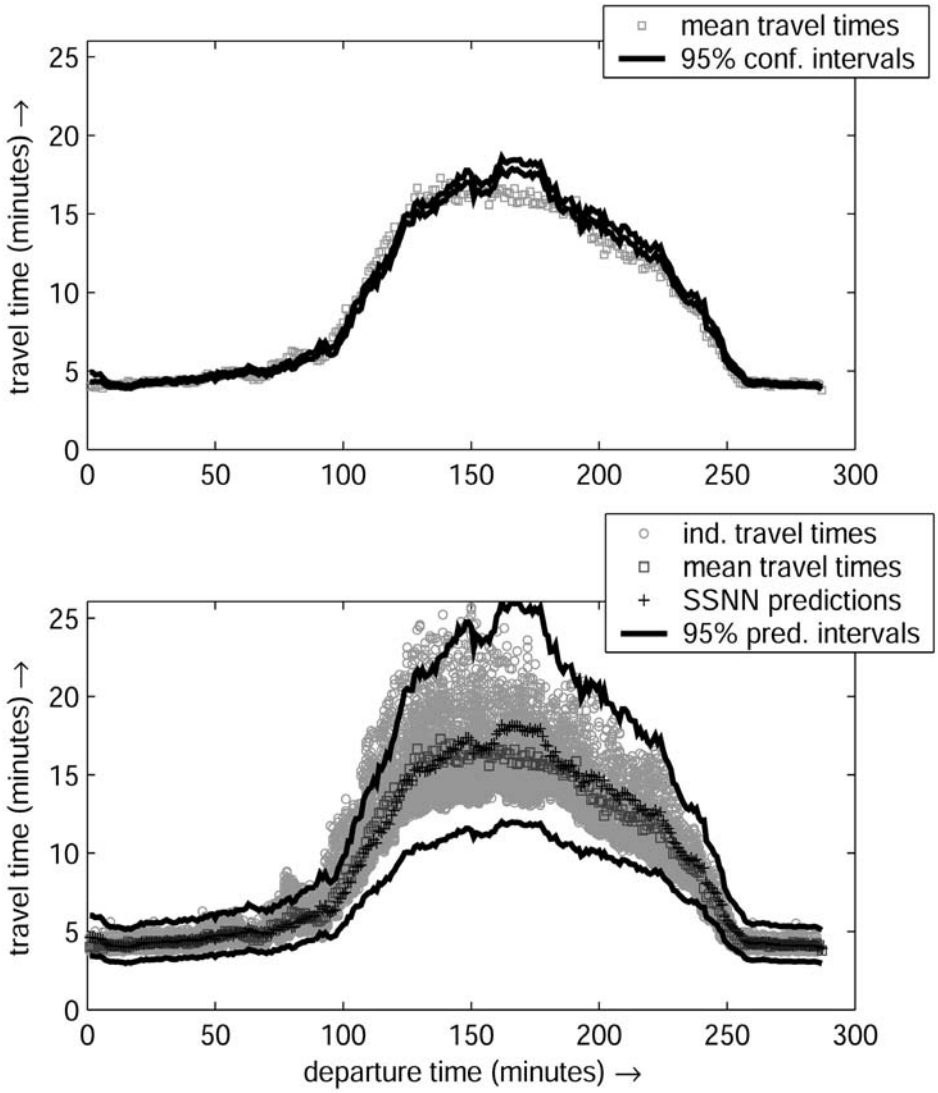


Figure 7.9: Confidence (top) and prediction intervals (bottom) for scenario 2a.

7.6.4 Quantitative results

In this final subsection we present some quantitative results in the form of CI_CP and PI_CP values of all scenarios over all (for each scenario) available test data covering 95% of the associated distribution. Confidence intervals are considered the result of a Gaussian distribution and prediction intervals are constructed with a lognormal distribution. Table 7.1 shows for each of the five scenarios the percentage of mean travel times that fall in the confidence intervals. Although quantitatively, these results are disappointing, we argue they are misleading, since they are based on the notion that the SSNN is an unbiased predictor. For example they provoke one to conclude that the SSNN model performs better in scenarios 2c and 3 than in scenarios 1 and 2a. In the former scenarios the confidence intervals cover more observations than in the latter two. As pointed out in the qualitative analysis above, quite the contrary is in fact the

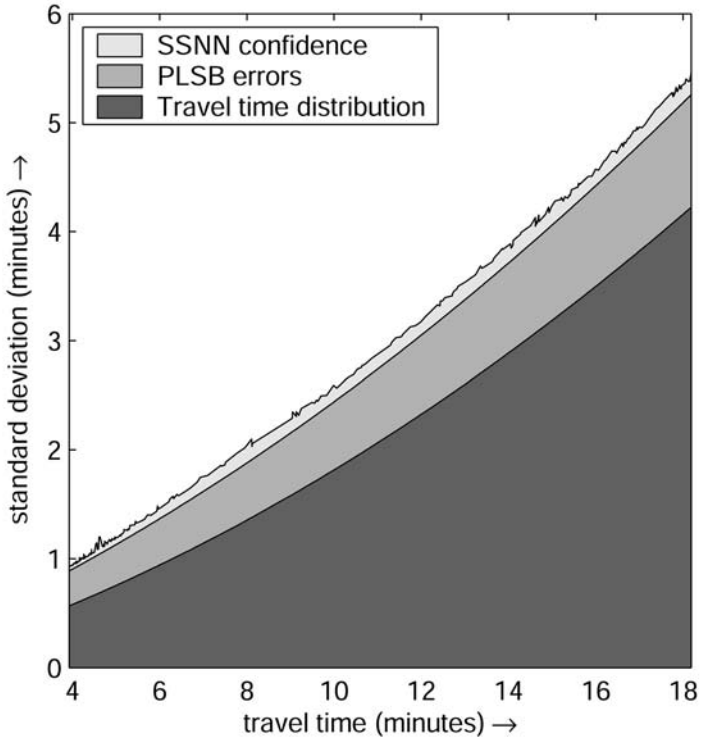


Figure 7.10: Contribution of each of the uncertainty components as a function of mean (predicted) travel time in Scenario 2b.

case; the SSNN performs significantly worse in scenarios 2c and 3 due to missing data and unknown traffic conditions respectively. Therefore, the main value of these confidence intervals is not in their (nominal) coverage of mean observations but in their quality as indicator for the amount of uncertainty associated with a particular prediction. In fact, large confidence intervals (leading to high CI_CP values) indicate large uncertainty rather than good performance.

Table 7.1: Confidence Interval Coverage Percentage (CI_CP) index for each of the five scenarios. This index reflects the percentage of *mean* travel times that fall within the prediction interval.

	1	2a	2b	2c	3
CI_CP (%)	22	21	4	35	30

Table 7.2 shows for each of the five scenarios the percentage of individual travel times that fall in the prediction intervals. Since the prediction intervals are constituted largely by our statistical assumptions of the distribution of travel times and the PLSB estimation errors, these results look much friendlier. Note that also prediction interval coverage is meaningful only in case of unbiased predictions, and provide hence little insight in how well the SSNN model is actually predicting travel time. Nonetheless, scenarios

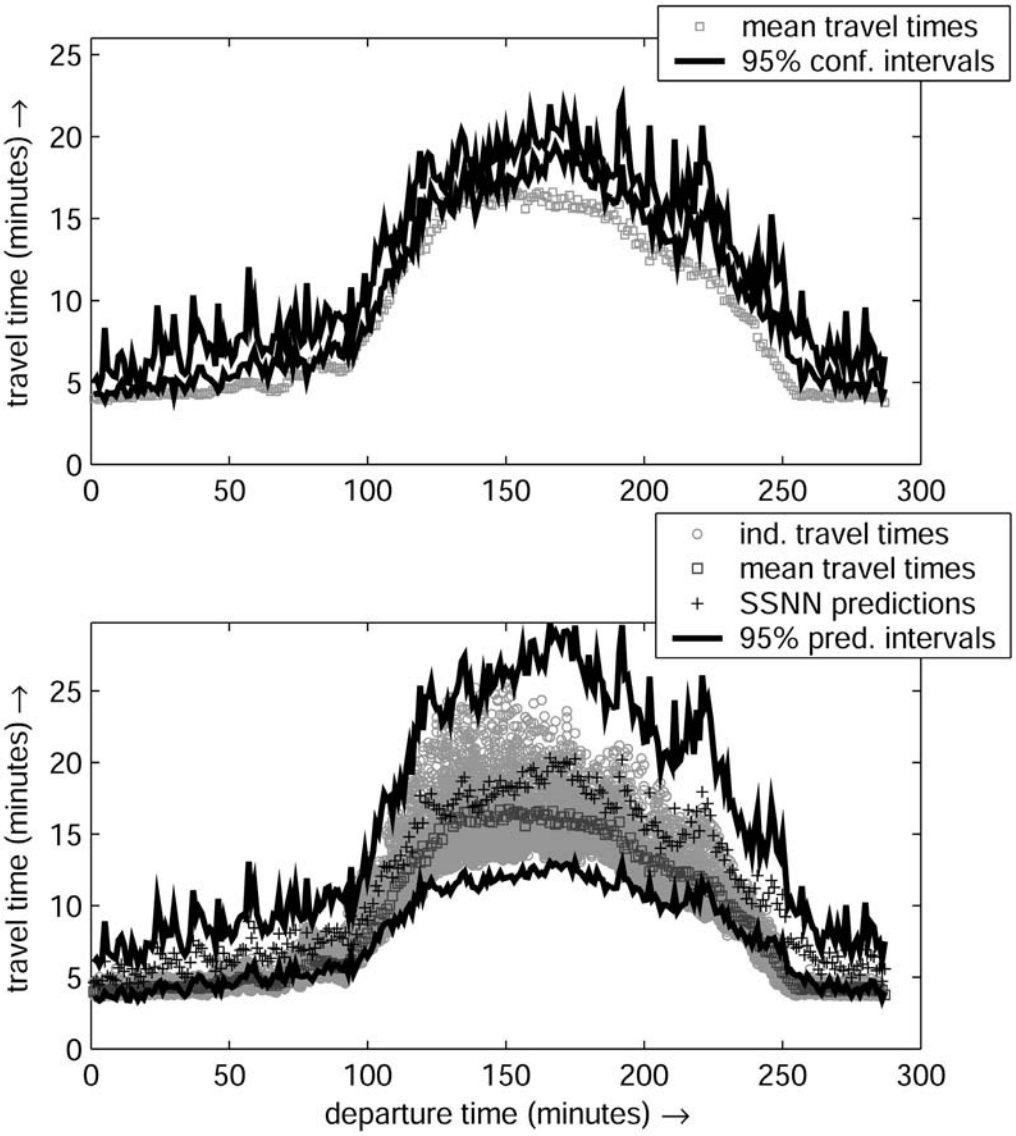


Figure 7.11: Confidence (top) and prediction intervals (bottom) for Scenario 2b.

1 and 2a score best in terms of the PI_CP index, while scenario 2b clearly produces the worst results. Note however that the 88% PI_CP for scenario 3 completely obscures the fact that in this case the SSNN makes some serious prediction errors.

7.7 Implications of Results

The most relevant and powerful result is that the model confidence component offers a quantitative indicator for the uncertainty associated with a particular travel time prediction, without having to measure actual travel times. In this sense large model uncertainty coincides with large prediction errors. As a quantitative indication the R^2 (squared correlation coefficient) value for the relation between the prediction error and

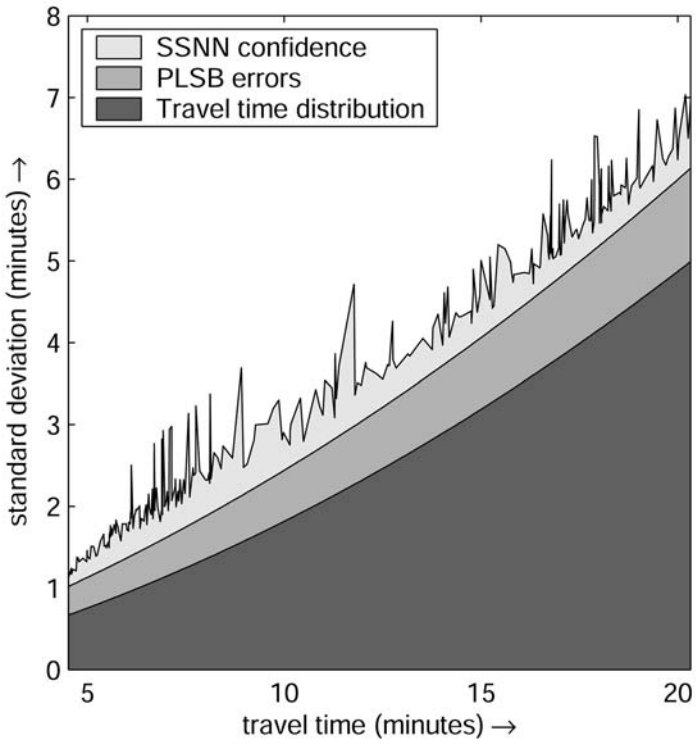


Figure 7.12: Contribution of each of the uncertainty components as a function of mean (predicted) travel time in Scenario 2b.

Table 7.2: Prediction Interval Coverage Percentage (PI_CP) index for each of the five scenarios. This index reflects the percentage of *individual* travel times that fall within the prediction interval.

	1	2a	2b	2c	3
PI_CP (%)	96	95	66	93	88

the width of the confidence interval over all scenarios equals 0.66, which indicates a fairly strong positive statistical relation.

As a general rule, the SSNN confidence intervals grow wider in case input data are (partially) outside its input-domain. In real-time operation, monitoring of these confidence intervals allows the traffic manager to track the quality of both the SSNN model as well as the data with which it is fed. In case of an increase of confidence intervals a traffic manager might run through the following checklist

1. Something is "wrong" with the input data:

   (a) one or more detectors are failing due to damage, power failure or other causes

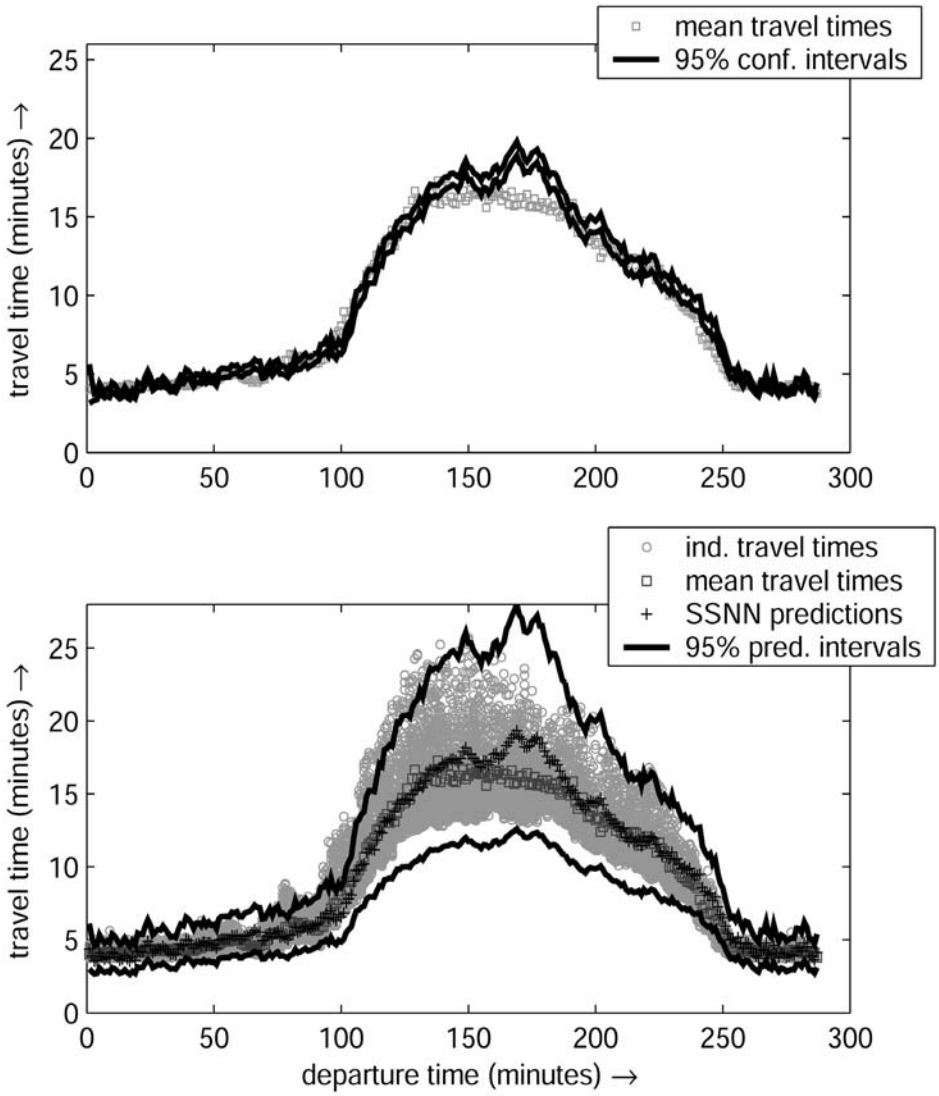


Figure 7.13: Confidence (top) and prediction intervals (bottom) for scenario 2c.

(b) the preprocessing layer is failing (due to software or hardware errors)

(c) other software or hardware problems in the information chain feeding the SSNN model

2. The SSNN model is unfamiliar with current traffic conditions (needs retraining):

(a) the route of interest has geometrically changed (number of lanes, new or missing on and off ramps, etc.)

(b) some traffic control measure causes new traffic patterns (permanent lane closures, speed-limits, etc.)

(c) some dynamic measure causes new traffic patterns (temporary lane closures, speed-limits, metering, etc.)

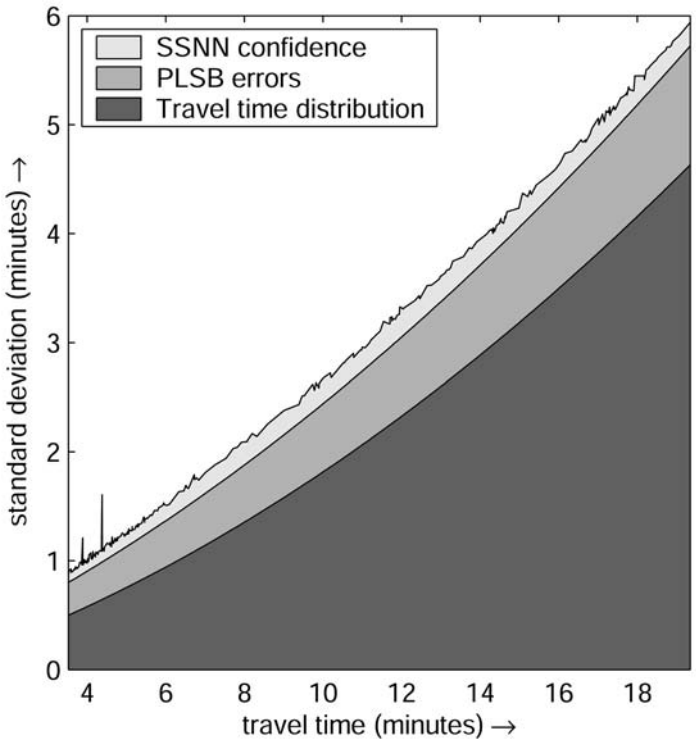


Figure 7.14: Contribution of each of the uncertainty components as a function of mean (predicted) travel time in Scenario 2c.

Since large model uncertainty coincides with large prediction errors, the confidence intervals do not reflect the upper and lower bounds to these predictions.

Prediction intervals encompass confidence intervals and are furthermore constituted by our distributional assumptions of travel times and the PLSB estimation errors. Since the travel time distributions of the simulated data are so wide the coverage percentages of these prediction intervals are reasonably good, even in case the (mean) prediction is far off. Another consequence of the very wide simulated travel time distributions is the following. Obviously, a prediction interval of ± 7 minutes around a mean prediction of 17 minutes is too large in a practical situation. The consequence would be that we would expect individual travel times in between 10 and 24 minutes, which is probably a more uncertain estimate than one would get from a historical profile. Since we expect real travel times to come from much smaller distributions, we can also expect more meaningful prediction intervals in that case.

Finally, the high coverage percentages of the (unrealistically wide) prediction intervals presented in the previous sections underline that quantitative analysis in terms of coverage percentages on all test data may obscure important trends or facts. A good example of the latter is the increase of CI_CP (confidence interval coverage percentage) values in scenarios 2c and 3, while in fact in both these scenarios the SSNN performance (in terms of the mean) is significantly worse than in scenarios 1 and 2a.

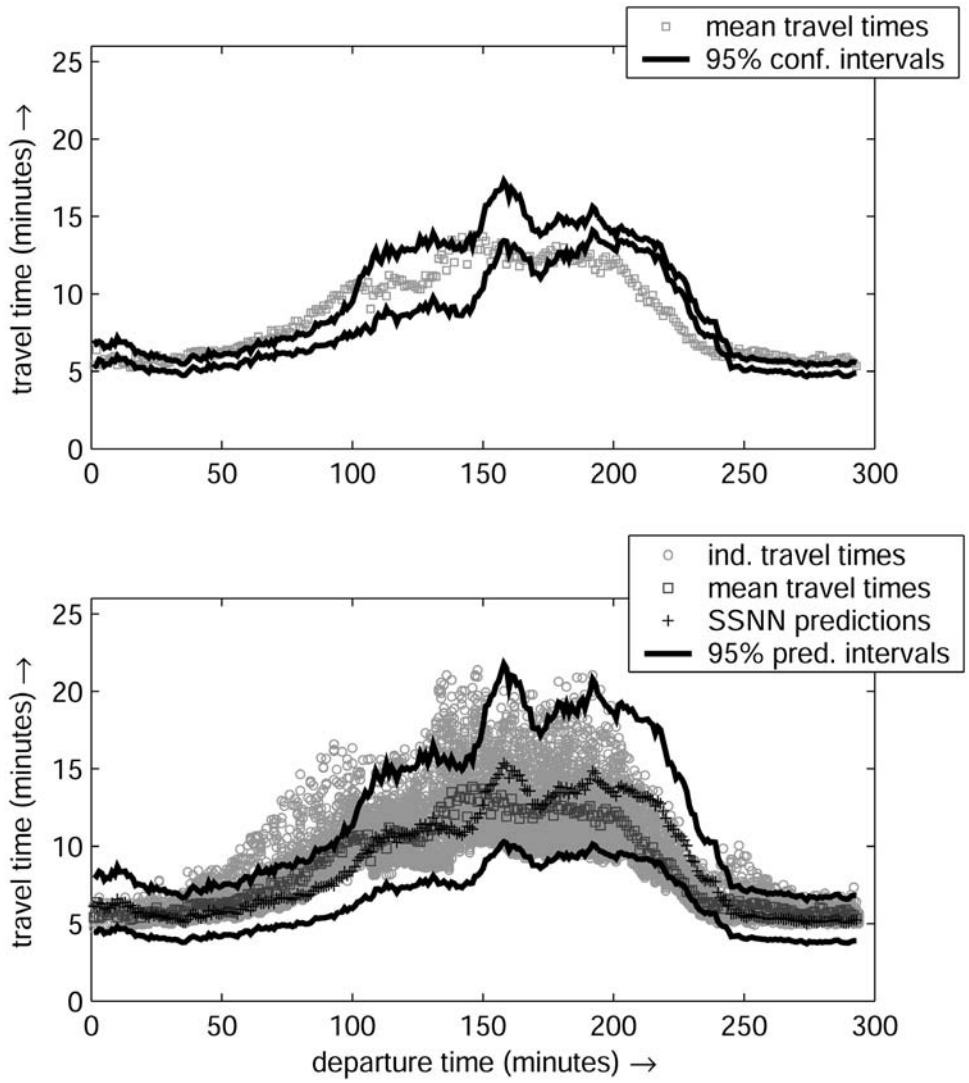


Figure 7.15: Confidence (top) and prediction intervals (bottom) for scenario 3.

7.8 Summary

In this chapter we presented methods to quantify (in a statistical sense) the uncertainty associated with the short term freeway travel time prediction framework developed in previous chapters. We identified three sources that contribute to that uncertainty. The first is due to the parameters of the SSNN travel time prediction model. The second and third depict the uncertainty due to the fact we train the SSNN with estimated rather than real travel times and the distribution of travel times itself. The first uncertainty component provides us with confidence intervals reflecting the 'difficulty' the SSNN has to predict travel time in a particular situation, while all components together give rise to prediction intervals which reflect the (predictive) distribution of travel times.

Model confidence is obtained automatically by using the Bayesian method for SSNN training. The main value of these confidence intervals is not in their (nominal) cover-

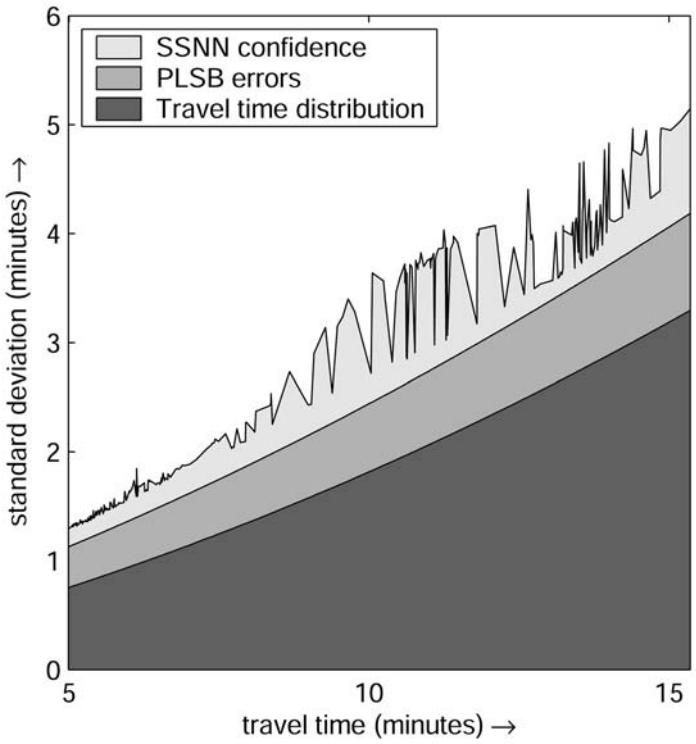


Figure 7.16: Contribution of each of the uncertainty components as a function of mean (predicted) travel time in Scenario 3.

age of mean observations but in their quality as indicator for the amount of uncertainty associated with a particular prediction. Uncertainty in this sense reflects the magnitude of the prediction error. We showed that in case of missing data, but also in case the SSNN is confronted with traffic conditions that are partially outside its input-domain, confidence intervals grow larger, which implies the SSNN is less certain of its predictions. In these cases, indeed prediction errors are large. In real-time operation, monitoring of these confidence intervals allows the traffic manager to track the quality of both the SSNN model as well as the data with which it is fed.

In the next chapter we will apply the tools and techniques presented here to real data. Although it will become clear that real travel times have rather different distributional properties than the simulated data used up to here, the methods and also conclusions presented in this chapter do apply in a real situation.

Chapter 8

Real-time Application of the SSNN Freeway Travel Time Prediction Framework

8.1 Introduction

In this chapter we put the results found in chapters 3, 5, 6, and 7 to practice. The Regiolab Delft project (appendix E or e.g. (Van Zuylen & Muller 2002)) provides the ideal test bed for such an application. Currently, the Regiolab Delft project is a public-private partnership in which the Delft University of Technology closely cooperates with the Test Center, Traffic Research Department and Directorate South-Holland of the Dutch Ministry of Transport, Public Works and Water Management, the province of South-Holland, the municipality of Delft and business partners Vialis and Siemens. Within this project detailed traffic data are collected on a real-time basis from a large part of the road-network (both freeway as well as provincial and urban roads) covering the southwest of The Netherlands, including the A13 freeway corridor connecting The Hague and Rotterdam (see fig. 8.1), which was used as a blueprint for the experiments based on synthetic data described in earlier chapters.

The 13 kilometer A13 Southbound (fig. 8.3) is one of the most densely used freeway corridors in the Netherlands with recurrent congestion every weekday afternoon peak and a average daily traffic load of 150,000 vehicles. Also this particular road stretch is in the top 10 roads in the Netherlands where the most severe congestion occurs. In 2001 there were over five hundred traffic jams (occurring on more than one location) with an average length of 3.7 kilometers and an average duration of one hour and 7 minutes (AVV 2002). However, regularly, the entire stretch is congested (13 kilometers) for several hours in the afternoon peak hour[1].

[1] personal observation and experience

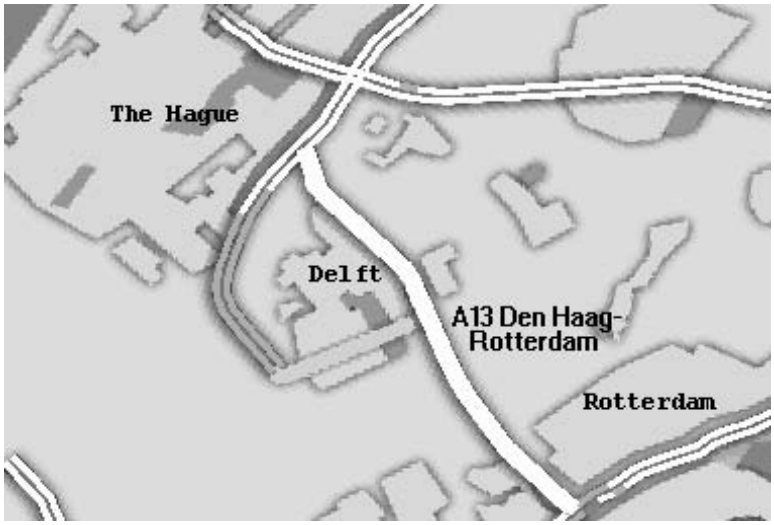


Figure 8.1: The road-network covered by the Regio-Lab Delft project, The Netherlands. The little squares on the roads depict detector locations.

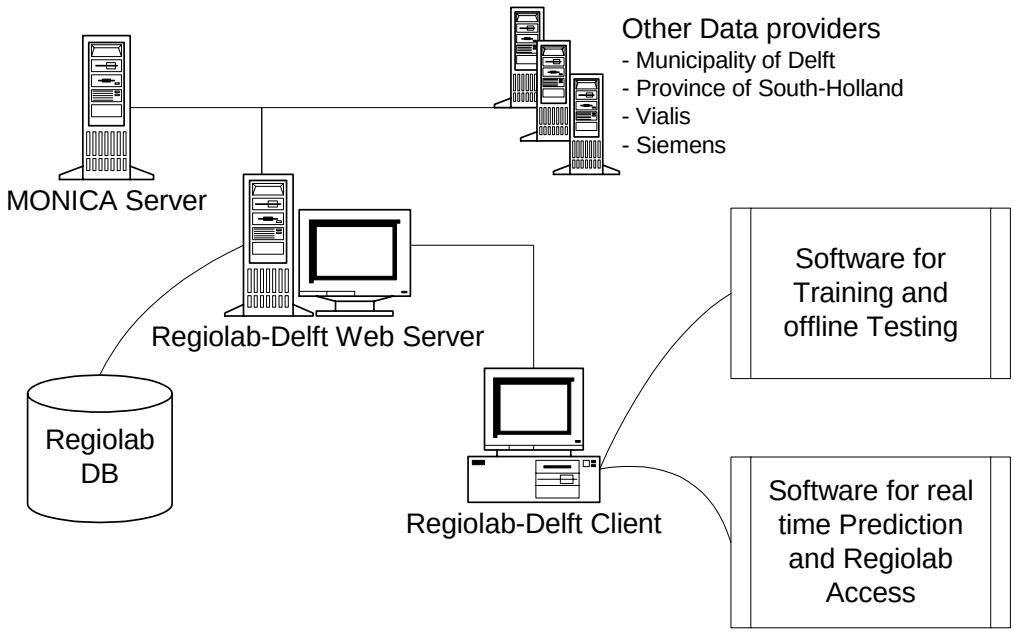


Figure 8.2: Functional architecture of travel time prediction framework applied as regiolab webclient.

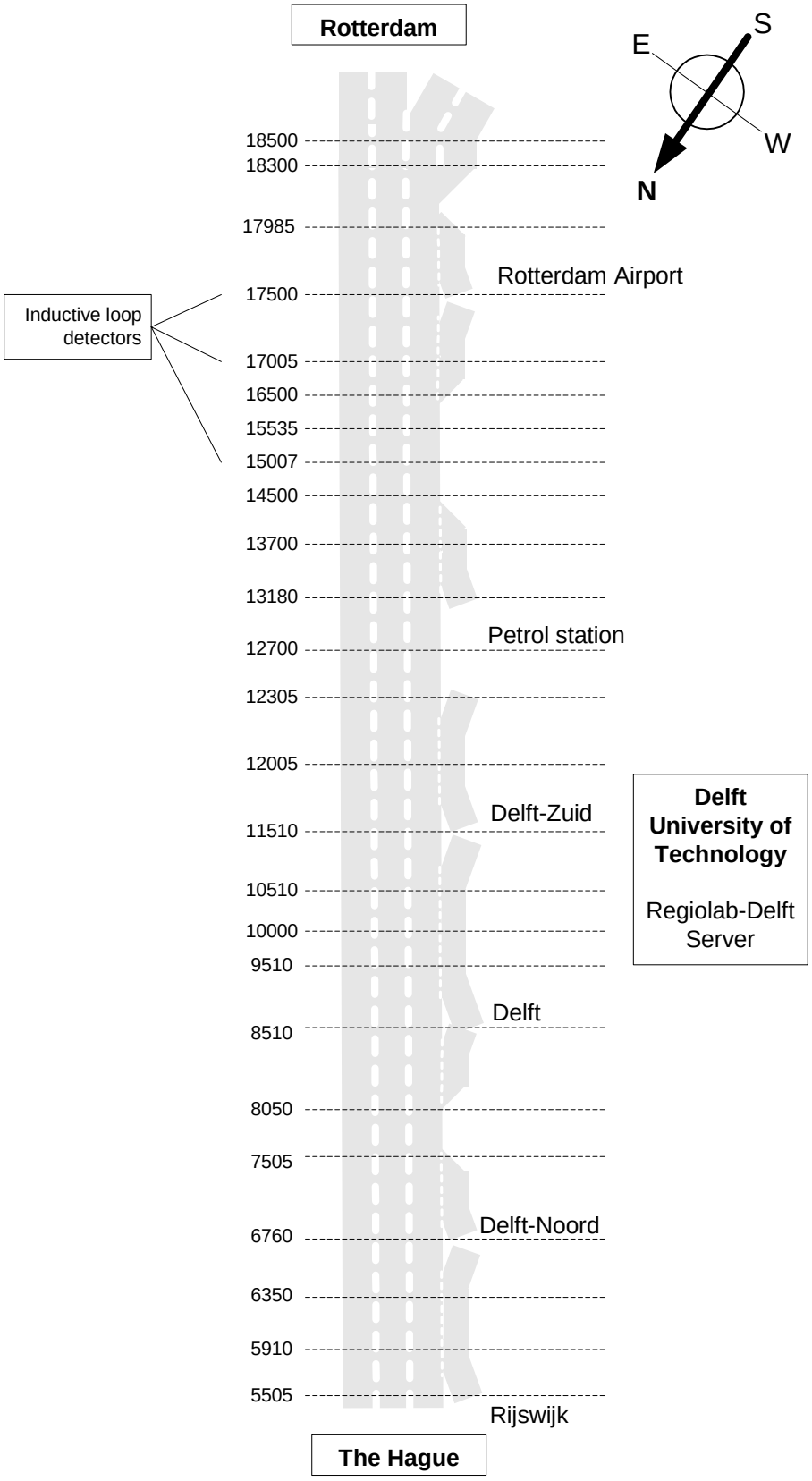


Figure 8.3: A13 Southbound freeway stretch between the Hague and Rotterdam. Note that for readability purposes this schematic view is not drawn to scale.

8.2 The SSNN Travel Time Prediction Framework

8.2.1 Functional Architecture

The freeway data-collection system MONICA provides one minute aggregated traffic flows and one minute averaged speeds from inductive loop detectors which are located about every 500 meters. Every minute all data are uploaded to the Regiolab Delft Server, which compresses and stores the data and subsequently makes it available through a webservice (Van Zuylen & Muller 2002).The travel time prediction framework (fig. 2.8 on page 34) as used throughout the previous chapters is applied in this case as a webclient application that uses this Regiolab Delft webservice (fig. 8.2). Note that separate software tools have been developed for training and offline testing, and real-time use of the SSNN framework. The former are written in Matlab, the latter in Delphi (Enterprise v6.0) and was developed as a demonstration tool, which runs on any PC which has an internet connection and permission to access the Regiolab Delft Webservice.

8.2.2 The SSNN Model

As fig. 8.3 shows there are 27 inductive loop detectors on the main carriage way of the A13 Southbound, of which two pairs are situated in parallel. This detector configuration results in a SSNN configuration with 24 hidden neurons, each representing a section enclosed by two consecutive detectors. On the A13, there are 6 on and 5 off ramps, of which 9 are located relatively close to each other in the Delft area. Not all of these ramps are equipped with loop detectors and in some cases there is a weaving section between consecutive on and off ramps. In all cases a hidden neuron receives both speeds and flows from detectors on the main carriage way. If a weaving section or ramp is connected to the section it represents, a hidden neuron will receive just flows from those detectors. Since the lay-out of our freeway stretch and its detectors has been determined, the topology of the SSNN model is fully defined and fixed. The sections are listed in table 8.1.

As a consequence the SSNN model used in this chapter contains 735 adjustable parameters. Although this is (by any standards) a large number of parameters, we may expect the Bayesian regularized training procedure to significantly reduce the complexity of this SSNN model (recall that in chapter 5 a reduction in model complexity of over 70% was achieved).

8.2.3 Data cleaning and preprocessing

Recall that the data cleaning and preprocessing layer involves data **checking**, **completion** and **correction** algorithms. Additionally, **pre-** and **postprocessing** algorithms are

Table 8.1: A13 Southbound Freeway Sections.

	from (m)	to (m)	Nr. Lanes	Input*)
1	5505	5910	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
2	5910	6350	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}
3	6350	6760	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
4	6760	7505	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{off}
5	7505	8050	3 + on ramp	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}
6	8050	8510	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
7	8510	9510	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{off}
8	9510	10000	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}
9	10000	10510	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}, q^{off}
10	10510	11510	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{off}
11	11510	12005	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}
12	12005	12305	3 + weaving section	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{off}
13	12305	12700	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
14	12700	13180	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
15	13180	13700	3 + on ramp	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{off}
16	13700	14500	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
17	14500	15007	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
18	15007	15535	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
19	15535	16500	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
20	16500	17005	3 + off ramp	$\{u,q\}^{up}$,$\{u,q\}^{down}$, q^{on}
21	17005	17500	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
22	17500	17985	3	$\{u,q\}^{up}$,$\{u,q\}^{down}$
23	17985	18300	4	$\{u,q\}^{up}$,$2 \times \{u,q\}^{down}$
24	18300	18500	4	$2 \times \{u,q\}^{up}$,$2 \times \{u,q\}^{down}$

required to make the data suitable for the SSNN model and convert the SSNN outputs. As such it performs the following routines

Data checking Detect missing data and replace them with *null* values.

1. MONICA outputs a reliability flag with each local detection. Measurements with these flags set to $false$ are dubbed "missing" by default. There are however a number of other circumstances in which data should (or should not) be dubbed unreliable
2. First generation inductive loop detectors (MTM1 detectors) do not measure speeds below 18 km/h. Occurrences of 18 km/h of these detectors actually depict speeds $\leq$ 18 km/h.
3. Speeds of 0 km/h can (per definition) not be measured locally. Any occurrence of measurements of 0 km/h are hence dubbed unreliable

Data completion and correction Repair missing / corrupted data.

1. In case of offline use (SSNN training), all data are repaired with the spatial/temporal interpolation routine presented in chapter 6.
2. In case of online use the moving average (MA) routine presented in chapter 6 is used, unless a particular detector is structurally failing (more than 10 time periods in a row), in which case the spatial interpolation routine is used
3. Correct for arithmetic time mean speeds. As discussed extensively in chapter 3, local time mean speed overestimates space mean speed and hence leads to underestimation of mean travel times estimated with the PLSB (or any speed based) trajectory method. Ideally, we should use the local harmonic time speed to avoid this bias, but these are not provided by MONICA detectors. We therefore apply the correction algorithm based on a differenced time series of time mean speeds proposed in chapter 3. This algorithm discriminates between free flowing and congested traffic by means of so called local density.

Pre- and postprocessing transform data in required (SSNN) input format

1. Scale all (corrected and completed) data for SSNN usage. As outlined in chapter 5 all input and output data to for the SSNN model is scaled to the interval $[0.1, 0.9]$[2]. Note that the scaling parameters of the training data sets, also govern the scaling of data presented to the model in real-time operation
2. Re-scale SSNN output and confidence intervals. Similarly as above, the SSNN output and the calculated confidence intervals have to be re-scaled back to their original domain[3]. Note that the model variance (sections 7.4 and 8.2.4) are zero-based and should only be re-scaled in terms of magnitude and not translated into the output domain[4]

[2]For example travel times are scaled using

$$\tau^* = 0.1 + \frac{(0.9 - 0.1)}{\tau_{\max} - \tau_{\min}} (\tau - \tau_{\min})$$

[3]The SSNN output is rescaled using

$$\tau = \tau_{\min} + \frac{\tau_{\max} - \tau_{\min}}{(0.9 - 0.1)} (\tau^* - 0.1)$$

[4]SSNN model variance is rescaled using

$$\sigma_\psi = \frac{\tau_{\max} - \tau_{\min}}{(0.9 - 0.1)} \left(\sqrt{\sigma_\psi^2}\right)$$

8.2.4 Confidence estimation for neural networks (II)

In chapter 7 we presented a analytical approach of quantifying the uncertainty in the SSNN parameters based on the assumption that the posterior distribution of these parameters can be approximated with a Gaussian integral (eqns 7.16, 7.17, and 7.18 in section 7.4). The variance of that multivariate normal distribution can then be derived analytically with the inverse of the Hessian matrix (second derivative of SSNN performance with respect to its parameters), which is already calculated during training. As a result the SSNN output can also be expressed as a Gaussian distribution with the SSNN prediction $G\left(\mathbf{u}\left(p-1\right),\psi\right)$ as mean and variance $\sigma_{\psi}^{2}\left(p\right)$.

In (Papadopoulos et al. 2001) three alternatives for neural network confidence estimation are compared., including the empirical Bayesian approach described in section 7.4. The other two approaches included non-parametric[5] bootstrapping (Efron 1979), and a method which augments a neural network such that model variance becomes one of the models' outputs. There are practical reasons why we have implemented an alternative method. The most important is that training an SSNN model with all available data is computationally very demanding with the Levenberg-Marquardt algorithm. In principle, for all three methods proposed in (Papadopoulos et al. 2001) using the entire data set available is a requirement. Nonetheless, we wish to train a model with as much input/output data (different traffic conditions) as possible. As an indication, training patterns are ordered as sequences, each of which reflects a 6 to 8 hour period with hence 360 to 480 observations (input vectors and targets). Let us denote such a sequence by $S_d = \{\mathbf{u}\left(p-1\right),\tau\left(p\right)\}_{p=1}^{P_d}$, where P_d denotes the number of observations in a particular daily sequence. We have a training database storing approximately 3 years of daily traffic sequences, totalling in over $375,000$ training records for the A13 highway alone.

A practical solution, which at the same time provides an alternative way of estimating model variance is *random subsampling* (Politis & Romano 1994) sometimes referred to as subsample bootstrapping or simply subsampling. Whereas non-parametric bootstrapping is a technique which exploits *resampling with replacement* to create multiple data sets $D_k, k = 1,..,K$, from some finite sample database $D = \{x_i, y_i\}_{i=1}^{N}$ of the same size (N) as the original data set; in subsampling data sets D_k of size B are *resampled without replacement*, with $B \ll N$. In case of the bootstrap, a statistical parameter θ (which could be the outcome of a ANN model) that makes inference based on data D can be applied to each of the bootstrapped data sets D_k. The mean $\widehat{\theta}$ over the bootstrapped estimates $\widehat{\theta}_k$ then provides a stable and asymptotically converging estimate of θ. The standard error of $\widehat{\theta}$ then equals the sample standard deviation over $\left\{\widehat{\theta}_k\right\}_{k=1}^{K}$ (Chernick 1999). This method is applied to confidence estimation for neural networks in for example (Heskes 1997) and (Papadopoulos et al. 2001). Similarly, subsampling also leads to a stable and asymptotically converging estimate of θ, again by taking the mean $\widehat{\theta}$ of the estimates $\widehat{\theta}_k$ on each sub data set D_k. As argued in (Politis et al. 2001),

[5] non-parametric pertains to the fact that no prior distributional assumptions are used in the procedure

"The method works under extremely weak conditions, it applies to independent, identically distributed (i.i.d.) observations as well as to dependent data situations, such as time series (possibly non-stationary), random fields, and marked point processes". Although the method asymptotically converges as $B/N \rightarrow 0$, convergence is almost guaranteed for $B \sim O\left(N^q\right)$ with $q < 1$ (Politis et al. 2001). The standard error of the estimate $\widehat{\theta}$ now equals the scaled sample standard deviation $\widehat{\sigma}^*_{\theta}$ of the individual subsample estimates of $\widehat{\theta}_k$, that is

$$\widehat{\sigma}_{\theta} = \sqrt{\frac{B}{N\left(K-1\right)} \sum_{k=1}^{K} \left(\widehat{\theta}_k - \widehat{\theta}\right)^2}$$

Since time dependence is crucial in our model a single 'datum' here depicts in fact an entire daily sequence S_d. Our database $D = \{S_d\}_{d=1}^{N}$ hence consists of a little over $1,000$ sequences. By taking enough subsamples, it is assured most (if not all) data are included in the resulting SSNN ensemble. Strictly speaking, if we train an ensemble of K SSNN models each on a subsampled data set $D_k \subset D$, we can make stable (statistical) inferences only over an entire sequence. We assume however, we can also use the method to estimate our model uncertainty $\sigma^2_{\psi}\left(p\right)$ for prediction on one particular input pattern $\mathbf{u}\left(p-1\right)$. Let

$$\overline{G}\left(\mathbf{u}\left(p-1\right)\right) = \frac{1}{K} \sum_{k=1}^{K} G_k\left(\mathbf{u}\left(p-1\right), \psi_k\right)$$

be the mean prediction of the ensemble of K models for datum $\mathbf{u}\left(p-1\right)$. The standard error (square root of $\sigma^2_{\psi}\left(p\right)$) around this mean then equals

$$\sigma_{\psi}\left(p\right) = \sqrt{\frac{B}{N\left(K-1\right)} \sum_{k=1}^{K} \left[G_k\left(\mathbf{u}\left(p-1\right), \psi_k\right) - \overline{G}\left(\mathbf{u}\left(p-1\right)\right)\right]^2} \qquad (8.1)$$

where we have scaled the standard deviation of the ensemble predictions with $\sqrt{B/N}$, in which B is the number of sequences per subsample and N the total number of sequences. In the ensuing we will use subsamples D_k of size $N/10$ to train the SSNN models, so that eqn (8.1) becomes

$$\sigma_{\psi}\left(p\right) = \sqrt{\frac{1}{10\left(K-1\right)} \sum_{k=1}^{K} \left[G_k\left(\mathbf{u}\left(p-1\right), \psi_k\right) - \overline{G}\left(\mathbf{u}\left(p-1\right)\right)\right]^2} \qquad (8.2)$$

8.3 Data

8.3.1 Subdivision of data sets

In this chapter we distinguish between three data sets, all three measured on the A13 Southbound freeway stretch introduced earlier. The first (data set A) is a large data

set we use for training the SSNN ensemble and consists inductive loop data and PLSB estimated travel times on 1071 afternoon peaks from 14:00 to 19:45 in the years 2000, 2001 and 2002, totalling in over $375,000$ individual records. Each record reflects a departure time period p and contains the PLSB estimated travel time and all speed and flow measurements from the detectors listed in table 8.1.

The other two data sets are used for testing purposes and are not included in the training data set. Test data set B is similar to data set A and contains inductive loop data and PLSB estimated travel times on 118 afternoon peaks from 14:00 to 19:45 in the first four months of 2003, totalling in over $41,000$ individual records. Test data set C finally, is a small data set on three afternoon peaks in 2002 on which we have actual travel time measurements. The measurements were recorded on July 4, 8 and 9 by means of video cameras. In this particular case, only vehicles passing the middle lane (of 3) are captured and time stamped on both the up- and downstream end of the route and semi-automatically matched afterwards. By means of an outlier detection algorithm, vehicles visiting the petrol station halfway the freeway are filtered out[6].

8.3.2 Input failure

Note that in all data sets input failure occurs. On the average, data sets A and B contain input failure of 12%, which implies that nearly 1 out of 8 measurements from the MONICA system are either missing or corrupt. On most days input failure is between 7-15%, albeit their are some serious outliers as the box plot in fig. 8.4 indicates (showing the minimum (left-most vertical bar), 25 percentile, median, 75 percentile (the box in the centre) and maximum (right-most vertical bar) percentages of input failure that occurred in test data set B).

Unfortunately, on the three afternoon peaks during which actual travel times were recorded (data set C), serious input failure occurred. There were numerous incidental input failures on a large number of detectors (cause unknown) while four main and three ramp detectors failed structurally on each of the three days (on locations 7505, 11510, 12005, 17005 [m]). From table 8.1 it can be verified that particularly the failing detectors on sections 10,11,12 may seriously affect the PLSB method, since it is that area (near on and off-ramp Delft-Zuid) where the heaviest congestion occurs. Table 8.2 gives some quantitative detail of the input failure occurring on these three days.

Although the travel time prediction framework does handle the missing data problem adequately (shown in the second part of this chapter), it does affect the analysis here. We can not make an unbiased estimate of the distribution of the PLSB estimation errors, since these may be influenced by the missing data problem. In chapter 6, it was

[6] It was found that the following procedure worked best: all travel times within a five minute departure time period that exceed the median observation in that period + 1.5 times the interquartile distance are considered outliers.

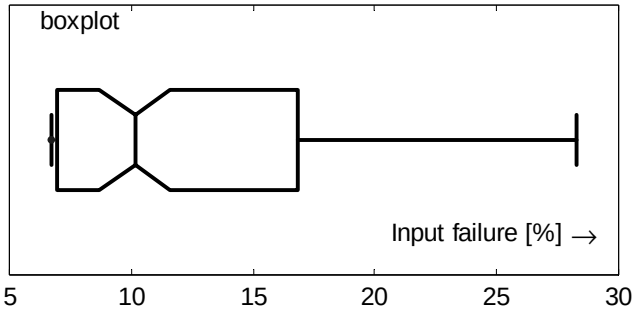


Figure 8.4: Boxplot of input failure in dataset B (118 afternoon peaks between 14:00 and 19:45). The figure indicates the minimum (left-most vertical bar), 25 percentile, median, 75 percentile (the box in the centre) and maximum (right-most vertical bar) percentages of input failure that occurred in test dataset B.

Table 8.2: Input failure on three afternoon peaks (July 7,8, an 9, 2002 between 14:30 and 18:30) on the inductive loop detectors along the A13 Southbound.

	Nr. Missing	Total Obs.	% Missing
4/7/02	2952	14760	20
8/7/02	2388	14760	16
9/7/02	2251	14760	15
Total	7591	44280	17

found that particularly structural failure increases both mean and variance of the estimation error, despite robust algorithms to repair the corrupted data. Consequently, we can not correct the SSNN prediction for possible bias due to training with PLSB estimated travel times. On the positive side, the corrupted data set allows us to demonstrate the methods developed on synthetic data to tackle data corruption (chapter 6) also work in real life. It also emphasizes the need for robust travel time prediction methods.

8.3.3 The distribution of real travel times

In this section the distribution of real travel times is investigated, on the basis of data set C. We adopt a slightly different approach than in section 7.2, due to the fact we have much fewer observations to support any hypothesis on the distributional characteristics of travel times. Considering that during small time periods p the mean travel time $\tau\ (p)$ may be considered stationary, the original time series can be written as

$$\tau_{it} = \tau\ (p) + e_\tau\ (p)\,,\ t \in p \tag{8.3}$$

in which $e_\tau\ (p)$ is a (zero mean) random noise term. Eqn 8.3 tells us the individual

travel time in departure period p equals the mean travel time in that period plus some random term $e_\tau(p)$ which is time dependent. If the period size p is set to for example one minute, and the mean $\tau(p)$ is subtracted from the time series, the result $\tau_{it} - \tau(p) = e_\tau(p)$ can be used to estimate the probability density $g(\tau)$ of travel times. Note that although subtracting the mean removes non-stationarity with respect to the mean, variance (governed by the noise term in eqn 8.3) may still be time dependent. Similarly to the analysis in section 7.2 we investigate whether or not the standard deviation of the travel time distribution $\sigma_\tau(p)$ can be expressed as a function of mean travel time $\tau(p)$.

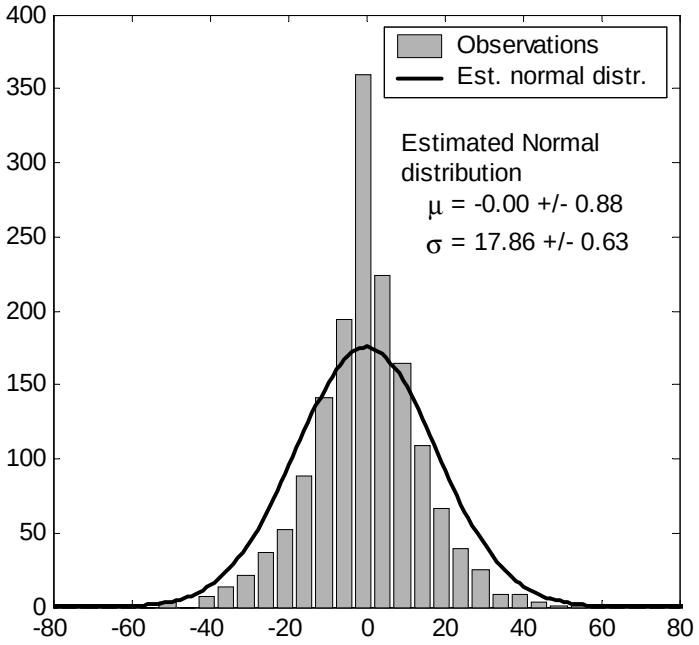


Figure 8.5: Histogram and estimated normal distribution of real measured travel times on the A13 Southbound for all speeds.

It appears the real travel time distribution can be approximated fairly closely with a zero mean *normal* distribution (fig. 8.5) for all speed ranges, with variance that is approximately stationary, that is, the width of the travel time distribution does not depend on mean travel time, as was the case with the simulated data. Moreover, real travel time standard deviation (square root of variance) is found much smaller than based on simulated data:

$$\sigma_\tau(p) \approx 17 \ [s]$$

In the introduction we already noted that the micro-simulation tool FOSIM is not designed to generate very accurate travel time distributions. A second reason for the

large discrepancy is that the data set of travel times used here only reflects a (biased) subset of all travel times occurring on the measurement days, since only vehicles on the middle lane were tracked. This may bias both mean travel time and travel time variance toward zero, since for example most trucks will not be recorded. Moreover, it also forces the distribution to become more Gaussian (less skewed).

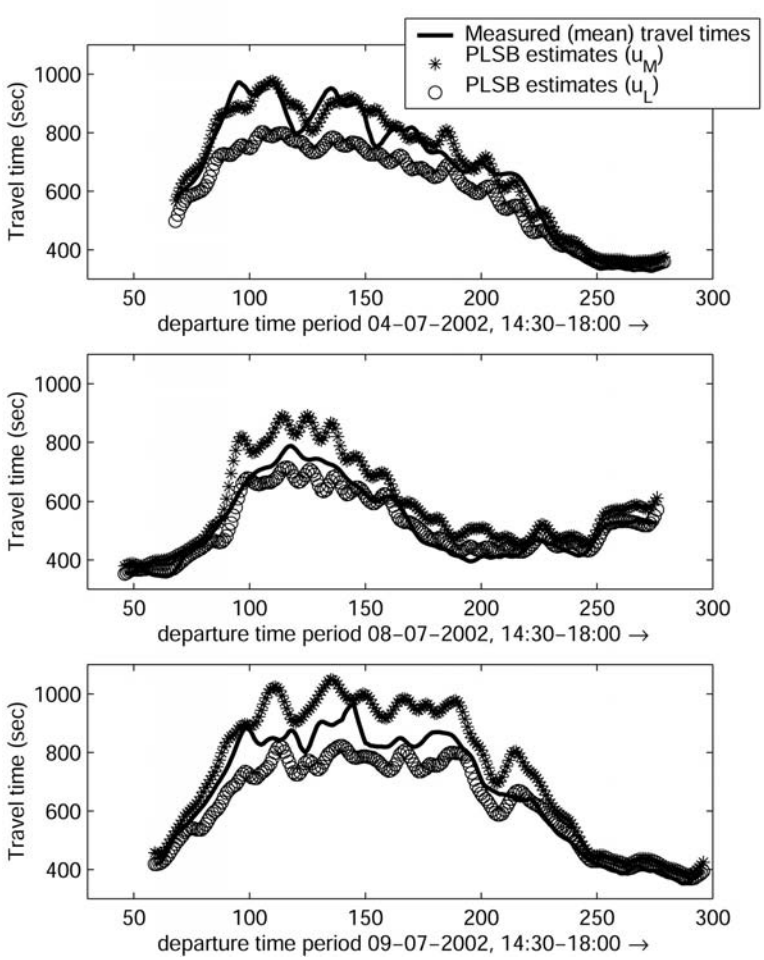


Figure 8.6: Performance of the PLSB method based on arithmetic and (estimated) harmonic mean speeds against measured travel times on three afternoon peaks on the A13 Southbound. The former underestimates travel times, while the latter overestimates (albeit less severely) travel time in congested conditions. Note that for better readability all data are filtered with a twosided moving average filter (5 minute window).

8.3.4 PLSB travel time estimation errors

The second source of uncertainty identified in the previous chapter was due to the PLSB travel time estimator, which we use to generate the target data for the SSNN model. In all previous chapters we used simulated data to generate *harmonic* time

Table 8.3: Performance of PLSB travel time estimator based on (estimated) harmonic mean speeds and arithmetic time mean speeds on real data (performance indicators from appendix A).

	PLSB ($\widehat{u}_M$)	PLSB (u_L)
ME (s)	25	-40
SE (s)	63	71
MRE (%)	5	-4
SRE (%)	10	10
RMSEP (%)	11	14

mean speeds at local detectors. This is a luxury we do not enjoy in case of the MONICA inductive loop detector system, which records arithmetic time mean speeds. Recalling chapter 3 arithmetic time mean speeds structurally overestimate space mean speeds and hence leads underestimation of mean travel times which are reciprocally related to space mean speeds. In order to obtain valid travel times with which to train the SSNN model we first need to correct for this intrinsic detector failure. In the ensuing we will determine the residual bias and variance due to the PLSB travel time estimation procedure with which we can quantify the uncertainty related to training the SSNN model with PLSB estimated travel times. Again data set C is used.

Correcting for arithmetic time mean speeds

In a stationary and homogeneous state the arithmetic time mean speed u_L is always larger or equal than the so called space mean speed u_M, which we require for travel time estimation, that is

$$u_L = u_M + \frac{\sigma_M^2}{u_M}$$

in which σ_M^2 denotes speed variance. In chapter 3 two algorithms were presented to correct for the bias $\left(\sigma_M^2/u_M\right)$ caused by arithmetic mean speeds. The first was a linear regression method (naive method) relating variance to time mean speeds and the second was based on a differenced time series of mean speeds, and local density (that is the traffic flow divided by time averaged speed). Here we use the latter one, which was found to perform best. Given an estimate for σ_M^2, harmonic mean speed is estimated with

$$\widehat{u}_M = \frac{1}{2}u_L + \sqrt{\frac{1}{4}u_L^2 - \widehat{\sigma}_M^2},\ \widehat{\sigma}_M \leqslant \frac{1}{2}u_L$$

Fig 8.6 shows the general difference between the PLSB estimates based on arithmetic mean speeds u_L and the one based on (estimated) harmonic mean speeds u_M against

measured travel times on the three afternoon peaks on the A13 Southbound. Note that all three time series have been filtered[7] for better readability. For free-flow and mildly congested conditions (travel times up to 500 seconds) both methods produce fairly accurate estimates, for congested conditions, the PLSB method based on u_L structurally underestimates travel time while the one based on $\widehat{u}_M$ tends to overestimate travel times. Table 8.3 confirms these results. Based on the small test data set and the aforementioned missing data problem, no preference for either method can be quantitatively confirmed. We argue, however, that overestimation on a few measurement days due to obvious data problems is preferable to underestimation due to clear theoretical errors made by taking arithmetic mean speeds.

Bias and variance of the residual PLSB errors

Fig 8.6 and table 8.3 show that based on (estimated) harmonic mean speeds the PLSB method still introduces significant bias (in this case a slight overestimation) and variance (relative standard error on the average 10%). Here we list the various reasons identifiable.

- The results are (as mentioned earlier) influenced by the high percentages of missing data mentioned earlier. This affects both the speed variance estimator which corrects time mean speeds, as well as the PLSB method itself. Although robust algorithms for dealing with input failure are applied, specifically structural detector failure in congested areas increase both mean and variance of the estimation error.
- Whatever speed based travel time estimation method is used, the basic assumptions of stationarity and homogeneity on a road segment enclosed by detectors within a single time period may not hold.
  - In real life, and certainly in transitional phases between free-flow and congested traffic, is not stationary, even within small (1 minute) time periods.
  - Probably more seriously, the detectors are not 'ideally' placed (for the PLSB procedure) on this particular road stretch. Some are located on weaving sections where locally traffic conditions may occur that are not representative at all for an entire section. The fact there are many off and on ramps connected to this route makes it likely the assumption of homogeneity is violated in many cases due to the complex traffic dynamics (e.g. shockwaves, lane change behavior, weaving) on the route.

[7] This filter replaces each datum with

$$\overline{u}(t) = \frac{1}{2N+1} \sum_{k=t-N/2}^{t+N/2} u(k)$$

where in this case $N = 2$

- We have a very small test set (3 afternoon peaks) of actual measured travel times. Moreover, the samples are non-random, that is, they consist of measurements from the middle lane only and may thus be biased (e.g. most trucks drive on the right lane).
- Finally, although we correct for the bias due to arithmetic time mean speeds, the correction algorithm may introduces extra variance and likely some bias in its own right.

Assuming that the residual $e_P(p) \sim N\left(\mu_P(p), \sigma^2_P(p)\right)$ is normally distributed with mean $\mu_P(p)$ and variance $\sigma^2_P(p)$ we could calculate (see section 7.3.2) through linear regression wether either mean and / or standard error $\sigma_P(p)$ of the PLSB estimation errors $e_P(p)$ are proportional to mean (estimated) travel time $\tau_{est}(p)$. Due to the reasons listed above, we argue this is not a very meaningful exercise. We will therefore refrain from applying bias correction (as in chapter 7), since we expect a linear relation found here is more related to the high degree input failure and idiosyncrasies of the limited test data than structural errors of the PLSB method. We will, however, use an estimate of the standard deviation based on test data set C for constructing prediction intervals, which equals:

$$\sigma_P(p) = 50 + 0.09\,\tau_{est}(p)\ [s]$$

and hence ranges between 70 and 130 seconds for free-flow and congested conditions, which is in line with the findings of chapter 7 (section 7.3.2).

8.4 Results

8.4.1 SSNN Training

Given the fact that the training data set (A) is substantially bigger than the ones used in (Heskes 1997), we will use a smaller sized neural network ensemble ($K = 26$, with subsampled data sets D_k of size $N/10$), than advocated in the aforementioned paper where the ensemble size was set to 50. Although training 26 SSNN models seems a bad idea in terms of computational expense, it actually appears that when the parameters of each the SSNN models are initialized with the parameters of a previously trained SSNN model (if available), training speeds up and training time is reduced in some occasions more than 50%. On average each model still required 6 hours and 7 minutes training time on a Pentium IV 3.1 GHz machine. We hypothesize the total time required for training the ensemble on subsampled data sets is still less than the time required for training one SSNN model on all data.

As noted above, a single SSNN model in this case has 735 adaptable parameters (weights). As expected, the number of effective parameters (refer to chapter 5) after training with the LM-BR algorithm is significantly lower, ranging from 150 to 200

in the 26 trained models, implying a reduction in model complexity ranging from 73 to almost 80%(!)

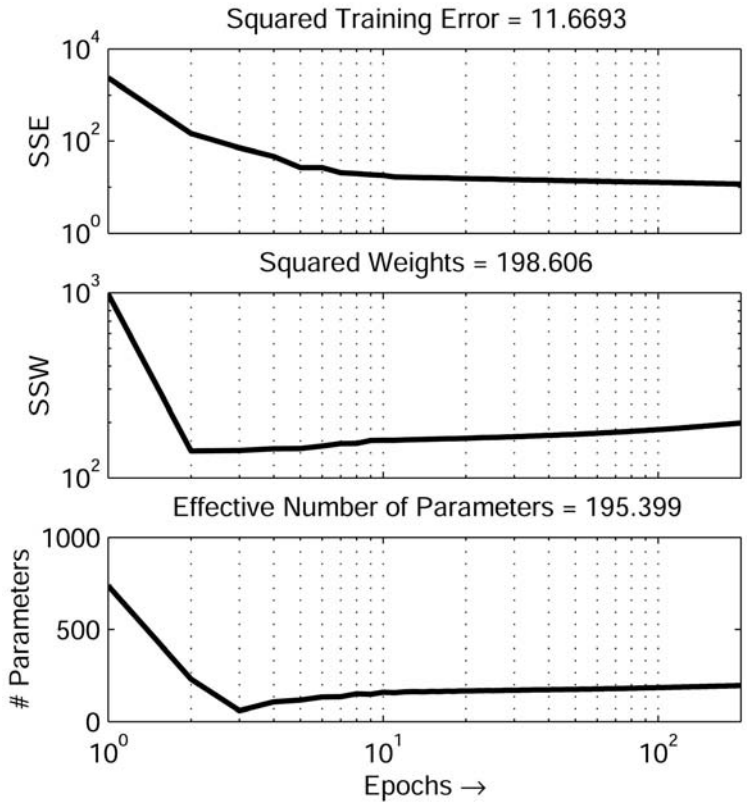


Figure 8.7: Typical result of SSNN training session. Note the horizontal axis (epochs) in all three graphs is also on a logarithmic scale

8.4.2 Performance on estimated travel times

In this section we use data set B and assume the (PLSB) estimated travel times are unbiased estimates of "real" travel times. Although due to reasons mentioned earlier this may not be a correct assumption, it does not affect the analysis. The travel time prediction problem still is a complex non-linear spatiotemporal mapping from current traffic conditions to mean travel times for vehicles departing in the next available time period. We assess the SSNN framework in real-time, similarly to chapters 5 and 7 by answering the following questions

1. Does the SSNN model accurately predict travel time under different traffic conditions? As a baseline comparison we will again use predicted travel times of the instantaneous travel time predictor, see definition on page 27). We also compare the SSNN performance against some (mostly data driven) models reported in literature. Although the circumstances between models and experiments may

differ largely (data used, size and geometry of freeway stretch, etc.), this comparison provides quantitative evidence of the suitability the proposed model (the SSNN) for ATIS in general.

2. If the model produces acceptable results, *what does the SSNN learn* from the data*?*

   (a) in terms of its short term memory (the internal states)

   (b) in terms of its long term memory, that is its parameters (weights and biases)

3. do the SSNN confidence intervals reflect the uncertainty in each prediction in case of missing data and unknown traffic conditions?

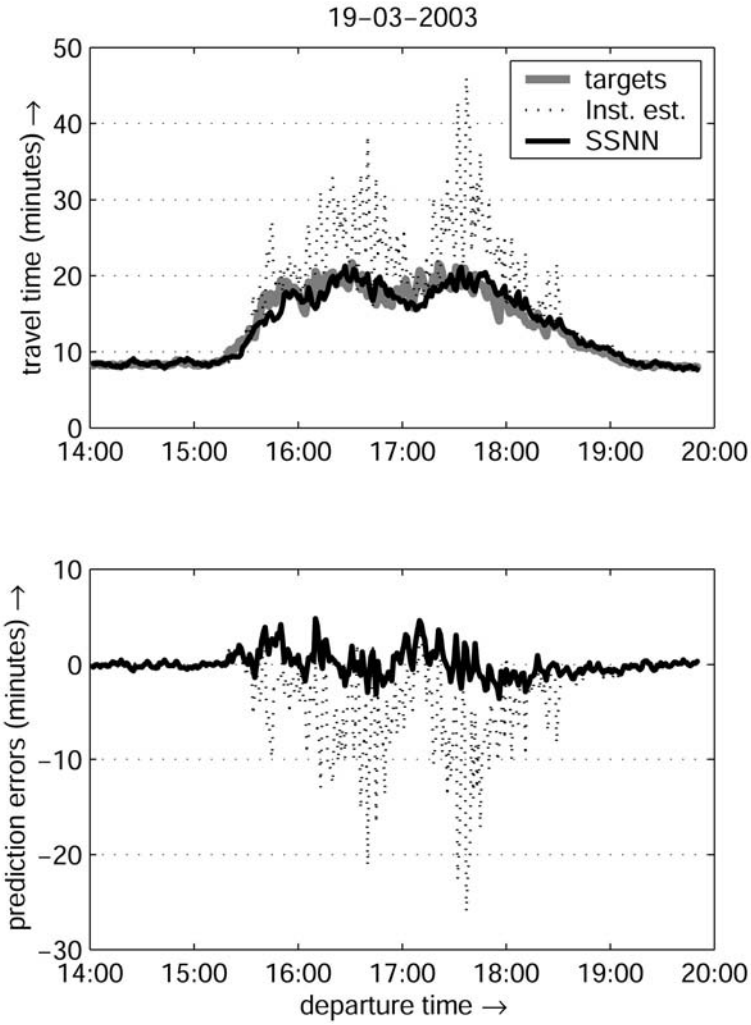


Figure 8.8: Performance of the SSNN model versus an instantaneous travel time predictor on a typical afternoon peak on the A13 Southbound.

Predictive performance

Based on data set B, table 8.4 shows the mean and standard deviation of the relative prediction error and the root mean of squared error proportional for both the SSNN and the instantaneous predictor. Note that both models are fed with cleaned and corrected data with the tools developed in earlier chapters.

Again, as expected, the SSNN outperforms the instantaneous predictor by far in terms of both bias and variance. As with the experiments on simulated data, the SSNN still produces approximately zero mean distributed residuals, while the instantaneous travel times are now even more biased. Fig 8.8, which shows the predictions of both models on a typical afternoon peak (March 18 2003), illustrates how the instantaneous estimator overestimates travel times largely and moreover, provides very unstable predictions. The main reasons are already outlined in chapter 4. Instantaneous travel time assumes stationary conditions at each departure time period, which clearly does not hold in congested traffic.

Table 8.5 compares the mean absolute relative error (MARE - see appendix A) of the SSNN on data set B with the same performance measure reported for some travel time prediction models in literature. To make the comparison more meaningful a column with remarks on the length of freeway route and the data used is also included, since these application-specific circumstances influence the reported results largely. Moreover, also the geometry and the prevailing traffic conditions differ between applications. For example, the relative amount of free-flow travel times in the test data set strongly biases the MARE (or any absolute performance measure) toward zero (Chun-Hsin et al. 2003). Also, the variability of actual travel times and the difference between free-flow and congested travel time in the test data set has a dramatic effect on performance. In our case, approximately four out of seven travel times (from a total of 41,000) are considerably higher than free-flow travel times. Finally, the time series approaches (e.g. (Park & Rilett 1999)) do not strictly speaking perform online prediction, but predict the next datum in a time series, which still most likely is a past travel time (see also chapter 4). Nonetheless, from table 8.5 a reasonable conclusion is that performance (MARE) in the range of 4 to 6% is generally considered "good". As such, the SSNN performance is satisfactory, certainly since in most other studies corrupted or missing data was left out of the test data set on before hand, while in our case it was left to the preprocessing layer and subsequently the SSNN to deal with input failure (on average over 12% was missing with some extremes up to 25%). In our case, handling with data corruption is an integral part of the prediction task. Particularly the instantaneous predictor is affected severely by these degrees of input failure resulting in a MARE of 15.4%.

We conclude the SSNN is both robust and accurate, not only compared to an instantaneous predictor on the same data set but also to a range of other travel time prediction models reported in literature. Due to application-specific circumstances we argue a ranking from worst to best performing models is not very meaningful, however, the

SSNNs' performance is comparable to the best performing models in literature.

Table 8.4: Predictive performance of SSNN model and instantaneous travel time predictor on 118 afternoon peaks in 2003.

	RMSEP (%)	MRE (%)	SRE (%)
SSNN	13	0.64	8.1
Inst. Travel Time	56	13	28

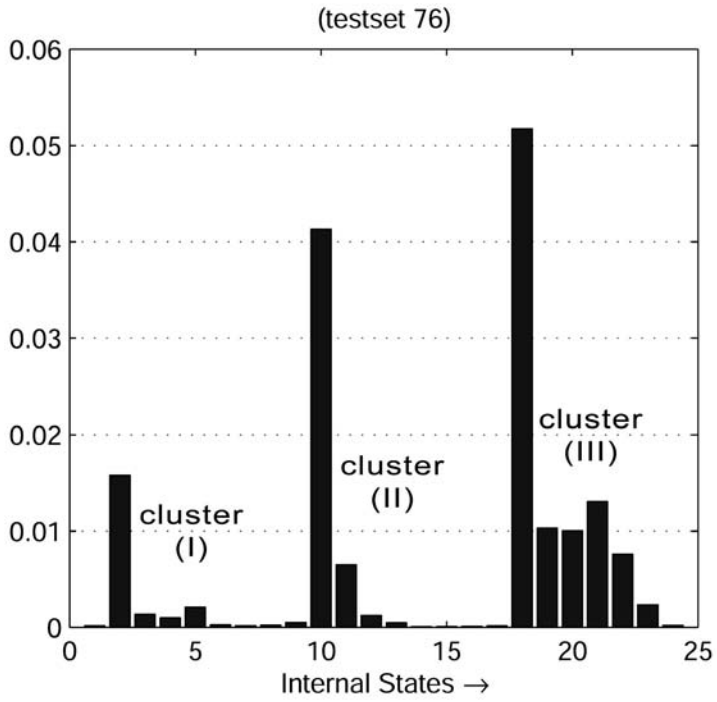


Figure 8.9: Relevance if the internal states of the SSNN calculated on March 19 2003

Analysis of the internal states

As in chapter 5, this section analysis the internal states of the SSNN model. For each SSNN model k in the ensemble the relevance $S_{m,k}$ of internal state m is calculated with equation 5.18 on page 125. The mean relevance for each internal state in the ensemble then equals

$$S_m = \frac{1}{K} \sum_{k=1}^{K} S_{m,k} \tag{8.4}$$

Fig 8.9 shows a typical result of the relevance of the internal states calculated for the same afternoon peak (March 18 2003) as in Fig 8.8. Very clearly, there exist three

Table 8.5: Predictive performance (mean absolute relative error- MARE) of SSNN model compared to performance of other models found in literature.

model	MARE (%)	remarks on test dataset
SSNN (this dissertation)	5.4	18 km, 108 afternoon peaks, dual loop data
Inst. Predictor (this dissertation)	15.4	idem
FNN (Huisken & Van Berkum 2003)	4.6	10 km, 13 peak periods, dual loop data
Inst. Predictor (Huisken & Van Berkum 2003)	10.7	idem
Kalman Filter (Park & Rilett 1998)	6.2	27.6 km, 231 days, AVI data
Modular FNN (Park & Rilett 1998)	8.1	idem
FNN (Park & Rilett 1998)	9.0	idem
Spectral-bases FNN (Park et al. 1999)	7.2	idem
FNN (Innamaa 2001)	4.9 - 6.9	10-28 km, 2-3 weeks, travel time data
Lin. Regr. / Inst. Predictor (Zhang & Rice 2003)	6-11	10 km, 20 weekdays, dual loops
Support-Vector Regression (Chun-Hsin et al. 2003)	3.9 - 4.4	45 km, 5 weeks, dual loops

clusters of relevant neurons: (I) a cluster representing sections on the upstream half, between km 6 and 8; (II) a cluster representing sections between km 10 and 12 km near the three closely situated on and off ramps around Delft; and (III) a cluster representing the downstream sections between km 15 and 17.

Fig. 8.10 shows a contour plot of mean (corrected) speeds on March the 18^{th}. Note that dark areas depict low speeds in fig. 8.10. Using the terminology of (Kerner 1999), we can view at least three typical congested patterns. At 15:00 a *wide moving jam* starts propagating upstream along the entire freeway stretch. At the same time speeds drop on most of the 6 downstream sections to around 20 m/s, indicating *synchronized flow* conditions. From 16:00 onwards wide moving jams propagate from cluster (III) to cluster (II) also from cluster (II) to cluster (I). Around the freeway sections represented by clusters (III) and (II) the so-called *general congested pattern* with very dense and low-speed (stop-and-go) traffic occurs. Although Kerners three phase traffic theory is not undisputed (see e.g. (Helbing & Treiber 2002)), one could argue the SSNN maps the measured traffic patterns in a similar way as Kerner does, and hence provides

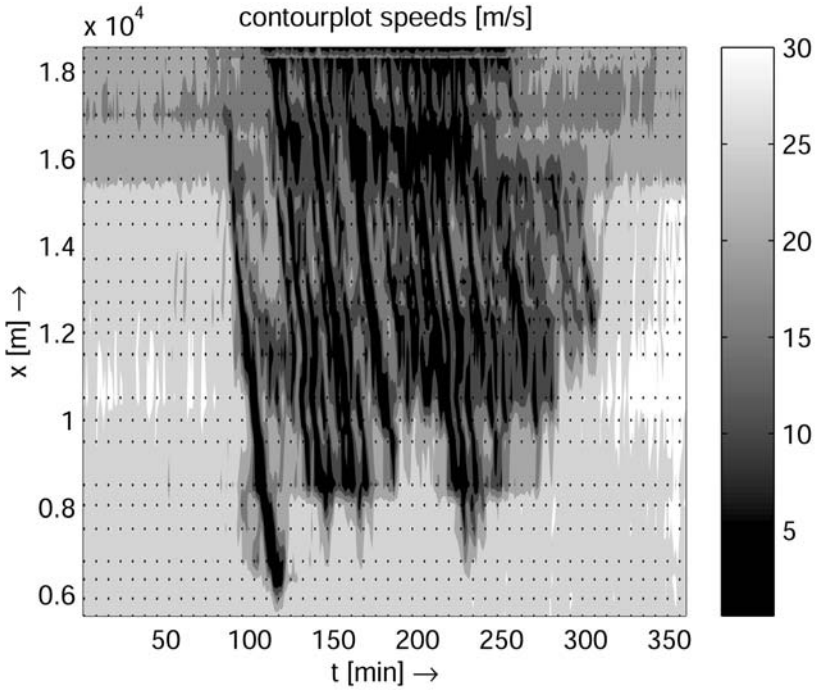


Figure 8.10: Contourplot of speeds on March 18 2003. The dotted horizontal lines depict the locations of main carriage way detectors.

some evidence for his theory. In case of the SSNN, given the observed training data (set A), this particular mapping (weight setting) is most probable. Two of the relevant clusters represent the two areas on which the general (and most severe) congested pattern occurs while the third represents the area on which the tail of the queue resides. Between the sections represented by these clusters wide moving jams propagate in the upstream direction (shockwaves).

Quantifying Uncertainty

Fig 8.11 shows 95% confidence intervals around the SSNN predictions on the same afternoon peak as above. As with the simulated data in chapter 7, the coverage percentage is low. Nonetheless, as can be expected the confidence intervals grow wider - up to 1.4 minutes (72 seconds) - as travel times are higher (around 20 minutes), reflecting the fact that the travel time prediction task in congested conditions is more complex than in free-flow conditions (intervals of 20 seconds, mean travel times around 8 minutes).

In different picture emerges on Friday January 3^{rd} 2003, on which in the late afternoon an accident occurred on the A13, closing down two out of three lanes (Fig 8.12). Travel times increase steeply, while the SSNN model does not immediately respond, since traffic flow patterns occur with which it is not familiar. The confidence intervals here are much wider than in Fig 8.11, up to 5 times larger than in the previous case. (72 seconds). Confidence levels here quantitatively indicate that "something is wrong"

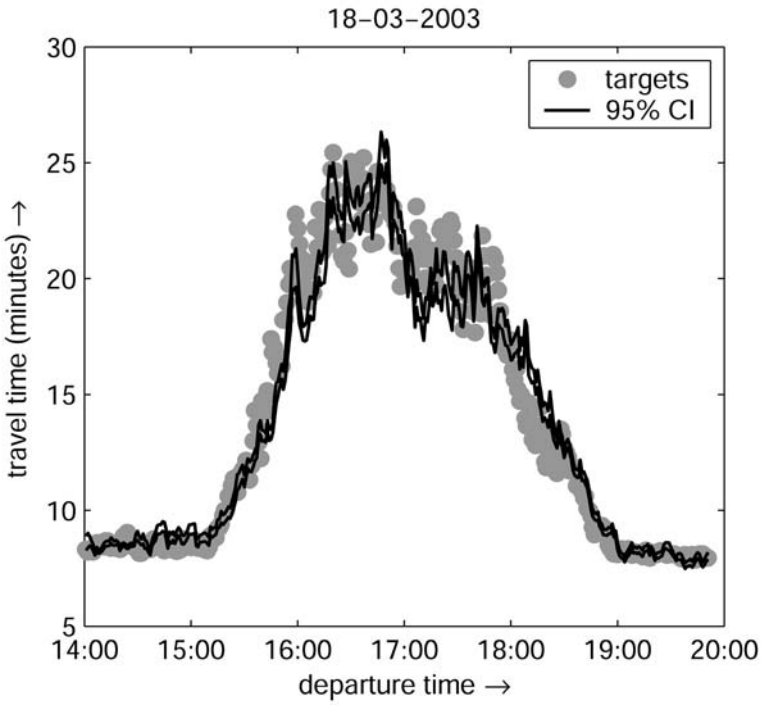


Figure 8.11: Confidence intervals around the SSNN prediction.

with the SSNN model or the data with which it is fed. In this case, the input data are outside the SSNNs input domain.

Fig. 8.13 illustrates that also on real data, the magnitude of the confidence intervals correlate positively to prediction error. The correlation coefficient ($R-value$) between the confidence interval width and the absolute prediction error on test data set B equals 0.6. Fig. 8.13 shows this relationship on the same two days also used in figs 8.11 and 8.12, March 18 and January the third respectively. On January 3 (Fig 8.13bottom) an accident occurs, yielding a steep increase of both prediction error and confidence interval width. In this case both quantities are over 400% of their "normal" magnitude (fig. 8.13top).

8.4.3 Performance on measured travel times

Finally, in this section we show - qualitatively - the SSNN performance on data set C (real measured travel times). On these three days, the prediction intervals are very wide, reflecting the distributional assumptions on the PLSB estimation errors, which are (due to reasons mentioned earlier) fairly large. Since we can not correct the SSNN for the bias due to training with the PLSB method, the prediction intervals cannot be interpreted as upper and lower bounds, albeit due to their width they cover almost 100% of all measurements.

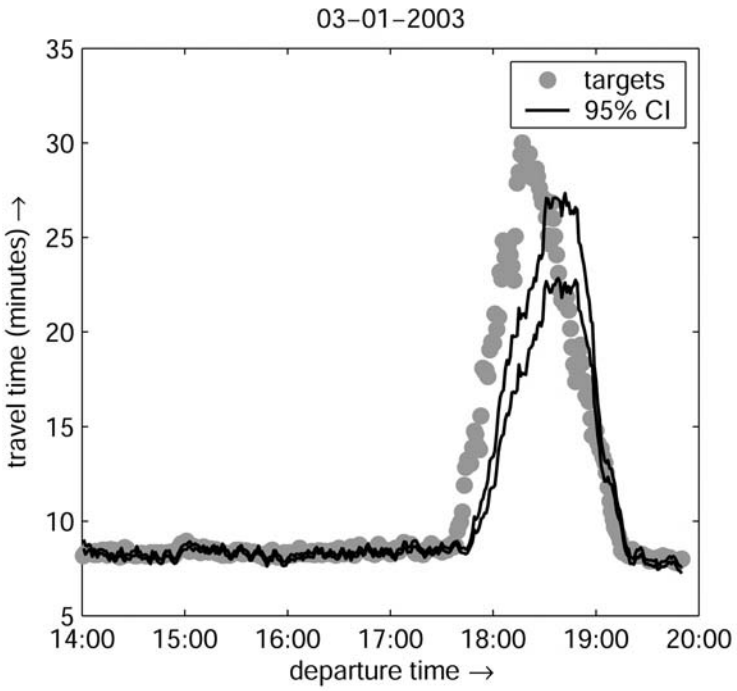


Figure 8.12: Confidence intervals in case of a major accident.

8.5 Comparison of Simulation and Real-time Results

Applying the SSNN framework in a real-time situation inherently is more complex and elaborate than in a simulated environment. The most prominent differences are

1. The discrepancy between the simulated travel time distribution and the real travel time distribution. The latter one does not widen as travel time increases, rather, it becomes smaller due to the fact that vehicles are constraint in congested conditions and individual travel times are strongly statistically dependent. As a consequence, the predictive distribution is much smaller than the one calculated in chapter 7.

2. The performance of the offline PLSB travel time estimator, although still good, is significantly worse on real data than on simulated data. The reasons are straightforwardly identifiable:

   (a) The analysis are based on a small and probably biased sample of real travel times

   (b) On the measurement days serious input failure clouds the results and introduces an extra source of bias and variance.

   (c) The basic assumptions of stationarity and homogeneity in space time regions $\{k, p\}$ are typically violated due to complex real life traffic processes and the particular locations of detectors.

Nonetheless, when we consider the PLSB estimated travel times as "true" travel times, the resulting travel time prediction problem still is a complex non-linear spatiotemporal mapping from current traffic conditions to mean travel times for vehicles departing in the next available time period. As such, we argue this chapter provides solid evidence the SSNN framework is both accurate, valid, robust and hence reliable.

1. the SSNN framework produces almost unbiased results on a large test data set
2. It handles moderate (12%) to sometimes high (20-30%) degrees of input failure adequately without serious loss of predictive accuracy
3. The warning mechanism based on confidence levels also works in real-time. On a day on which a major incident occurred, confidence intervals were three times larger than normal

We recommend, however, a larger scale study to improve the offline travel time estimation procedure, on a number of different freeway stretches. Such a study would require larger scale travel time measurements than the ones used here. With a better offline estimation procedure, inherently, a even more accurate SSNN model (with respect to actual travel times) can be trained.

8.6 Summary

In this chapter we showed the short term freeway travel time prediction framework developed in this thesis can be successfully applied in a real-time environment. The SSNN model (chapter 5) is capable of predicting (PLSB estimated) travel times in various traffic conditions. From the 735 parameters in the SSNN model, 20-25% were dubbed effective after the Bayesian regulated training scheme, a reduction in model complexity of up to 80%. One could argue the SSNN model efficiently classifies spatiotemporal traffic patterns in a manner resembling three phase traffic theory proposed by (Kerner 1999). The SSNN has learnt on which freeway segments the most severe congestion occurs (general congested patterns) and infers expected delays based on the evolution of traffic conditions on those sections.

Furthermore, the framework handles missing data in a similar (robust) manner as in chapter 6. Even at high levels of corruption (sometimes over 20%, on the average 12%) the SSNN model still produces almost unbiased results on a test data set of 118 afternoon peaks from 14:00 to 20:00 in 2003. The uncertainty in the SSNN predictions can be quantified in the same unified way as was demonstrated in chapter 7. Confidence levels also in real life traffic, provide a very useful "warning" mechanism. In case of a lane closure, it is shown that confidence intervals grow very large, depicting large uncertainty in the SSNN models prediction, offering the traffic manager an invaluable

tool to monitor the predictive quality of the SSNN framework as well as the quality of information chain feeding this framework.

Finally, we recommend a larger scale study to improve the offline travel time estimation procedure. Such a study would require larger scale travel time measurements on a number of different freeway stretches. In the next chapter, we discuss a number of limitations of the SSNN model in its current form and propose extensions to the model to account for those limitations, hence increasing its predictive power and widen its applicability.

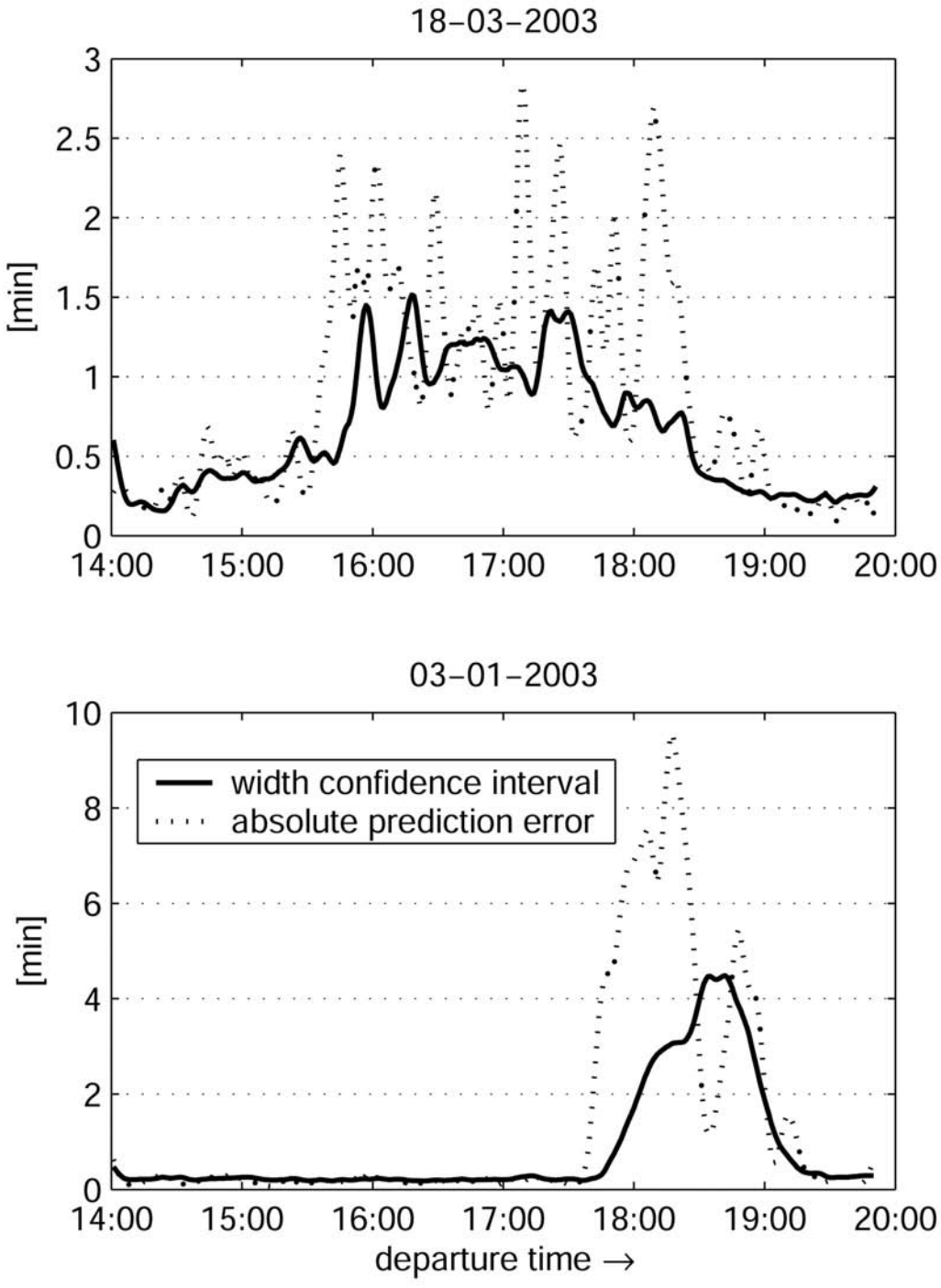


Figure 8.13: Width of confidence interval and absolute prediction error plotted for two afternoon peaks in 2003. On the third of January (bottom) an accident occured, causing an increase in both error and confidence interval width of over 400% with respect to normal conditions (top).

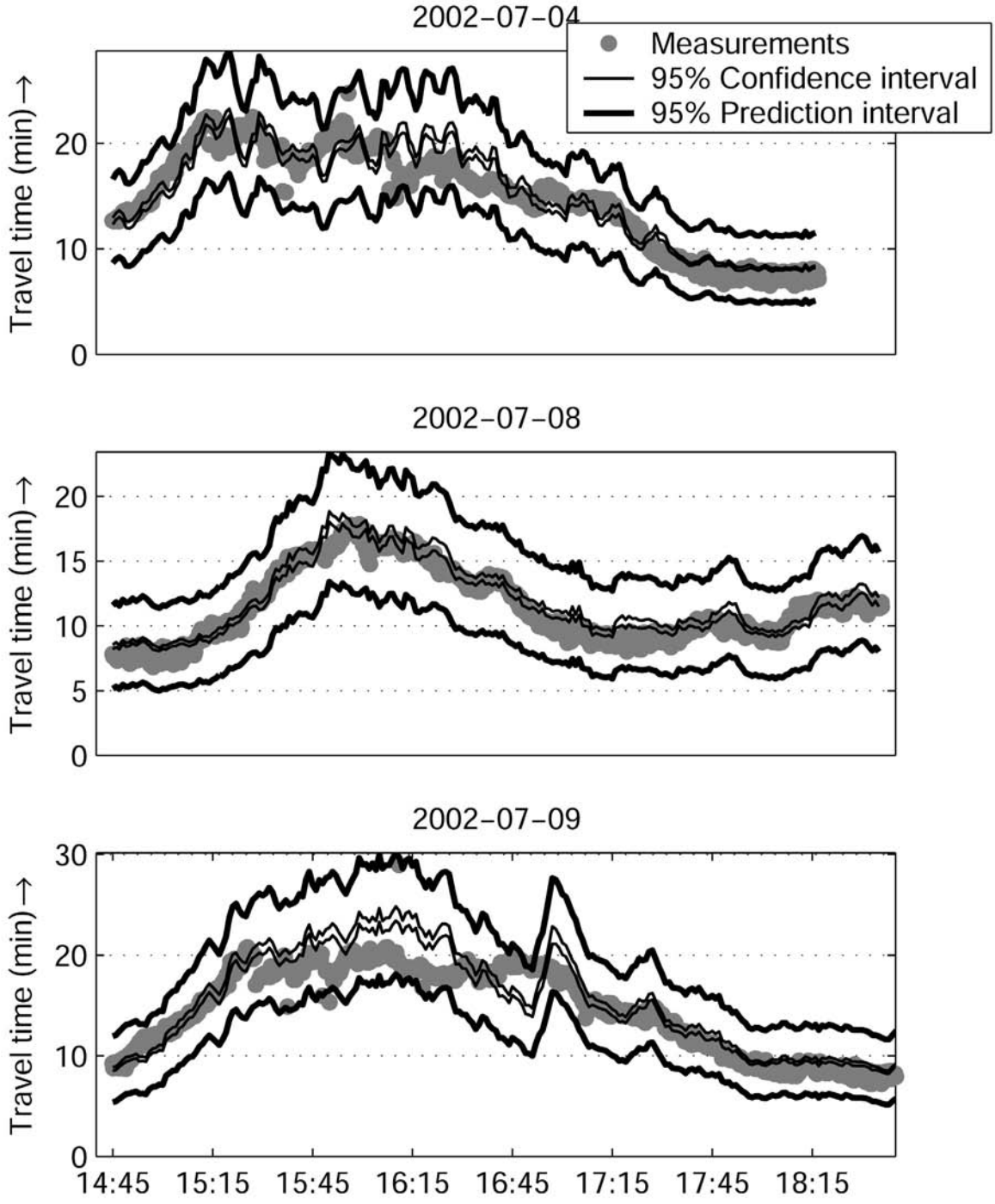


Figure 8.14: Confidence and prediction intervals around the SSNN predictions on Dataset C (measured travel times).

Chapter 9

Extensions to the SSNN Framework

9.1 Introduction

In this final chapter of the main body of this dissertation thesis a number of limitations of the short term freeway travel time prediction framework are discussed and subsequently extensions are proposed to overcome these limitations. Note that no numerical underpinning nor validation of these concepts will be provided. Nonetheless, since the proposed extensions follow naturally from the framework and the topology of the state space neural network (SSNN) on which this framework centers, it is reasonable to assume that they lead to potentially useful models. We will discuss extensions to account for the effect of

1. traffic conditions elsewhere in the network
2. ambient factors such as weather.
3. traffic control

We will also briefly discuss application of the SSNN model based on different traffic data collection systems. We argue that as long as quantities are measured along the route which allow for a state space representation, the SSNN is an appropriate modelling choice for travel time prediction.

Next, we will (qualitatively) elaborate on travel time prediction on urban road facilities. The predominant difference between urban and freeway networks is in the fact that travel times on urban networks are strongly influenced by traffic control. As a result, the SSNN might not be a very obvious modelling approach for urban travel time prediction. Furthermore, we briefly discuss an online correction algorithm that uses the last known (actual or offline estimated) travel time to correct the current prediction, hence improving the robustness and reliability of our framework. Finally, we will briefly address the issues involved in longer term travel time prediction.

9.2 Extending the SSNN model

Recall (chapter 5 and appendix B) the equations that constitute the SSNN model:

$$\mathbf{x}(t) = \Theta(\upsilon_0 + \upsilon\mathbf{u}(t) + \nu\mathbf{x}(t-1)) \tag{9.1}$$

$$y(t) = \phi\,(\omega_0 + \omega\mathbf{x}(t)) \tag{9.2}$$

in which $\mathbf{x}(t)$ reflects the vector of internal states at time instant t, $\mathbf{u}(t)$ the vector of current inputs on the route of interest, $y(t)$ the SSNN output and $\phi\,(z)$ the sigmoid transfer function ($\Theta(\mathbf{z})$ its matrix version). Furthermore, in equations (9.1) and (9.2), υ_0 and ω_0 denote the hidden layer and output bias, and υ , ν , and ω denote weight vectors associated with the SSNN inputs, context layer and output layer respectively.

9.2.1 Limitations of the SSNN Model

In its current for the SSNN model can only be applied on freeway routes on which the predominant causes of delay (e.g. bottlenecks) are located on the route itself. The reason is that the causes of delay (extra travel time) have to 'visible' on the route itself, since the SSNN is exclusively fed with data measured on the route of interest. If there are other causes (e.g. spill back from a connecting freeway stretch), there is no way this relationship can be distilled (at least in a way that makes sense from a traffic engineering point-of-view) with the current input-output data.

Secondly, the SSNN model is fed only with data from a traffic data collection system. As outlined in chapter 2 there are also many external factors (not related to the traffic processes themselves) that influence travel time, for example

1. Ambient conditions such as weather and visibility strongly influences driver behavior and hence traffic conditions. These effects may be visible in aggregate measurements but are not accounted for explicitly in the SSNN model.

2. Traffic control and information influences driver behavior and collective traffic operations and hence mean travel times. Again the net effect of those measures ultimately are "visible" in raw traffic measurements, but the SSNN does not explicitly account for them.

Arguably, the effect of these factors are intrinsically "visible" in traffic data. Nonetheless, incorporating them explicitly in the SSNN model, may lead to a better and more general travel time prediction model. Although in chapter 5 we found the internal workings of the SSNN are closely related to the traffic process, and that each internal state $x_m(t)$ represents the expected conditions on its associated freeway section

m, nothing prevents us from adding more inputs and / or hidden neurons to solve the above mentioned limitations. The thing required is proper training with enough historical data to represent the input conditions which might occur in real-time. Below we will illustrate how this could be done for the three input factors listed above.

9.2.2 Accounting for traffic conditions elsewhere in the network

We can account for the fact if for example spill back from a connected freeway is the predominant cause of congestion on the route of interest, by simply adding (an) extra neuron(s) to the hidden layer of the SSNN. Consider the example in fig. 9.1.

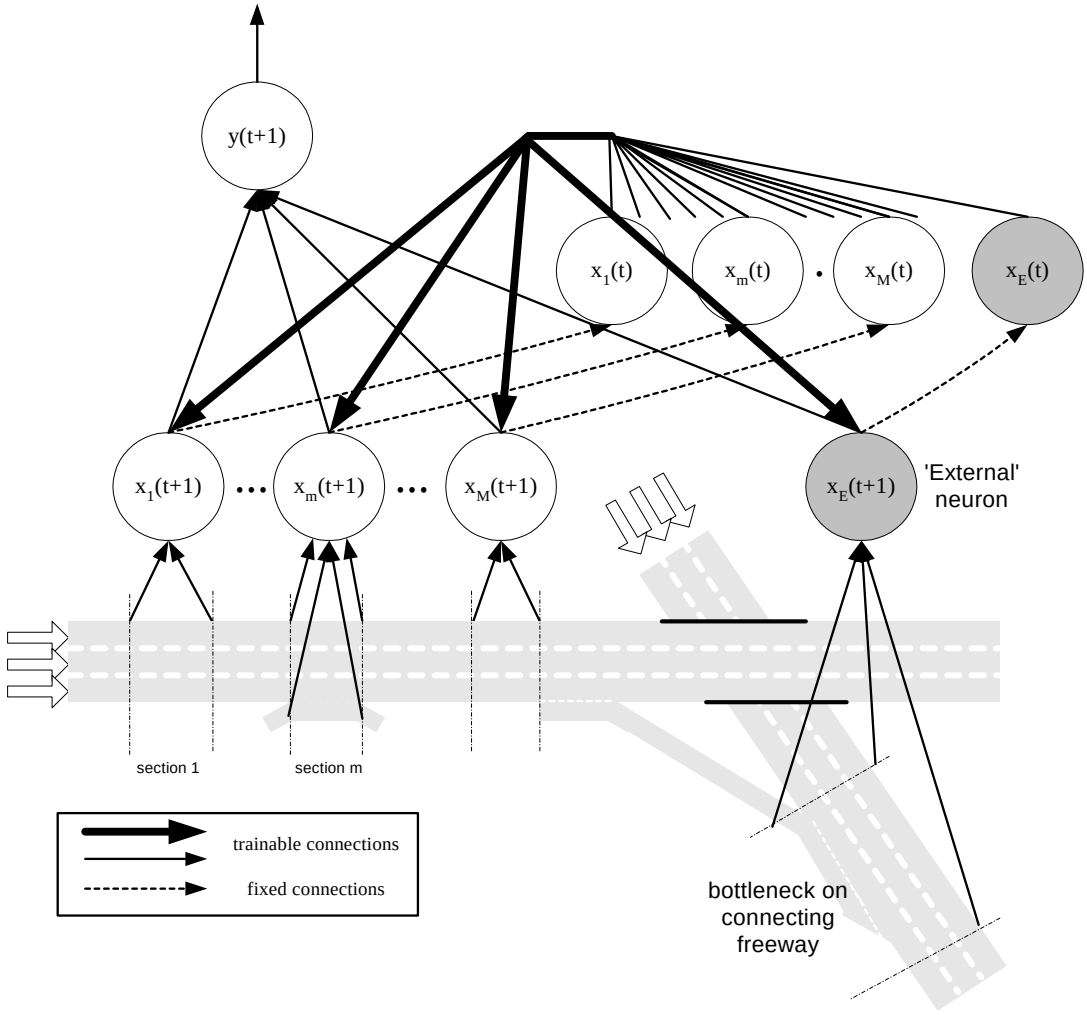


Figure 9.1: SSNN configuration for freeway route where delay (congestion) is partially due to a bottleneck on a connecting freeway.

In this case an extra neuron $x_E(t)$ is added to the hidden layer, which is fed with input (flows, speeds, other) related to the bottleneck on the connecting freeway, which causes queue spillback and hence delay on the freeway route of interest. Also an extra

corresponding neuron is added to the context layer, since the relationship of this extra internal state to the other internal states should be considered dynamic. The bottleneck may induce extra delay on the route of interest, but vice versa, the traffic conditions (outflow at the on-ramp) on the route of interest may also induce oversaturation at the bottleneck. In principle, this extended SSNN topology operates in exactly the same way as before. Note that the SSNN designer may add as many extra hidden neurons as he or she believes is required for the freeway stretch under consideration.

9.2.3 Accounting for the effect of weather

In a similar way we can account for the effect of weather on mean travel time by adding a "specialized" neuron to the hidden layer. Suppose we have a weather station recording a number of characteristic weather quantities on the route of interest (e.g. type and quantity of precipitation and visibility), as in fig. 9.2. Let $\pi\ (t)$ denote a vector of these quantities at time t. Since $\pi\ (t)$ most likely affects the traffic conditions on the entire route we can adopt the same approach as above, that is, we add an extra neuron to the hidden layer that captures weather and ambient conditions.

$$x_W(t) = \phi\left(\pi\ (t)\ , \chi\right)$$

in which χ denotes the input weight vector for that neuron. Since the internal states (representing expected delays) do not affect this 'weather' neuron we do not add a corresponding neuron and recurrent connection to the context layer. We can, however, create a context neuron $x_W(t-1)$ exclusively for the "weather neuron" to capture the possible dynamic effect of weather. For example, after a heavy shower or rainfall, the road-surface may remain wet and slippery for a number of periods.

We can either feed the other neurons with $x_W(t)$ or connect the weather neuron to the output neuron directly. The former approach lets the SSNN figure out the effect of weather on each internal state, while the latter only captures the net effect of weather on the SSNN output. Finally note that $\pi\ (t)$ may (or preferably should) contain forecasts of weather or ambient conditions.

9.2.4 Accounting for the effect of traffic control

As outlined in chapter 2, both Advanced Traffic Management Systems (ATMS) and Information Systems (ATIS) may potentially affect the travel time on a freeway route. In chapter 7 we already found that confronting the SSNN with traffic conditions under a typical ATMS measure (dynamic speed limits) seriously changes the corresponding travel times and leads to poor performance if the SSNN has not been trained with data in which ATMS were applied. We can explicitly account for the affect of ATMS and ATIS by incorporating these as extra inputs to the SSNN. As an example we take the dynamic speed-limits from chapter 7.

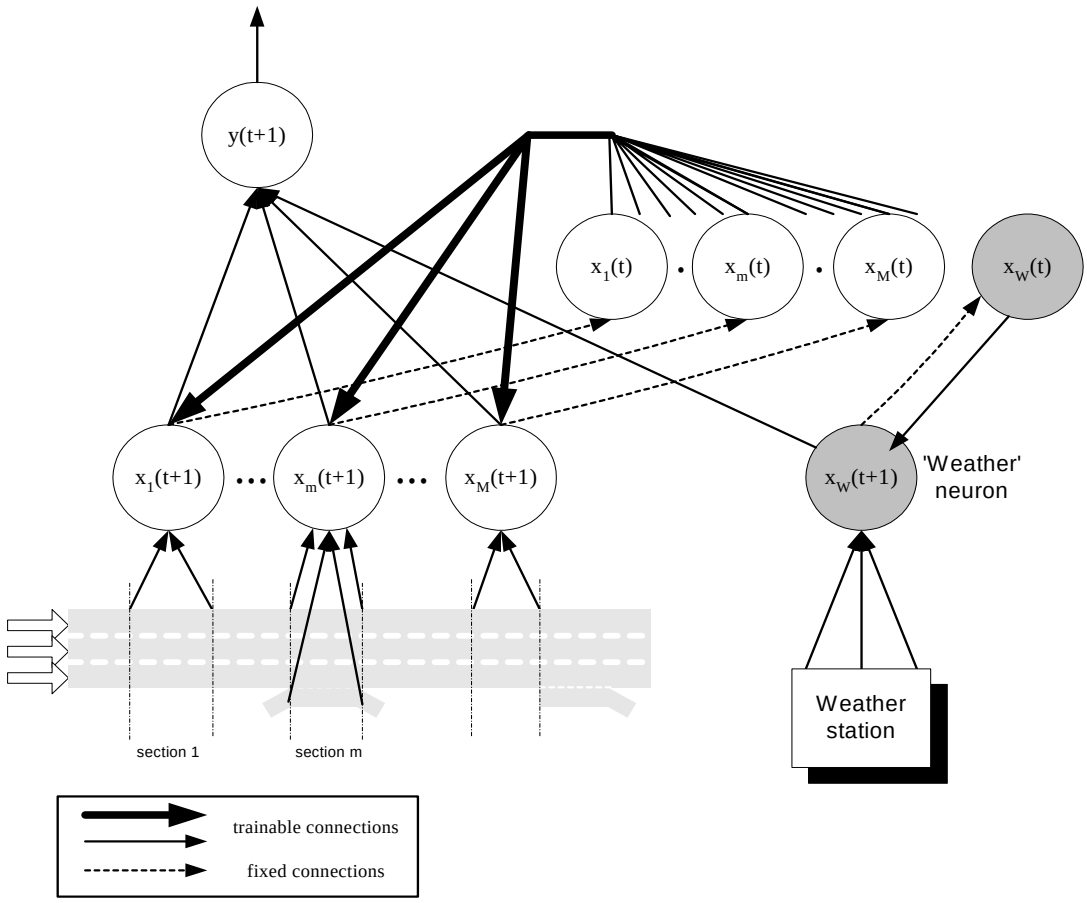


Figure 9.2: SSNN configuration with added neuron for weather and ambient conditions. In this case this 'weather' neuron outputs connects directly to the outputlayer, however, it could alternatively (fully) connect to the hidden layer.

Let $\gamma(t)$ denote a vector of dynamic speed-limits applied for each of the M sections constituting a freeway route. Note that the section specific speed-limits $\gamma_m(t)$ can be either scalars or lane-specific vectors, i.e. $\gamma_m(t) = \{\gamma_{m1}(t), ..., \gamma_{mL}(t)\}$, where L denotes the number of lanes on section m. The speed-limits can be fed to the SSNN as an extra input vector yielding for the internal states

$$\mathbf{x}(t) = \Theta(v_0 + v\mathbf{u}(t) + \mu\gamma(t) + \nu\mathbf{x}(t-1))$$

Again this extra input does not change the SSNN operation nor training. Lane control (e.g. closure) could be modelled similarly, e.g. by letting

$$\gamma_{m1}(t) = \begin{cases} 0 & \text{lane closed} \\ v_{\text{lim}} = \{..., 50, 70, 90, 100, ...\} & \text{dynamic speed limit} \end{cases}$$

Like with the other inputs $\mathbf{u}(t)$, we scale $\gamma(t)$ in $[0.1, 0.9]$ for faster and more stable learning. Other ATMS control measures (e.g. ramp metering) that affect the traffic

flow on sections directly can be modelled in a similar fashion. Measures that have an (hypothesized or actual) effect on route level are better suited with a 'specialized' neuron, as was the case in the previous two sections. Apriori, since the SSNN is a data driven method, we expect the SSNN to learn the (non-linear) relationships between traffic flow, traffic control measures and travel time provided sufficient input, output data are available.

Alternatively a second, and much smaller neural network type model (could also be a SSNN) could be designed, which takes as inputs $\gamma(t)$ and the SSNN output (prediction) and corrects that output based on the applied traffic control. In this case the SSNN is left as is, and a specialized network models traffic control.

9.2.5 Using the SSNN with data from different traffic data collection systems

In this thesis, the topology of the SSNN was based mainly on the configuration of local (inductive loop) detection equipment. We argue, however, that it is straightforward also to apply the SSNN model on data from different traffic data collection systems, given a proper state space representation can be formulated. As noted in chapter 1 the SSNN framework can be applied on a particular freeway route given that

1. A traffic data collection system along the route of interest is installed.
2. Actual (mean) travel times (per departure time period) along the route are either measured or estimated from data.
3. A sufficiently large historical database is compiled of input and output data (travel times) per departure time period.

Below, we briefly discuss the characteristics of other traffic data collection systems than the dual loop based system used in the previous chapter, with respect to application of the SSNN model.

Automatic vehicle identification (AVI) systems

In this category fall systems that recognize and record a particular vehicle on locations A and B, and consequently derive the travel time (and mean journey speed) of that vehicle along A and B. These systems can either be camera based (using advanced license plate recognition software) or based on for example transponders. In case the AVI systems are located at short distances from each other, an SSNN model can be designed very similar to the one based on inductive loops. In this case the freeway can be subdivided in sections enclosed by two AVIs. Subsequently a choice has to be made for the appropriate temporal resolution. Similar to the SSNN based on dual loops this

could be set to for example one minute. As section specific inputs one could choose the number of vehicles recorded in that minute, their mean section level travel time, and the travel time variance. Another appropriate choice would be the space mean speed (mean journey speed) and speed variance. Taking into account speed or travel time variance is beneficial in detecting onset and dissolve of congestion (Hoogendoorn 1999). Since an AVI system also measures route level travel times, all ingredients are available for compiling a large database for training and testing the SSNN model. Given the AVI detectors are relatively densely spaced along the route, they probably constitute the most appropriate detection system for short term travel time prediction.

However, as the distance between the AVI detectors becomes larger, or if very few are installed on a longer route, the 'value' of the state space approach as proposed in this thesis may also vanish. In that case advanced feedforward approaches as in (Park & Rilett 1998), (Park et al. 1999) are an appropriate alternative. Note that - counter-intuitively - systems that *measure travel times on entire routes* only, may not necessarily be the ideal data collection system for travel time prediction purposes, since measurements (realized travel times) of those systems in fact reflect past traffic conditions rather than current, especially on longer routes.

Floating car data (GSM / GPS)

Floating car data (FCD) potentially delivers traffic data (individual positions and speed at high temporal resolutions) in the most detailed format available. Although many (also commercial) parties have advocated FCD in the last decade, and a number of actual pilots were succesfully executed (e.g. (Jochem et al. 1998), (Huber et al. 1997)), few actual FCD based traffic data collection are operational today. This is among other things due to the fact that for FCD systems to be suitable as data collection source for ATIS, there are requirements to be met in terms of the number of probe vehicles in operation[1]. In (Cheu et al. 2002), for example, it is stated that *"to achieve an absolute error in the estimated average link speed of less than 5 km/hr at least 95% of the time, results indicate that there needs to be 4% to 5% active probe vehicles in the total network volume"*, reflecting results obtained through simulation on arterial roads. In (Brackstone et al. 2001), the authors state that "*about one vehicle in every 417, or about 0.24% of the vehicle population using that road are required"* for freeway state estimation (in this case speed), that is, given the fairly unrealistic assumption that the FCD population is uniformly distributed (over both time and space) on the freeway of interest. In as early as 1991, (Boyce et al. 1991) estimated necessary sample sizes for a dynamic route guidance model at 4,000 probe vehicles for a 520 squared kilometer-sized urban network, while (Srinivasan & Jovanis 1996) published similar results, with the note that the number of required probes increased non-linearly as the reliability

[1] a minimum percentage of probe vehicles is not only required for technical reasons (e.g. accuracy and availability of section level travel time estimates), but also for commercial reasons (critical mass for ATIS services)

criterion grew more stringent. Finally, (Huber et al. 1997) reports that for freeway travel time information *"proportions varying from 1 to 5 % depending on what level of quality of traffic information is required"*. The bottom line is that the number of probes required greatly depends on both application and network characteristics.

In our view, greater potential lies in combining different data collection systems for estimating the traffic state (in terms of mean speeds, travel times) for ATIS on both freeway and urban networks (e.g. (Kuhne 1997), (Boker 2000), (Boker & Lunze 2001), (Sariks 1997), appendices D & E). In these cases much lower (and perhaps more realistic) probe vehicle numbers are required. Since most of these studies indicate that fusing data leads to better state estimates than estimates based on a single data source (loops or FCD), we may expect that combining data from different sources in the input to the SSNN model also leads to good results. The requirement of a large (and preferably accurate) database of actual travel times per departure time minute, along with data from the various sources, should still be met, in order to train and validate the model. Since FCD and its potential for ATIS have been studied and demonstrated extensively in the past decade, the obvious recommendation to be made is that an FCD based traffic data collection system should be actually deployed. Due to the fact that this requires investments mainly from industrial parties (e.g. the automobile industry, ICT system integrators and telecom providers), there is, however, no saying as to when and where a first economically and technically viable FCD system will become operational.

9.3 Some Notes on Travel Time Prediction for Urban Networks

From a modelling point of view, there are a number of major differences between freeway and urban road facilities. First and most importantly, in urban networks traffic is controlled (e.g. through traffic lights). Depending on the controller scheme and the degree of saturation, the delays at intersections constitute a very significant part of the travel time on urban networks and also in the uncertainty around that travel time. The travel time prediction problem in urban networks thus is very closely related to modelling delays at intersections, which in congested conditions and certainly in case of coordinated traffic control on larger urban corridors, is a field in which still much progress has to be made. For example, due to the stochasticity of the queueing process, intersection delays may occur that are an order in magnitude higher than the delays calculated with widely used deterministic models ((Viti & Van Zuylen 2004), (Van Zuylen & Viti 2003)), for example those listed in the Highway Capacity Manual (Transportation Research Board 2000). In general, traffic control schemes classify as fixed, semi-fixed (time tabled), or vehicle actuated. On a network level, traffic control systems can be coordinated (e.g. providing green waves along corridors) or isolated. There are many different systems that (adaptive) optimize cycle times and green times

in urban networks, both for isolated intersections as well as coordinated for entire networks (for example Scoot (Bretherton et al. 2003), Utopia SPOT (Turksma 2001), and Integrated Traffic Urban Control (In-TUC) (Diakaki et al. 2000)). In sum, travel times in urban networks are strongly related to intersection delays, which are a product of intersection control, which may in turn be based or even optimized on prevailing (network!) traffic conditions, which result in the travel times we wish to predict in the first place. Thus, similarly to freeway travel time prediction, urban travel time prediction is a very complex non-linear spatio temporal problem.

Secondly, since in an urban network there is an interplay between traffic streams from opposite or conflicting directions, observations from different locations are much stronger correlated than in a freeway network. More generally, the spatiotemporal characteristics of urban (controlled) traffic are very different than the spatiotemporal characteristics of freeway traffic. This implies that a model taking into account only data from the route of interest may not be sufficient to predict travel times on that route. Predicting travel time on some urban route may involve measurements from locations in very different parts of the network. On the other hand, since travel times are influenced largely by intersection control, simpler solutions may suffice to predict travel times between intersections. In the simplest case the travel time on an urban route equals the free flow travel plus the expected delay at intersections. An example of such an algorithm is deployed in Rotterdam (on the so-called Maastunnel traverse (Tampere et al. 1999)) where travel time predictions are presented to drivers on a VMS at the beginning of the route. Although the algorithm in principle could be classified as an instantaneous predictor (since it uses an estimate of intersection delays at the current time instant), the results so far indicate that the assumption of stationarity may not be that far off in this particular situation for short time periods of 10 to 15 minutes. On the negative side, the flow based algorithm is very sensitive to data idiosyncrasies, since it depends on vehicle conservation (see chapter 3). A tentative conclusion could be that travel times on urban routes may be less dynamic (time varying) than on freeway routes. However, urban travel times are likely to be more influenced by spatial traffic patterns elsewhere in the network than is the case in a freeway network, since an urban traffic network is much denser and traffic is much more dispersed than a freeway network.

Finally, a more practical difference is in the traffic data collection systems used in urban networks. Urban networks usually are not as comprehensively covered by measurement equipment than freeway networks. Larger parts of urban networks may not be covered at all. Despite the (often advocated) unlimited possibilities of ICT technologies (WAP, GSM, GPRS, GPS, UMTS, DAB, Bluetooth, RDS-TMS) for traffic information dissemination purposes and the large scale implementation of advanced driver assistance systems (e.g. route navigation) in many new vehicles, the real-time content (traffic information) needed to feed these systems usually comes to an abrupt halt at freeway exits to urban (or rural) areas. In most cases, the traffic data collection systems that are available are tailor made for controlling intersections or for example

tunnel surveillance and not for comprehensive traffic monitoring at a network level. Apart from probable data quality issues, in most cases no ICT facilities exist for using these data for other purposes than feeding local controllers. Moreover, since mostly only major urban routes are equipped, many "black spots" on the route boundaries exists where simply no detection equipment is installed.

Nonetheless, in the last decade, also in urban traffic monitoring some major developments have taken place. As a small scale example, within the Regiolab Delft project a provincial road (the Kruithuisweg - (Van Zuylen & Muller 2002)) is equipped with a video-based license plate recognition system, which provides travel times in between camera locations. On the same route, real-time data from all controlled intersections on that is stored, predominantly single loop detection equipment, measuring cumulative flow within one cycle time. From these multiple data sources a very detailed picture is obtained of the traffic conditions on that route. A larger scale example is the "intermezzo" project commissioned by the Dutch Ministry of Transport, Public Works and Water Management in close cooperation with provincial and municipal authorities in the provinces of South Holland and Brabant in 2002. In this project traffic public-private traffic monitoring projects in which non-freeway traffic conditions are estimated with GPS equipped probes and GSM based location tracking systems are explored.

Due to the differences outlined above, we argue the SSNN approach in its current form is not an obvious choice for urban travel time prediction. More research is neccesary into models that can be applied for travel time prediction on urban networks.

9.4 Improving Robustness and Reliability

In this section we suggest an online correction algorithm that allows for automatic correction of SSNN travel time predictions based on the last known (offline estimated or actually measured) travel time. Since the offline PLSB travel time estimator can calculate travel times as soon as all the required traffic measurements along a route are available, it offers an extra possibility to both online correct travel time prediction as well as assess the reliability and quality of the travel time predictions.

9.4.1 Online correction algorithm

Suppose at some departure time period $p_0 - n$ a full trajectory can be constructed and an offline PLSB estimate can be calculated:

$$\tau_{est}\left(p_0 - n\right) = PLSB\left(\mathbf{u}\left(p_0 - n\right), \ldots, \mathbf{u}\left(p_0\right)\right)$$

For that particular time instant the SSNN model also made a prediction:

$$\tau_{pred}(p_0 - n) = G(\mathbf{u}(p_0 - n - 1), \psi)$$

We can use the prediction error at a given time period p $e(p) = \tau_{pred}(p) - \tau_{est}(p)$ to adjust the SSNN travel time prediction in a Kalman-like manner (see appendix D). In this case the SSNN prediction becomes

$$\tau_{pred}(p) = G(\mathbf{u}(p-1), \psi) + e(p)$$

Since $e(p)$ is not available (for any reasonable sized route), we need to correct the current prediction based on past errors. Let

$$\tau_{est}(p) = \left[\tau_{est}(p), ..., \tau_{est}(p-N)\right]^T$$

denote a column vector of the last N offline PLSB estimates, where N is a sufficiently large number, and let

$$\tau_{pred}(p) = \left[\tau_{pred}(p), ..., \tau_{pred}(p-N)\right]^T$$

depict a column vector of the last N SSNN travel time predictions. Now define a state vector as the SSNN travel time predictions of the last N time periods

$$\mathbf{x}(p) = \mathbf{A}\tau_{pred}(p) + \xi(p)$$

where $\xi(p) = \left[\xi(p), ..., \xi(p-N)\right]^T$ denotes a known zero mean white noise error vector, and $\mathbf{A}$ is an $N \times N$ matrix. In the simplest case A is a unity matrix, but other choices are possible, which would render the state equation into a AR type process (in which the current state is a linear combination of current and past predictions). For the noise component we could for example use the SSNN model variance σ^2_{ψ} defined in chapter 7 and set $\xi(p) \sim N(0, \sigma^2_{\psi}(p))$. The state noise in that case equals the model confidence component defined in chapters 7 and 8. The system output equation simply equals the vector of prediction errors

$$\mathbf{y}(p) = \mathbf{C}\left[\mathbf{x}(p) - \tau_{est}(p)\right] + \zeta(p)$$

where $\mathbf{C}$ is an $N \times N$ matrix, which in the simplest case is a unity matrix. Other choices would effectively convert the output equation into an MA procedure, in which the outputs are a linear combination of current and past errors. The output noise vector $\zeta(p) = \left[\varsigma(p), ..., \varsigma(p-N)\right]^T$ is a zero mean white noise vector, that can be thought of as the error the PLSB procedure makes with respect to real (but unobserved) mean travel times. Again we could use the findings in chapter 7 and set $\varsigma(p) \sim N(0, \sigma^2_P(p))$, where $\sigma^2_P(p)$ is the variance of the PLSB estimation error (which can only be estimated if a data set of actually measured travel times is available). Since PLSB estimated travel times are available only for time periods $\leq p_0 - n$

the errors for time periods $> p_0 - n$ are unobserved. In this case, the output is considered unknown, meaning that the Kalman filter will consider the SSNN model prediction 100% reliable. Practically in these cases we set $\tau_{est}(p) = \tau_{pred}(p)$.

Furthermore, it is assumed that $\xi(p)$ and $\zeta(p)$ have a known covariance structure

$$\begin{aligned} \left\langle \xi(p)\xi(l)^T \right\rangle &= \mathbf{S}(p)\delta_{pl} \\ \left\langle \zeta(p)\zeta(l)^T \right\rangle &= \mathbf{R}(p)\delta_{pl} \\ \left\langle \xi(p)\zeta(l)^T \right\rangle &= \mathbf{T}(p)\delta_{pl} \\ \delta_{kl} &= \begin{cases} 1 & p = l \\ 0 & otherwise \end{cases} \end{aligned}$$

in which $\mathbf{S}(p)$ and $\mathbf{R}(p)$ are nonnegative and positive definite matrices respectively. Furthermore, the state $\mathbf{x}(0)$ is uncorrelated to both $\xi(0)$ and $\zeta(0)$ and considered a Gaussian random variate with mean

$$\widehat{\mathbf{x}}(0) = \langle \mathbf{x}(0) \rangle$$

and covariance matrix

$$\Sigma(0) = \left\langle [\mathbf{x}(0) - \widehat{\mathbf{x}}(0)] [\mathbf{x}(0) - \widehat{\mathbf{x}}(0)]^T \right\rangle$$

Based on these conditions, we can now apply the same Kalman Filtering algorithm as used in chapter 6 for tackling the missing data problem. In the first step the next state is predicted. This simply means applying the SSNN model as in the previous chapters. The second step then corrects this prediction based on the past errors. The algorithm is listed in detail in appendix D. For the algorithm to work and produce stable corrections, the proper setting of the variance covariance matrices $\mathbf{S}(p)$, $\mathbf{T}(p)$ and $\mathbf{R}(p)$ of the state and output vector is crucial. These matrices govern to which extent past errors are used to correct current predictions. Also, the model designer could experiment with different settings for matrices $\mathbf{A}$ and $\mathbf{C}$ which would essentially transform the online correction algorithm into an autoregressive (AR) type model.

9.4.2 Implications of online correction algorithm

An online correction algorithm would provide the automatic means to adapt the SSNN predictions to the last known (offline estimated) travel times. In this sense it could improve the predictive accuracy, robustness and reliability. Since the correction mechanism does not alter the current SSNN parameters, the model confidence component can still be used as an indicator for deteriorating predictive performance (see chapter 7). Practically, as the SSNN performance is below some desired threshold, a new model can be trained (with an updated training data set) in the background, while the old one can still be safely used and produce valid predictions in combination with the online Kalman filtering correction algorithm.

9.5 Long term Travel time Prediction

Finally, we conclude with some thoughts on longer term travel time prediction. As we outlined in chapter 2, the longer the prediction horizon, the less we can use current traffic conditions as inputs to our travel time prediction models, and the more we need to rely on modelling assumptions. Since the SSNN model uses current conditions (possibly augmented with other factors as shown in this chapter) as its inputs, it is not suitable for longer term travel time prediction in its current form.

One of the possible routes to predict travel times in the long term is by using historical travel time profiles (for example (Boyce et al. 1993), (Chien & Kuchipudi 2003)). A historical profile contains mean or percentile travel time values for a given time-of-day (TOD) and day-of-the-week (DOW) and possibly month-of-the-year (MOY). In fig. 9.3 historical profiles on 15 minute TOD periods on Thursdays, Fridays and Saturdays are show for the freeway route used throughout this thesis. Note that the profiles are based on PLSB estimated travel times from inductive loop data for 2001 and 2002. On Thursday afternoon peaks between 16:00 and 16:15 in 50% of the cases travel times of 24 minutes occurred. However, in less than 5% of the cases (that is Thursdays between 16:00 and 16:15) no congestion occurred at all (5th percentile value) while in another 5% of the cases travel times occurred of 40 minutes and more (95th percentile). Given the bandwidth of travel times the day-to-day variation of travel times on this road stretch is large. It is, however, conceivable to predict different characteristic travel time percentiles (say the 10th, 50th and 90th) from a large database of historical profiles. As inputs one could start with for example a combination of TOD and DOW[2] indicators, and - if available - weather conditions or any of the other factors influencing travel times. In (Van Lint et al. 2004) an example of such a model is proposed, producing almost unbiased predictive results for a comparable freeway stretch.

As outlined in chapter 2 the interrelationships between all factors influencing travel time must be considered non-linear, stochastic and dynamic, which make ANN models also an appropriate choice for the long term travel time prediction problem. Furthermore, the tools to explicitly calculate the relevance of particular neurons or inputs to the ANN output (see chapter 5), and to quantitative the uncertainty in the ANN prediction can also be readily used in this case. Finally, ANNs are - after training - computationally very efficient models, which makes them suitable for online usage (e.g. internet-based route-planners).

9.6 Summary

In this chapter we outlined possible extensions to the SSNN framework for short term travel time prediction, that solve some of its current limitations. Specialized neurons

[2] when chosing 15 minute TOD intervals there are $24 * 4 * 7 = 672$ unique TOD/DOW week periods

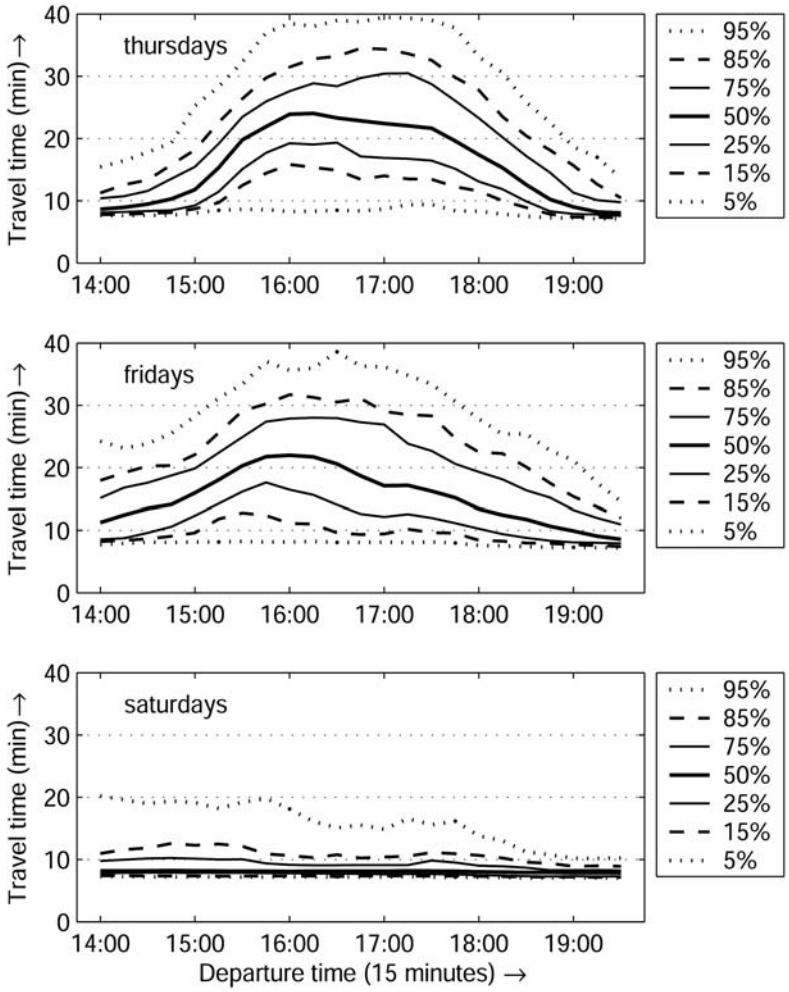


Figure 9.3: Travel time profiles (for different percentiles) on three days on the A13 Southbound from The Hague to Rotterdam (The Netherlands) estimated on loopdata from theyears 2001 and 2002.

can be added to capture the effect of traffic conditions elsewhere on the network that affect the travel time on the route of interest, for example due to queue spillback . Similarly, a specialized neuron could be added to capture the effect of weather or other ambient conditions on travel time. Traffic control can be modelled as an extra (section or lane specific) input to the SSNNs section specific neurons. In principle, the SSNN framework can be applied in case other types of traffic data collection systems are available, for example automatic vehicle identification systems (AVIs) or floating car data (FCD) based systems. In both cases succesfull application depends on characteristics of the system installed, particularly the density of AVI posts and the penetration (percentage) of equipped vehicles (FCD) alonbg the freeway route of interest.

Also, we briefly discussed travel time prediction on urban networks. Due to the differences between freeway and urban traffic flow and differences between traffic data collection systems usually applied on either type of network, application of the SSNN framework is not an obvious modelling approach. Next, we presented a possible online

correction algorithm, that allows the SSNN framework to adjust its performance based on the last available offline travel time estimate. The algorithm is based on a Kalman filter.

We concluded with some thoughts on long term travel time prediction which is a very different type of problem than the short term travel time prediction problem on which this dissertation thesis focussed. As such the SSNN model may not be suitable for this problem. A neural network approach based on travel time historical profiles seems a promising approach.

Chapter 10

Conclusions & Recommendations

In the final chapter of this dissertation thesis we summarize the main conclusions and results. Furthermore, we make some recommendations based on the findings of this thesis for both practitioners and scientists in the field of traffic and transport. We conclude this chapter with directions for future research.

10.1 Conclusions

This dissertation thesis presented a reliable framework for short term travel time prediction on freeways. This framework is based on a so-called state space neural networks (SSNN) for travel time prediction. We showed the SSNN performance is comparable or slightly better than a number of state-of-the-art travel time prediction models found in literature, and offers significant improvement in terms of robustness to missing or corrupt input data. Moreover, the SSNN model outperforms current travel time prediction models in Dutch practice by far, reducing mean relative error *more than twenty times* and reducing variance over three times in a real life test case over 118 six-hour afternoon peak periods.

We divide the conclusions into five parts. The first part contains general conclusions that summarize the most relevant conclusions from all chapters. The other four parts pertain to the findings of chapters 3 (travel time estimation), 4 (travel time prediction, State-of-the-Art), 5 (the state space neural network model), and 6 & 7 (robustness and reliability). The findings of chapter 8, which evaluates all models and algorithms in a real test case, are listed in the appropriate subsections below.

10.1.1 General conclusions

1. Artificial neural networks (ANN), and particularly state space neural networks (SSNN) are appropriate and accurate models for short term prediction of traffic conditions on freeways and travel times in particular.

2. On the basis of currently available data it is possible to predict reliable travel times for on-trip advanced traffic information systems (ATIS), such as variable message signs (VMSs).

3. The SSNN framework we propose is practical and generic in that it can be applied on any freeway route equipped with a traffic data collection system, given a historical database of measurements and actual travel times is available for calibration and validation. Design of the SSNN model is straightforward and based on the geometry of the freeway route.

4. In case no actual travel times are available, a novel speed based offline travel time estimation tool (the PLSB method) can be used to gather large databases of travel times on freeway routes equipped with inductive loops. This makes the SSNN framework both portable and generic, that is, it can be applied even if no actual travel times are available, but an accurate travel time estimation method (e.g. the PLSB method) can be used.

5. The SSNN framework is indeed reliable, that is robust with respect to missing data, accurate, valid and adaptive. It deals adequately with both structural and random input failure percentages of 10 to 20%, without loss of predictive accuracy. The model is adaptive in the sense that given its mathematical structure stays in tact, its parameters can easily be retrained as circumstances require this. Finally, it offers a quantitative measure to monitor reliability of the entire framework, including the input data with which the SSNN model is fed.

10.1.2 Traffic data analysis and freeway travel time estimation

1. Travel time estimation (although sometimes misused as a synonym for prediction) is the translation of other traffic variables (e.g. mean speeds, flows, densities or occupancies) into section or route travel times. Theoretically, there are two main (non-parameterized!) approaches. The first uses cumulative vehicle counts, the second uses space mean speeds. Both have its merits and disadvantages.

   (a) Speed based methods are intuitively appealing given the direct relation between mean speed and mean travel time and are not very sensitive (robust) to vehicle miscounts- or false counts since they are based on mean quantities. They suffer, however, from the fact that low speeds are difficult to measure with local detection equipment

   (b) Flow-based methods are applicable in all traffic conditions, are simple to implement but very sensitive to miscounts- and false counts at local detectors. Given the requirement of robustness we chose speed based methods throughout this dissertation thesis. Nonetheless, if proper care is taken

to account for data idiosyncrasies, flow-based methods could be easily plugged in the framework.

2. Arithmetic mean speeds cause biased estimates of travel time. Therefore local detection equipment should record harmonic mean speeds whenever possible. We propose, however, means to correct for this bias, which require estimation of local speed variance with the quantities that are measured. We propose a method based on time series of arithmetic mean speeds and so-called local density, which almost completely accounts for the bias. However, the method does not decrease variance (random errors) of the travel time estimation procedure.

3. As an extension of the widely used Piece-wise Constant speed based (PCSB) Trajectory Method, we propose the Piece-wise Linear speed based (PLSB) Trajectory Method for freeway travel time estimation. The PLSB method reduces both bias and variance in comparison with the PCSB method, however, both are based on the assumption of stationary and homogeneous traffic conditions in space (between detectors) and time (single measurement periods), which in real life traffic may not always a realistic assumption.

10.1.3 The short term travel time prediction problem

1. Short term freeway travel time prediction is a complex spatiotemporal problem, for which we require either sophisticated traffic flow models and prediction algorithms for the boundary conditions, or intelligent inductive models that are able to learn the complex traffic dynamics from data on the route of interest directly. Both methodologies have their merits and disadvantages.

   (a) Model based approaches allow for in depth analysis in the causes and effects of possible delays and allow a modeler to include for example traffic control measures that effect travel time. They suffer, however, from the fact they require (predicted) inputs on the boundaries of the freeway stretch of interest. The predictive quality of a model based approach hence can only be as good as the predictive quality of its inputs.

   (b) Data driven approaches do not require apriori knowledge of traffic processes and allow for direct prediction of travel times based on all current and near-past input factors which are measured. On the down side, they often require tedious input- and model selection procedures and often result in location specific solutions which are neither portable nor general.

2. In current practice (at least in the Dutch situation) a third class of models, so-called instantaneous predictors are used. These instantaneous predictors assume stationary conditions for an indefinite time period. As such they transform the travel time prediction problem into a travel time estimation problem. Although

their mathematical simplicity and computational efficiency makes them very suitable for real-time use, they seriously suffer from the fact the stationarity does not hold on any practically sized freeway route in congested conditions, which is when accurate travel time forecasts are most desirable.

10.1.4 State space neural networks for short term freeway travel time prediction

1. Dynamic processes are generally modelled in two different ways in artificial neural networks (ANNs):

   (a) By means of an input shift-register, in which case time series are treated as fixed length input vectors. There are many successful applications of such feed-forward neural networks (FNN) reported, in traffic prediction, but also in many other domains. Nonetheless, these models suffer from two problems. The first (and most serious) is that such a representation of temporal patterns may seriously increase the complexity of the problem to be solved (*the semantic problem*), the second is that it is generally difficult to select the appropriate inputs from the appropriate time lags, potentially leading to sub-optimal models (*the input selection problem*).

   (b) By means of short term memories that store past neuron activities and allow the ANN to develop a dynamic (spatiotemporal) mapping. These so-called recurrent neural networks (RNN) are generally used in complex problems such as speech and voice recognition, automated guidance and process control. They suffer from the fact that they are more difficult to train and less stable in use than feed-forward models.

2. Given the complexity and spatiotemporal dynamics of the short term freeway travel time prediction problem we conclude that the class of RNNs is most suitable to tackle it. In the first place because for an FNN approach it is very difficult to apriori select the appropriate inputs and time lags from detectors along a freeway route and secondly, because of the analogy of the RNN solution chosen (state space neural network) with traffic flow theory. Formulating the freeway travel time prediction problem in state space form (in line with macroscopic traffic flow models) leads to a general class of recurrent neural networks evolves, the so-called state space neural network (SSNN), which have been widely used in the past decade in numerous application fields. The temporal dynamics are dealt with by means of a short term memory, which allows the SSNN to predict travel time based on current measurements in the context of its previous internal states. In many respects, the SSNN operates like a macroscopic traffic flow model. For its design, only the geometry and detector configuration of the freeway route of interest are required, alleviating the model designer of tedious input selection procedures common to neural network design.

3. On the basis of both synthetic data (chapter 5) and real data (chapter 8) it is shown that the SSNN is an accurate freeway travel time prediction model.

   (a) It outperforms current models from Dutch practice by far, reducing mean relative error *more than twenty times* and reducing variance over three times in a real life test case over 118 six-hour afternoon peak periods.

   (b) Its results are comparable with or better than a number state-of-the-art models reported in literature.

4. By using a Bayesian regulated backpropagation algorithm for training, the parameters of the SSNN consequently are set in a very efficient way that is closely related to the actual traffic processes that determine travel times. The Bayesian method provides a unified and mathematically sound way to avoid overfitting and automatically prune the network of irrelevant parameters and as a bonus, provides for quantitative information on the uncertainty in each prediction.

5. We also propose a measure to calculate the relevance of each individual neuron and input signal and found, based on the back propagation training algorithm. The advantage of this heuristic over other relevance measures, is that it can be calculated directly for a (regularized) neural network, instead of iteratively. Using this heuristic we can conclude

   (a) The state space structure of the SSNN offers sufficient memory depth to capture the dynamics of the freeway travel time prediction problem.

   (b) The SSNN classifies traffic patterns in a manner very similar to for example three phase traffic theory (Kerner 1999). Those freeway sections where the most severe congestion occurs (the general congested pattern) are consequently selected as the most relevant neurons. As such, the internal states of the SSNN can be explicitly interpreted as indicators of which freeway sections contribute the most to the delay and also as indicators of bottlenecks on the route of interest.

   (c) The relevance heuristic also enables the model designer to rationally reduce the SSNN model if required by removing those neurons that are classified as irrelevant.

10.1.5 Robustness & reliability

1. Robustness to missing data is a key requirement for any travel time prediction model applied in a real-time environment. Real-time traffic data collection system exhibit detector failure due to temporal power or communication problems (incidental failure), structural power or communication problems, e.g. incidents and accidents or maintenance backlogs (structural failure) and also suffer from intrinsic problems such as miscounts- and false counts and for example arithmetic averaging of speeds.

2. Input failure affects the framework in two ways:

   (a) It affects (deteriorates) the quality of the data (PLSB estimated travel times for targets and speeds and flows as inputs) with which the SSNN is trained. Not tackling the missing data problem for at least the target values (travel times) means learning the SSNN model "the wrong thing"

   (b) it affects (deteriorates) real-time operation of the SSNN

3. We identified four strands of approaches to tackle the missing data problem, that is, null imputation (do nothing or fill gaps with some default value), simple imputation (fill gaps with sensible replacements such as regression forecasts), model based imputation (in combination with for example Kalman filters) and multiple imputation (creating multiple plausible input data sets).

   (a) For both items **2a** and **2b** it appeared that simple imputation methods lead to the most robust framework. For example, even at 40% incidental input failure, a simple exponential filter procedure reconstructs the input data such that the SSNN model can still predict travel times as accurately as with 100% clean data (increase of RMSEP from 8 to 9% only).

   (b) Alternatively, it is possible to train the SSNN with missing input data. This does improve robustness largely, albeit at the cost of predictive performance. This is due to the fact this inherently makes the problem to be solved (travel time prediction) more complex.

4. We chose to express reliability in a statistical sense, that is, by means of confidence and prediction intervals. The first reflects the uncertainty in the model's parameters, the second all uncertainty, including the variance in the output signal. Since the model *predicts* travel time the predictive distribution must be viewed as a *possible or most plausible* travel time distribution, given the mean prediction. Three conclusions can be drawn

   (a) Since the SSNN may not classify as an unbiased model, the value of these intervals in terms of upper- and lower-bounds to the prediction is often questionable.

   (b) As a result, confidence or prediction interval coverage percentages are not very informative as performance indicators for the short term travel time prediction problem. Instead, high coverage may well indicate large errors. The absolute magnitude of confidence intervals is correlated strongly to prediction errors, that is, they are large as the SSNN makes larger errors, either random (variance) or structural (bias).

   (c) This does, however, provide a very powerful tool (a "warning mechanism"), since it enables a traffic manager to monitor the predictive quality without

having to measure actual travel times. It appears that large confidence intervals in fact coincide with large prediction errors, which occurs for example in case of

i. unknown traffic conditions, for example due to very bad weather or as dynamic speed limits are applied, which were not present in the training data.

ii. missing or corrupted input data, if for some reason the data cleaning procedures did not function correctly or the missing data percentages are too high for accurate travel time prediction

10.2 Recommendations

We divide this subsection into two parts. The first lists recommendations aimed at practitioners in field of traffic and transport, the second part is aimed at scientists and researchers in the field.

10.2.1 Recommendations for practitioners

1. Dual loop detectors can and hence should record harmonic mean speeds, since these are the quantities required for many (offline) analysis, including travel time estimation, capacity estimation, and traffic flow simulation. Recording arithmetic mean speeds (as is the case in the MONICA dual loop system on Dutch freeways) leads to significant bias, that is, overestimation of speeds and hence underestimation of travel times, and is therefore theoretically and practically incorrect, and moreover, computationally much more inefficient than calculating harmonic averaged speeds.

2. We recommend the SSNN framework be applied for presenting travel times on VMS panels throughout the Dutch freeway network. The results here indicate that the SSNN outperforms current (instantaneous) models by far and is much more robust to input failure which is a serious problem within the MONICA system installed on a large part of the Dutch freeway network. To further validate the approach a larger scale database of actual travel times on some of these routes is desirable, particularly to refine and further improve the offline travel time estimation algorithm (the PLSB method).

3. As stated in the previous section, the most relevant and powerful result of the research into the uncertainty associated with the SSNN model is that the model confidence component offers a quantitative indicator for the uncertainty associated with a particular travel time prediction, without having to measure actual travel times. As a result, we recommend confidence levels be used as an automatic and quantitative "warning mechanism", which notifies a traffic manager

whether or not "something is wrong", either with the SSNN model or the data that feeds it.

10.2.2 Recommendations for researchers

1. A better understanding and use of artificial neural network (ANN) type models in traffic and transportation is necessary. We therefore make two recommendations

   (a) For improving generalization, (avoiding under- and overfitting) a wide range of heuristic methods (cross-validation, early-stopping), but also theoretically sound Bayesian techniques are readily available (chapter 7, and appendices B, C). Particularly, Bayesian regularization consistently produces efficient parameter settings regardless of the initial number of parameters in the model. In this sense, overfitting is not a problem inherently associated with ANN models but a problem resulting from using improper ANN training algorithms. We strongly recommend the Bayesian approach be used, whenever this is possible.

   (b) Given that an ANN is trained with a backpropagation type of training algorithm, and proper care has been taken to avoid overfitting (preferably the Bayesian method) it is in fact straightforward to explicitly quantify the contribution (relevance) of each of its parameters and neurons, including the inputs (see chapter 5). Depending on the ANN topology and application specific circumstances we recommend a measure of relevance be developed similar to the one developed in this thesis.

2. We argue there is great potential in combining traffic theory and advanced data driven techniques as the SSNN and recommend researchers from both fields to collaborate more closely. Data driven models can be designed more intelligently using theoretical concepts, but theory could also benefit from features of data driven tools. For example, using the techniques described in this thesis to determine relevance of parameters and input, an SSNN-like model could be used as a rapid prototyping tool in traffic theory or model development.

3. Finally, we found large discrepancies between travel time distributions from a microscopic simulation model real travel time distributions (per departure time period). The predominant difference is that in the simulated data travel time distributions grow wider as traffic is more congested, while in reality the opposite can be observed (the distribution grows smaller as mean travel time increases). Since the microscopic traffic flow model used here[1] has been extensively calibrated with mean speeds and flows measured at Dutch highways in free-flow and near capacity conditions, and contains similar car following and lane changing routines as in many commercial simulation models (e.g. VISSIM, Paramics), we

[1] Freeway Operations SImulation Model (FOSIM), see e.g. (Vermijs & Schuurman 1994)

believe this discrepancy is due to fundamental deficiencies in calibration and validation of microscopic models in general, particularly in congested conditions.

We recommend the parameters (but perhaps more fundamentally the model equations themselves) in microscopic traffic flow simulation models should be calibrated and validated with mean travel times travel time distributions per road segment per departure time period. This will most likely lead to more realistic models of driver behavior in congested traffic.

10.3 Future Research

In this section we discuss some of the perspectives this dissertation thesis offers for further research. We will re-address some of the aspects mentioned in chapter 9 (limitations and extensions to the SSNN framework), but also stipulate other research directions, not directly related to short term freeway travel time prediction.

10.3.1 Research directions related to the SSNN framework

1. This thesis validated the SSNN framework for an inductive loop based traffic data collection system. An interesting and relevant research question is wether similar good results can be obtained using a different type of data collection system, specifically non-infrastructure bound systems (based on GSM / GPS).

2. In chapter 9 we outlined possible extensions to the SSNN framework for short term travel time prediction, that solve some of its current limitations. Further research is needed to assess whether these ideas are indeed applicable and improve the generality and performance of the SSNN model. These include

   (a) Specialized neurons that capture the effect of traffic conditions elsewhere on the network that affect the travel time on the route of interest, for example due to queue spillback .

   (b) Specialized neurons to capture the effect of weather or other ambient conditions on travel time. This may not only lead to a more general travel time prediction model, but also in scientific insight in the effect of weather and ambient conditions on travel time (using e.g. the relevance heuristic).

   (c) Modelling the effect of traffic control by means of extra (section or lane specific) inputs to the SSNNs section specific neurons.

3. We do not believe the SSNN framework in its current form can be applied on urban networks, due to the effect of urban traffic control, differences between freeway and urban traffic flow and different traffic data collection systems used on either type of network. As a consequence, the following research topics need to be addressed:

   (a) Design and evaluation of an urban travel time prediction model

   (b) If no urban travel time measurement systems are available, proper offline travel time estimation tool for urban networks need to be developed and calibrated.

4. We also presented a possible online correction algorithm, that allows the SSNN framework to adjust its performance based on the last available travel time measurement (or offline travel time estimate). The algorithm is based on a Kalman filter. Besides validation of such an algorithm, which is principally external to the model, an interesting option would be to research online learning algorithms that operate directly on the parameters of the SSNN. In (Yang et al. 2004) a similar type neural network (also a SSNN model) for single point traffic prediction is trained online with a so-called temporal difference learning algorithm. The key issue is whether such an online learning algorithm could be designed in conjunction with the Bayesian techniques discussed earlier, since these are essential for obtaining models that are not over-specified and hence general.

10.3.2 Other research directions

1. We concluded that long term travel time prediction is a very different type of problem than the short term travel time prediction problem on which this dissertation thesis focussed. The differences relate to the dynamics of the problem as well as the models and input data required to solve it. As such, development of long term travel time forecasting models is a field which requires more research effort. Approaches based on historical profiles of percentiles values of the distribution of travel times for a given time-of-day / day-of-week seem promising (Van Lint et al. 2004), but also model based approaches (combining traffic simulation and dynamic traffic assignment) have great potential (e.g. (Ben-Akiva 1998)).

2. In the introduction we argued (based on a large body of research) that perception and attitude of individual drivers with respect to traffic information, greatly influences the success of ATIS systems, even if these are based on for example reliable travel time prediction models such as the one developed in this thesis. Therefore more research is needed in the following areas

   (a) Development of a methodology to determine accuracy and reliability requirements for ATIS, as a function of application (VMS, In-car, web service), envisaged user class (commercial vehicles, general public, commuters), network and distribution (commercial, public) characteristics. How reliable must a traffic information service be in relation to these issues and what are the potential effects in terms of individual and collective traffic operations?

(b) Theoretical and empirical knowledge and practical guidelines for presentation of traffic information. For example, we derived methods to quantify uncertainty around travel time predictions and argued this extra information is particularly useful as a monitoring tool for traffic managers. How users respond to explicit information on the reliability of traffic information is not well understood, neither the potential effects on individual and collective traffic operations..

3. Finally, the Bayesian (probabilistic) approach to model fitting is applicable to any (non-linear) parameterized model. Discrete choice models for example (e.g. logit models) have a functional form which closely resembles artificial neural networks ((Hoogendoorn-Lanser & Hoogendoorn 2000) it is argued Logit models are in fact special classes of ANNs). As such the parameters of those models could be calibrated by means of a Bayesian regulated optimization method.

   (a) Discrete choice modelling is an scientific area which is assumption-rich but data-poor. The Bayesian method allows for assumptions only if warranted by the data without resorting to particular statistical tests (metrics), which do not necessarily reflect whether or not assumptions are really supported by the data (compare (chapter 5) the correlations between SSNN hidden states and output with the actual relevance of these hidden states), see (MacKay 1995).

   (b) More complex (and assumed more realistic) choice models have nested structures which make the relationship between data and parameters highly non-linear. This implies that (as in neural networks) the parameter sets found are by no means unique solutions. A probabilistic measure reflecting their uncertainty would render conclusions drawn from these models much more realistic and valuable. If no closed form expressions can be constructed for the posterior distribution of these parameters, than at least this distribution should be approximated with for example bootstrapping or subsampling.

Bibliography

Abdel, A., Kitamura, R. & Jovanis, P. (1997), 'Using stated preference data for studying the effect of advanced traffic information on drivers' route choice', *Transportation Research Part C: Emerging Technologies* **5C**, 39–50.

Abdulhai, B., Porwal, H. & Recker, W. (1999), Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks, *in* 'Proceedings of the 78th Annual Meeting of the Transportation Research Board', National Academies Press, Washington D.C., USA.

Armitage, W. & Lo, J.-C. (1994), Enhancing the robustness of a feedforward neural network in the presence of missing data, *in* 'Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on', Vol. 2, pp. 836–839 vol.2.

Arnott, R., de Palma, A. & Lindsey, R. (1991), 'Does providing information to drivers reduce traffic congestion?', *Transportation Research A* **25A**(5), 309–318.

AVV (2002), Verkeersgegevens jaarrapport 2001 (traffic statistics yearly report), Technical report, Ministerie van Verkeer en Waterstaat, Directoraat-Generaal Rijkswaterstaat, Adviesdienst Verkeer en Vervoer (AVV). AVV Transport Research Centre, Ministry of Transport, Public Works and Watermanagement (Dutch).

Ben-Akiva, M. (1998), Dynamit, a simulation-based system for traffic prediction and guidance generation, *in* 'Proceedings of the TRISTAN III conference', San Juan, Puerto Rico.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. N. & Mishalani, R. (2002), Network state estimation and prediction for real-time transportation management applications, *in* 'Transportation Research Board Annual Meeting, CD-Rom', National Academies Press, Washington D.C. USA.

Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. & Yang, Q. (2003), 'Evaluation of freeway control using a microscopic simulation laboratory', *Transportation Research Part C:-Emerging Technologies* **11**(1), 29–50.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press, United Kingdom.

Boker, G. (2000), 'Traffic state analysis on the basis of floating car data', *Automatisierungstechnik* **48**(8), 365–371.

Boker, G. & Lunze, J. (2001), 'State estimation in freeway traffic with floating car data', *Automatisierungstechnik* **49**(11), 497–504.

Bonsall, P. (2000), Travellers' response to uncertainty, *in* 'Reliability of Transport Networks', Traffic Engineering Series, Research Studies Press Ltd, pp. 1–10.

Bovy, P. H. L. & Thijs, R. (2000), *Estimators of Travel Time for Road Networks, New Developments, Evaluation Results and Applications*, Delft University Press, Delft, the Netherlands.

Box, G. E. P. & Jenkins, G. (1976), *Time Series Analysis: Forecasting and Control*, Holden-Day.

Boyce, D. B., Hicks, J. & Sen, A. (1991), In-vehicle navigation requirements for monitoring link travel times in a dynamic route guidance system, Technical report, Urban Transportation Center, University of Illinois at Chicago.

Boyce, D., Rouphail, N. & Kirson, A. (1993), Estimation and measurement of link travel times in the advance project, *in* 'Proceedings of the 4th IEEE-IEE Vehicle Navigation and Information Systems Conference', Institute of Electrical and Electronics Engineers, New York, Ottawa, Ont., Canada, pp. 62–66.

Bozic, S. (1979), *Digital and Kalman Filtering*, Edward Arnold pub., London.

Brackstone, M., Fisher, G. & McDonald, M. (2001), The use of probe vehicles on motorways, some empirical observations, *in* 'Proceedings of the World Congress on Intelligent Transport Systems', Sydney, Australia.

Bretherton, D., Bowen, G. & Wood, K. (2003), Effective urban traffic management and control : Recent developments in scoot, *in* 'Transportation Research Board Annual Meeting, CD-Rom', National Academies Press, Washington D.C, USA, pp. 1–15.

Brockwell, P. & Davis, R. (1996), *Introduction to Time Series and Forecasting*, Springer-Verlag, New York.

Buisson, C., Lebacque, J. P. & Lesort, J. B. (1998), Travel times computation for dynamic assignment modelling, *in* G. H. Bell, ed., 'Transportation Networks: Recent Methedological Advances. Selected Proceedings of the 4th Euro Transportation Meeting', Pergamon Press, pp. 303–317.

Chen, H., Grant-Muller, S., Mussone, L. & Montgomery, F. (2001), 'A study of hybrid neural network approaches and the effects of missing data on traffic forecasting', *Neural Computing and Applications* **10**, 277–286.

Chen, H., Mussone, L., Montgomery, F. & Grant-Muller, S. (1998), Effects of missing data on neural network performance in forecasting flow, *in* 'Proceedings of the 1998 Conference on Traffic and Transportation Studies ICTTS', Beijing, China, pp. 320–329.

Chen, M. & Chien, S. I. J. (2001), 'Dynamic freeway travel-time prediction with probe vehicle data - link based versus path based', *Transportation Research Record* **1768**, 157–161.

Chen, P. S. T., Srinivasan, K. K. & Mahmassani, H. S. (1999), Effect of information quality on compliance behavior of commuters under real-time traffic information, *in* 'Proceedings of the 77th Transportation Research Board Annual Meeting', National Academies Press, Washington D.C. USA.

Chen Shu, C., Shyu Mei, L., Zhang, C. & Strickrott, J. (2002), 'A multimedia data mining framework: Mining information from traffic video sequences', *Journal of Intelligent Information Systems* **19**(1), 61–77.

Cheng, H. H., Shaw, B. D., Palen, J., Larson, J. E., Hu, X. & van Katwyk, K. A. (2001), 'Real-time laser-based detection system for measurement of delineations of moving vehicles', *IEEE/ASME Transactions on Mechatronics* **6**(2), 170–187.

Chernick, M. R. (1999), *Bootstrap Methods: A Practitioner's Guide*, Wiley, New York.

Cheu, R.-L. (1998), Freeway traffic prediction using neural networks, *in* 'Proceedings of the International Conference on Applications of Advanced Technologies in Transportation Engineering', ASCE, Reston, VA, USA, pp. 247–254.

Cheu, R. L., Xie, C. & Lee, D. H. (2002), 'Probe vehicle population and sample size for arterial speed estimation', *Computer-Aided Civil and Infrastructure Engineering* **17**(1), 53–60.

Chien, S. I. J. & Kuchipudi, C. M. (2003), 'Dynamic travel time prediction with real-time and historic data', *Journal of Transportation Engineering-Asce* **129**(6), 608–616.

Chun-Hsin, W., Chia-Chen, W., Da-Chun, S., Ming-Hua, C. & Jan-Ming, H. (2003), Travel time prediction with support vector regression, *in* 'Proceedings of the 2003 IEEE Conference on Intelligent Transportation Systems', IEEE, Shanghai, China.

Daamen, W. & Hoogendoorn, S. P. (2003), 'Experimental research of pedestrian walking behavior', *Transportation Research Record* **1828**, 20–30.

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

D'Angelo, M. P., Al-Deek, H. M. & Wang, M. C. (1999), 'Travel-time prediction for freeway corridors', *Transportation Research Record* **1676**, 184–191.

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

Dia, H. (2001), 'An object-oriented neural network approach to short-term traffic forecasting', *European Journal of Operational Research* **131**(2), 253–261.

Diakaki, C., Papageorgiou, M. & McLean, T. (2000), 'Integrated traffic-responsive urban corridor control strategy in glasgow, scotland: Application and evaluation', *Transportation Research Record* **1727**, 101–111.

Dorffner, G. (1996), 'Neural networks for time series processing', *Neural Network World* **6**, 447–468.

Dougherty, M. (1995), 'A review of neural networks applied to transport', *Transportation Research C* **3**(4), 247–260.

Efron, B. (1979), 'Bootstrap methods: another look at the jackknife', *Annals of Statistics* **7**, 1–26.

Elefteriadou, L. & Lertworawanich, P. (2003), Defining, measuring and estimating freeway capacity, *in* 'Transportation Research Board Annual Meeting CD-ROM', National Academies Press, Washington D.C., USA.

Elman, J. (1990), 'Finding structure in time', *Cognitive Science* **14**, 179–211.

Ervin, R., Bogard, S. & Fancher, P. S. (2001), 'Radar detection of vehicles in a string: Gaining situation awareness of a propagating conflict', *Transportation Research Record* **1779**, 33–39.

Fallah-Tafti, M. (2001), 'The application of artificial neural networks to anticipate the average journey time of traffic in the vicinity of merges', *Knowledge-Based Systems* **14**, 203–211.

Faris, P. D., Ghali, W. A., Brant, R., Norris, C. M., Galbraith, P. D. & Knudtson, M. L. (2002), 'Multiple imputation versus data enhancement for dealing with missing data in observational health care outcome analyses', *Journal of Clinical Epidemiology* **55**(2), 184–191.

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

Gabrys, B. (2002), 'Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems', *International Journal of Approximate Reasoning* **30**(3), 149–179.

Geuze, M. J., Van der Berg, W. D., Noort, M. & Heskes, T. M. (1998), De invloed van het weer op de verkeersafwikkeling, eindrapport vooronderzoek (the influence of weather on traffic flow, final report), Technical report, Meteo Consult / Stichting Neurale Netwerken (Foundation of Neural Networks).

Hagan, M. T. & Menhaj, M. B. (1994), 'Training feed-forward networks with the marquardt algorithm', *IEEE transactions on Neural Networks* **5**(6), 989–993.

Haj-Salem, H. & Lebacque, J. P. (2002), 'Reconstruction of false and missing data with first-order traffic flow model', *Transportation Research Record* **1802**, 155–165.

Harvey, B. A., Champion, G. H. & Deaver, R. (1993), Accuracy of traffic monitoring equipment field tests, *in* 'Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference', Ottawa, Ontario, Canada, pp. 141–144.

Hecht-Nielsen, R. (1990), *Neurocomputing*, Addison-Wesley Publishing Company, Readen, Massachusetts, USA.

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

Helbing, D. & Treiber, M. (2002), 'Critical discussion of "synchronized flow"', *Cooperative Transportation Dynamics* **1**, 2.1–2.24.

Heskes, T. (1997), Practical confidence and prediction intervals, *in* M. Mozer, M. Jordan & T. Petsche, eds, 'Advances in Neural Information Processing Systems', Vol. 9, MIT Press, Cambridge, Massachusets, USA, pp. 176–182.

Hoogendoorn-Lanser, S. & Hoogendoorn, S. P. (2000), A fuzzy genetic approach to travel choice behavior in public transport networks, *in* 'CD-ROM Preprints Transportation Research Board 79th Annual Meeting', The National Academies, Washington D.C.

Hoogendoorn, S. P. (1997), Optimal control of dynamic route information panels, *in* M. Papageorgiou & A. Pouliezos, eds, 'Preprints of the 1997 IFAC/IFIP/IFORS Symposium', Technical University of Crete, Chania, Greece, pp. 427–432.

Hoogendoorn, S. P. (1999), *Multiclass Continuum Modelling of Multilane Traffic Flow*, Delft University Press, Delft, The Netherlands.

Hoogendoorn, S. P. (2000), Multiclass traffic filtering with applications to travel time estimation, *in* P. H. L. Bovy & R. Thijs, eds, 'Estimators of Travel Time for Road Networks', Delft University Press, Delft, pp. 75–106.

Hoogendoorn, S. P., Botma, H. & Minderhoud, M. M. (2003), *Traffic Flow Theory and Simulation*, Delft University of Technology, Faculty of Civil Engineering and Geosciences, Transportation and Planning Section, Delft. Lecture notes for 5th grade masters course CT4821.

Hoogendoorn, S. P. & Bovy, P. H. L. (2001), 'State-of-the-art of vehicular traffic flow modeling', *Proc. Institution of Mechanical Engineers* **215**(1), 283–303.

Hu, T. Y. (2001), 'Evaluation framework for dynamic vehicle routing strategies under real-time information', *Transportation Research Record* **1774**, 115–122.

Huber, W., Lädke, M. & Ogger, R. (1997), Extended floating car data for the acquisition of traffic information, *in* 'Mobility for Everyone: Proceedings of the 4th World Congress on Intelligent Transport Systems', Berlin, Germany.

Huisken, G. & Van Berkum, E. C. (2003), A comparative analysis of short-range travel time prediction methods, *in* 'TRB 2003 Annual Meeting CD-ROM', National Academies Press, Washington D.C., USA.

Hussain Tarik, M. (1995), 'Infrared pyroelectric sensor for detection of vehicular traffic using digital signal processing techniques', *IEEE transactions on vehicular technology* **44**(3), 683–689. Using Smart Source Parsing eng.

Innamaa, S. (2001), Short term prediction of highway travel time using mlp neural networks, *in* 'Proceedings of the 8th World Congress on Intelligent Transport Systems', Sidney, Australia.

Ishak, S. & Al-Deek, H. (2002), 'Performance evaluation of short-term time-series traffic prediction model', *Journal of Transportation Engineering* **128**(6), 490–498.

Ishak, S. & Alecsandru, C. (2003), Optimizing traffic prediction performance of neural networks under various topological, input, and traffic condition settings, *in* 'Transportation Research Board Annual Meeting CD-ROM', National Academies Press, Washington, D.C., USA.

Ishak, S., Kotha, P. & Alecsandru, C. (2003), Optimization of dynamic neural networks performance for short-term traffic prediction, *in* 'Transportation Research Board Annual Meeting CD-ROM', National Academies Press, Washington D.C., USA.

Jarviluoma, M. & Heikkila, T. (1995), 'Ultrasonic multi-sensor system for object locating applications', *Mechatronics* **5**(4), 433–440.

Jeffrey, S. J., Carter, J. O., Moodie, K. B. & Beswick, A. R. (2001), 'Using spatial interpolation to construct a comprehensive archive of australian climate data', *Environmental Modeling and Software* **16**(4), 309–330.

Jochem, A., De Hoog, A. & Zijderhand, F. (1998), Floating car data in the netherlands., *in* 'Towards the New Horizon Together: Proceedings of the 5th World Congress on Intelligent Transport Systems', Seoul, Korea. Paper no. 2109.

Kalman, R. (1960), 'A new approach to linear filtering and prediction problems', *ASME Basic Engineering Journal* .

Kappen, B. & Gielen, S. (1997), *Neural Networks: Best Practice in Europe*, Proceedings of the Stichting Neurale Netwerken Conference, World Scientific Publishing Co. Pte. Ltd., Singapore.

Kay, J. W. & Titterington, D. M. (1999), *Statistics and Neural Networks. Advances at the Interface*, Oxford University Press, London, UK.

Kaysi, I., Ben-Akiva, M. & Koutsopoulos, H. (1993), 'An integrated approach to vehicle routing and congestion prediction for real-time driver guidance', *Transportation Research Record* **1408**, 66–74.

Kerner, B. S. (1999), *The Physics of Traffic*, Physics World.

Khattak, A. J., Schofer, J. L. & Koppelman, F. S. (1995), 'Effect of traffic information on commuters propensity to change route and departure time', *Journal of Advanced Transportation* **29**(2), 193–212.

Khattak, A. J., Yim, Y. & Stalker, L. (1995), 'Does travel information influence commuter and noncommuter behavior?', *Transportation Research Record* **1694**, 48–58.

Kraan, M., Van der Zijpp, N. J., Tutert, B., Vonk, T. & Van Megen, D. (1999), 'Evaluating network effects of vms's in the netherlands', *Transportation Research Record* **1689**, 69–67.

Kremer, S. C. (2001), 'Spatiotemporal connectionist networks: A taxonomy and review', *Neural Computation* **13**, 249–306.

Kuhne, R. D. (1997), Data fusion for dynamic route guidance systems, *in* M. Papageorgiou & A. Pouliezos, eds, 'Proceedings of the 8th IFAC/IFIP/IFORS Symposium', Vol. 3, Pergamon, Oxford, UK, pp. 1319–1323.

Kwon, J., Coifman, B. & Bickel, P. (2000), 'Day-to-day travel-time trends and travel-time prediction from loop-detector data', *Transportation Research Record* **1717**, 120–129.

Lan, C. J. & Miaou, S. P. (1999), 'Real-time prediction of traffic flows using dynamic generalized linear models', *Transportation Research Record* **1678**, 168–178.

Lebacque, J. P. (1996), The gudunov scheme and what it means for first order traffic flow models, *in* J. B. Lesort, ed., 'Proceedings of the 13th International Symposium on Transportation and Traffic Theory', Lyon, France, pp. 647–677.

Leung Chi, S. & Chan Lai, W. (2003), 'Dual extended kalman filtering in recurrent neural networks', *Neural Networks* **16**(2), 223–239.

Leutzbach, W. (1987), *Introduction to the theory of traffic flow*, Springer-Verlag, Berlin Heidelberg.

Lighthill, M. & Whitham, G. (1955), 'On kinematic waves ii: A theory of traffic flow on long crowded roads', *Proc. R. Soc* **A 229**(1178), 317–345.

Lindveld, C. D. R. & Thijs, R. (1999), On-line travel time estimation using inductive loop data: The effect of instrumentation peculiarities on travel time estimation quality, *in* 'proceedings of the 6th ITS World Congres', Toronto Canada.

Lindveld, C. D. R., Thijs, R., Bovy, P. H. L. & Van der Zijpp, N. J. (2000), 'Evaluation of online travel time estimators and predictors', *Transportation Research Record* **1719**, 45–53.

Lingras, P., Sharma, S. & Zhong, M. (2002), 'Prediction of recreational travel using genetically designed regression and time-delay neural network models', *Transportation Research Record* **1805**, 16–24.

Logghe, S. (2003), *Dynamic Modelling of Heterogeneous Vehicular Traffic*, Faculty of Applied Science, Katholieke Universiteit Leuven, Leuven.

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

MacKay, D. J. C. (1994), 'Bayesian non-linear modelling for the prediction competition', *ASHRAE Transactions* **100**(2), 1053–1062.

MacKay, D. J. C. (1995), 'Probable networks and plausible predictions: A review of practical bayesian methods for supervised neural networks', *Network: Computation in Neural Systems* **6**(3), 469–505.

Mahmassani, H. S. & Liu, Y.-H. (1999), 'Dynamics of commuting decision behavior under advanced traveller information systems', *Transportation Research C* **7**, 91–107.

MathWorld (2004), 'http://mathworld.wolfram.com'.

Meert, K. (1996), A real-time recurrent learning network structure for dealing with missing sensor data, *in* 'Proceedings of the 1996 IEEE Conference on Neural Networks', Vol. 3, Washington D.C, USA, pp. 1600–1605.

Michalopoulos, P. & Hourdakis, J. (2001), 'Review of non-intrusive advanced sensor devices for advanced traffic management systems and recent advances in video detection', *Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering* **215**(14), 345–355.

Middelham, F. (2001), 'Predictability: Some thoughts on modeling', *Future Generation Computer Systems.* **17**(5), 627–636.

Nam, D. H. & Drew, D. R. (1996), 'Traffic dynamics: Method for estimating freeway travel times in real time from flow measurements', *Journal of Transportation Engineering* **122**(3), 186–191.

Nguyen, D. & Widrow, B. (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.

Pancratz, A. (1991), *Forecasting with dynamic regression models*, Wiley-InterScience Publication, NY, USA.

Papadopoulos, G., Edwards, P. J. & Murray, A. F. (2001), 'Confidence estimation methods for neural networks: A practical comparison', *IEEE Transactions on Neural Networks* **12**(6), 1278–1287.

Papageorgiou, M., Blosseville, J. M. & Haj-Salem, H. (1989), 'Macroscopic modeling of traffic flow on the boulevard peripheric in paris', *Transportation Research B* **23**, 29–47.

Park, D. J. & Rilett, L. R. (1998), 'Forecasting multiple period freeway link travel times using modular neural networks', *Transportation Research Record* **1617**, 163–170.

Park, D. J. & Rilett, L. R. (1999), 'Forecasting freeway link travel times with a multilayer feed-forward neural network', *Computer Aided Civil and Infrastructure Engineering* **14**(5), 357–367.

Park, D. J., Rilett, L. R. & Han, G. (1999), 'Spectral basis neural networks for real-time travel time forecasting', *Journal of Transportation Engineering* **125**(6), 515–523.

Petty, K. F., Bickel, P., Ostland, M., Rice, J., Schoenberg, F., Jiang, J. & Ritov, Y. (1998), 'Accurate estimation of travel times from single loop detectors', *Transportation Research A* **32**(1), 1–17.

Philips, W. F. (1979), 'A kinetic model for traffic flow with continuum implications', *Transportation Planning and Technology* **5**, 131–138.

Polak, J. & Oladeinde, F. (2000), An empirical model of travelers' day-to-day learning in the preceense of uncertain travel times, *in* 'Reliability of Transport Networks', Traffic Engineering Series, Research Studies Press Ltd., pp. 11–30.

Politis, D. N. & Romano, J. P. (1994), 'Large sample confidence regions based on subsamples under minimal assumptions', *Annals of Statistics* **22**, 2031–2050.

Politis, D. N., Romano, J. P. & Wolf, M. (2001), 'On the asymptotic theory of subsampling', *Statistica Sinica* **11**, 1105–1124.

Prechelt, L. (1998), 'Automatic early stopping using cross validation: Quantifying the criteria.', *Neural Networks* **11**(4), 761–767.

Rice, J. & Van Zwet, E. (2001), A simple and effective method for predicting travel times on freeways, *in* 'Proceedings of the IEEE Conference on Intelligent Transportation Systems', Oakland, CA, United States, pp. 227–232.

Rilett, L. R. & Park, D. (2001), 'Direct forecasting of freeway corridor travel times using spectral basis neural networks', *Transportation Research Record* **1752**, 140–147.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, U.K.

Sariks, C. (1997), The advance project: Formal evaluation of the targeted deployment, volume 1, Technical report, Argonne National Laboratory.

Schafer, J. L. (1997), *Analysis of Incomplete Multivariate Data*, Chapman and Hall, London, UK.

Schrader, C. C., Kornhauser, A. L. & Friese, L. M. (2004), Using historical information in forecasting travel times, *in* 'Transportation Research Board Annual Meeting, CD-Rom', National Academies Press, Washington D.C.

Setiono, R. & Gaweda, A. (2000), Neural network pruning for function approximation, *in* 'Proceedings of the International Joint Conference on Neural Networks', Vol. 6, IEEE Neural Network Council, Como, Italy, pp. 443–448.

Setiono, R. & Liu, H. (1997), 'Neural-network feature selector', *IEEE Transactions on Neural Networks* **8**(3), 654–62.

Smith, B. L. & Demetsky, M. J. (1997), 'Traffic flow forecasting: Comparison of modeling approaches', *Journal of Transportation Engineering* **123**(4), 261–266.

Smulders, S. A., Messmer, A. & Knibbe, W. J. J. (1999), Real-time application of metanet in traffic management centres, *in* 'Proceedings of the 6th World Congress on Intelligent Transport Systems (ITS)', Toronto, Canada.

Srinivasan, K. & Jovanis, P. P. (1996), 'Determination of number of probe vehicles required for reliable travel time measurement in urban network', *Transportation Research Record* **1537**.

Sun, H., Liu, H. X., Xiao, H., He, R. R. & Ran, B. (2003), Short-term traffic forecasting using the local linear regression model, *in* 'Transportation Research Board Annual Meeting CD-ROM', National Academies Press, Washington D.C., USA.

Tampere, C. M. J., Berghout, E. A. & Westerman, M. (1999), Travel time algorithm for the urban road network. part 1: the general applicable computational kernel. (reistijd algoritme voor het stedelijk wegennet. deel 1: algemeen toepasbare rekenkern.), Technical report, TNO INRO, The Netherlands.

Thijs, R., Lindveld, C. D. R., Van der Zijpp, N. J. & Bovy, P. H. L. (1998*a*), Deliverable d10.3: Final report of assesment results, volume i, Technical Report D10.3, Consortium for the EU Telematics Applications Programme TRANSPORT, project TR1017 DACCORD.

Thijs, R., Lindveld, C. D. R., Van der Zijpp, N. J. & Bovy, P. H. L. (1998*b*), Deliverable d10.3: Final report of assesment results, volume ii, Technical report, Consortium for the EU Telematics Applications Programme TRANSPORT, project TR1017 DACCORD.

Thijs, R., Lindveld, C. D. R., Van der Zijpp, N. J. & Bovy, P. H. L. (1999), *Evaluation of Travel Time Estimation and Prediction Algorithms: Evaluation Results from the DACCORD Project*, Transportation and Planning Section, Faculty of Civil Engineering and Geosciences, Delft University of Technology.

Thodberg, H. H. (1991), 'Improving generalization of neural networks through pruning', *International Journal of Neural Systems* **1**(4), 317–26.

Thodberg, H. H. (1996), 'A review of bayesian neural networks with an application to near infrared spectroscopy', *IEEE Transactions on Neural Networks* **7**(1), 56–72.

Transportation Research Board, N. R. C. (2000), *Highway Capacity Manual 2000*, National Academies Press, Washington D.C., USA.

Turksma, S. (2001), An overview of utopia-spot deployment in northern europe, *in* '8th World Congress on Intelligent Transport Systems, CD-Rom', Sydney, Australia, pp. 1–8.

Van Berkum, P. & Van der Mede, P. (1993), *The Impact of Traffic Information: Dynamics in Route and Departure Time Choice*, Delft University Press, Delft, the Netherlands. PhD thesis of the Transport and Planning Department, Faculty of Civil Engineering and Geosciences, Delft University of Technology.

Van der Vlist, J. M. (1995), The online estimator of capacity (de on-line schatter van de actuele capaciteit), Technical report, Ministerie van Verkeer en Waterstaat, Adviesdienst Verkeer en Vervoer, Hoofdafdeling Infrastructuur en Benutting.

Van der Zijpp, N. J. & Hoogendoorn, S. P. (1999), Network level evaluation of dtm tools within daccord: An analysis of the minimum number of probe vehicles required, *in* 'World Transport Research: Planning, Operation, Management and Control', Vol. 2, pp. A469–A482.

Van der Zijpp, N. J. & Lindveld, C. D. R. (1999), Evaluation of queue length display at the amsterdam orbital motorway, *in* 'Proceedings of the 6th ITS World Congress', Toronto, Canada.

Van Grol, H. J. M., Danech-Pajouh, M., Manfredi, S. & Whittaker, J. (1999), Daccord: On-line travel time prediction, *in* 'World Transport Research: Planning, Operation, Management and Control', Vol. 2, pp. A455–A467.

Van Grol, R., Middelham, F. & Van Ruremonde, A. (1997), Daccord/boss: Its developments in the amsterdam region, *in* 'Mobility for Everyone, Proceedings of the

4th World Congress on Intelligent Transport Systems', ITS Congress Association, Berlin. paper no. 2164.

Van Lint, J. W. C. (2003), Confidence intervals for real time freeway travel time prediction, *in* 'Proceedings of the 2003 IEEE Conference on Intelligent Transportation Systems', IEEE, Sjanghai, China.

Van Lint, J. W. C. (2004), Quantifying uncertainty in real-time neural network based freeway travel prediction, *in* '83rd Transportation Research Board Annual Meeting', National Academies Press, Washington D.C., USA.

Van Lint, J. W. C., Hoogendoorn, S. P. & Van Zuylen, H. J. (2000), Robust and adaptive travel time prediction with neural networks, *in* 'Proceedings of the 6th Annual TRAIL Congress, Part 2', Delft University Press, Delft, The Netherlands.

Van Lint, J. W. C., Hoogendoorn, S. P. & Van Zuylen, H. J. (2002*a*), 'Freeway travel time prediction with state-space neural networks - modeling state-space dynamics with recurrent neural networks', *Transportation Research Record* **1811**, 30–39.

Van Lint, J. W. C., Hoogendoorn, S. P. & Van Zuylen, H. J. (2002*b*), State space neural networks for freeway travel time prediction, *in* 'International Conference on Artificial Neural Networks - ICANN', Vol. 2415 of *Lecture Notes in Computer Science*, pp. 1043–1048.

Van Lint, J. W. C., Hoogendoorn, S. P. & Van Zuylen, H. J. (2003), Toward a robust framework for freeway travel time prediction: Experiments with simple imputation and state-space neural networks, *in* 'Transportation Research Board Annual Meeting, CD Rom', National Academies Press, Washington D.C, USA.

Van Lint, J. W. C., Tu, H. & Van Zuylen, H. J. (2004), Travel time reliability on freeways, *in* 'Submitted for presentation and publication at the 10th World Conference on Transport Research (WCTR)', Istanbul, Turkey.

Van Lint, J. W. C. & Van der Zijpp, N. J. (2003*a*), An improved travel-time estimation algorithm using dual loop detectors, *in* 'Transportation Research Board Annual Meeting, CD Rom', National Academies Press, Washington D.C, USA.

Van Lint, J. W. C. & Van der Zijpp, N. J. (2003*b*), 'Improving a travel time estimation algorithm by using dual loop detectors', *Transportation research Record* **1855**, 41–48.

Van Toorenburg, J. A. C. (1998), Astrival functionele specificatie algoritme, rijtijd en filelengteschatter voor meetvak (in dutch), Technical report, AVV Transport Research Centre, Ministry of Transport, Public Works and Water Management.

Van Zuylen, H. J. & Muller, T. H. J. (2002), Regiolab delft, *in* 'Proceedings of the 9th World Congress on Intelligent Transport Systems, CD-Rom', Chicago, Illinois, USA. http://www.regiolab-delft.nl.

Van Zuylen, H. J. & Viti, F. (2003), Uncertainty and the dynamics of queues at signalized intersections, *in* 'Proceedings CTS-IFAC conference', Elsevier, Amsterdam, Tokyo.

Vermijs, R. G. M. M. & Schuurman, H. (1994), Evaluating capacity of freeway weaving sections and on-ramps using the microscopic simulation model fosim, *in* 'Proceedings of the second international symposium on highway capacity', Vol. 2, Sydney, Australia, pp. 651–670.

Viti, F. & Van Zuylen, H. J. (2004), Modeling queues at signalized intersections, *in* 'Annual Meeting of the Transportation Research Board, CD-Rom', Washington D.C., USA.

Wang, Y., Papageorgiou, M. & Messmer, A. (2003), Motorway traffic state estimation based on extended kalman filter., *in* 'CD-ROM European Control Conference ECC Š03', Cambridge, UK.

Werbos, P. J. (1990), 'Backpropagation through time: what it does and how to do it.', *Proc. IEEE* **78**(101), 1550–1560.

Westerman, M. (1995), *Real-Time Traffic Data Collection for Transportation Telematics*, Delft University Press, Delft, The Netherlands.

Williams, B. M. (2001), Multivariate traffic flow prediction: An evaluation of arimax modelling, *in* 'Transportation Research Board 80th Annual Meeting, CD-Rom', National Academies Press, Washington D.C. USA.

Yang, F., Sun, H., Tao, Y. & Ran, B. (2004), Temporal difference learning with recurrent neural network in multi-step ahead freeway speed prediction, *in* 'Transportation Research Board Annual Meeting, CD-Rom', National Academies Press, Washington D.C.

Zhang, X. Y. & Rice, J. A. (2003), 'Short-term travel time prediction', *Transportation Research Part C-Emerging Technologies* **11**(3-4), 187–210.

Zwahlen, H. T. & Russ, A. (2002), 'Evaluation of the accuracy of a real-time travel time prediction system in a freeway construction work zone', *Transportation Research Record* **1803**, 87–93.

Zwaneveld, P., Wilmink, I., Immers, B., Malipaard, E. & Heyse, D. (1998), An overview of incident management projects in the netherlands., *in* 'Traffic Management and Road Safety. Proceedings of Seminars J. and K. At the AET European Transport Conference', PTRC Education and Research Services Ltd., Glenthorne House, Hammersmith Grove, London W6 0LG, United Kingdom.

Appendix A

Performance Indicators

Let N be the total number of observations, y_n be the n^{th} predicted value for the n^{th} input, output pattern $\{u_n, t_n\}$. Further more let

$$\overline{y} = \frac{1}{N}\sum y_n$$

be the mean prediction over all N patterns, and

$$\overline{t} = \frac{1}{N}\sum t_n$$

be the mean observed value over all N patterns, and

$$e_n = y_n - t_n$$

denote the prediction error for data pattern n. Table A.1 lists the performance indicators[1] used throughout this dissertation thesis

Finally, let σ_W^2 denote the variance associated with predicting $\{y_n\}_{n=1}^N$, due to e.g. variance in the model parameters and let σ_D^2 be the variance associated with the target

[1]Note that $RMSE^2 = Bias^2 + RRE^2$:

$$\begin{aligned} RRE^2 &= \frac{1}{N}\sum \left((y_n - \overline{y}) - \left(t_n - \overline{t}\right)\right)^2 \\ &= \frac{1}{N}\sum \left((\overline{t} - \overline{y}) - (t_n - y_n)\right)^2 \\ &= \frac{1}{N}\sum \left((\overline{t} - \overline{y})^2 - 2(t_n - y_n)(\overline{t} - \overline{y}) + (t_n - y_n)^2\right) \\ &= -Bias^2 + RMSE^2 \end{aligned}$$

Table A.1: Performance indicators

Abbr.	Meaning	Formula
ME	Mean Error	$\frac{1}{N}\sum e_n$
SE	St.dev. of Error	$\sqrt{\frac{1}{N-1}\sum (e_n - ME)^2}$
MRE	Mean Relative Error	$100\frac{1}{N}\sum \frac{e_n}{t_n}$
SRE	St.dev. of Relative Error	$100\sqrt{\frac{1}{N-1}\sum \left(\frac{e_n}{t_n} - \frac{MRE}{100}\right)^2}$
MARE	Mean Absolute Relative Error	$100\frac{1}{N}\sum \left\lvert\frac{e_n}{t_n}\right\rvert$
SSE	Sum Squared Error	$\sum (e_n)^2$
MSE	Mean Squared Error	$\frac{1}{N}\sum (e_n)^2$
RMSE	Root Mean Squared Error	$\sqrt{MSE}$
Bias	(difference between means)	$\overline{y} - \overline{t} = ME$
RRE	Root Residual Error	$\sqrt{\frac{1}{N}\sum \left((y_n - \overline{y}) - (t_n - \overline{t})\right)^2}$
RMSEP	Root Mean Squared Error Proportional	$100\frac{RMSE}{\overline{t}}$

data $\{t_n\}_{n=1}^{N}$. The Confidence Interval Coverage Percentage (CI_CP) then denotes the number of observations t_n that fall in the interval

$$y_n \pm c \times \sigma_W$$

in which c denotes the appropriate value from for example a Student-t distribution or a number reflecting nominal coverage of the assumed predictive distribution. The CI_CP index quantifies confidence in our model's prediction. Similarly, the Prediction Interval Coverage Percentage (PI_CP) denotes the number denotes the number of observations t_n that fall in the interval

$$y_n \pm c \times \sigma_{tot}$$

where

$$\sigma_{tot}^2 = \sigma_W^2 + \sigma_D^2 + ...$$

denotes the total uncertainty associated with the prediction. Assuming that various sources of uncertainty are independent, total variance is the sum of variance due to model specification, variance inherent to the target data, and perhaps other (observed) sources of uncertainty. The PI_CP index thus quantifies upper and lower bounds to our prediction. Note that necessarily prediction intervals contain the confidence interval. On a the same data set hence always $PI\_CP \leq CI\_CP$.

Appendix B

Mathematical Description of State Space Neural Network (SSNN)

This appendix assumes some background in the field of artificial neural networks. For comprehensive introductions in the field of artificial neural networks we refer the reader to for example (Bishop 1995), (Ripley 1996), or (Hecht-Nielsen 1990).

B.1 SSNN Topology

The state space neural network (SSNN) model (fig. B.1) is a First Order Context Memory (Kremer 2001) consisting of three layers: an *input layer* which distributes section specific input vectors to the *hidden layer*. The latter also receives signals from the *context layer*, which stores the hidden layer states (that is, the hidden layers output) of the previous time instant. The *output layer* finally processes the hidden layer outputs and produces a scalar output, which is the mean travel time on the route of interest.

The size of the hidden layer is determined by the number of sections M that constitute the route R of interest. However, nothing prevents the SSNN model designer from adding additional neurons to this layer, for example to model effects on travel time that are exogenous, such as weather conditions or traffic measurements from up- or downstream locations relative to R. By definition, the size of the context layer is equal to the size of the hidden layer. The output layer's size is 1, that is, it consists of only one neuron receiving signals from the hidden layer only.

The input layer merely distributes input signals to the hidden layer. Each hidden neuron receives only those inputs associated to that neuron. For hidden neurons that represent sections m on the route, the respective time dependent input vector $\mathbf{u}_m(t)$ contains mean speed and flow measurements from up- and downstream detectors enclosing the section, and, if available, in- and outflows of detectors on on- and off ramps connected

to the section, for example

$$\mathbf{u}_m(t) = \left(q^{m,up}(t), \overline{v}^{m,up}(t), q^{m,down}(t), \overline{v}^{m,down}(t), q^{m,on}(t), q^{m,off}(t)\right)^T$$

In neural network terminology, the input layer is *partially* connected to the hidden layer, while hidden, context and output layer are *fully* connected. The input weights connecting the input signals to the hidden layer are adjustable during SSNN training. Similarly, the weights connecting the context neurons output signals to the hidden layer and the weights connecting the hidden layers' signals to the output layer are also adjustable. The weights connecting the hidden layers signals to the context layer are fixed at 1.0. The context layer merely functions as a storage layer for past hidden neurons signals.

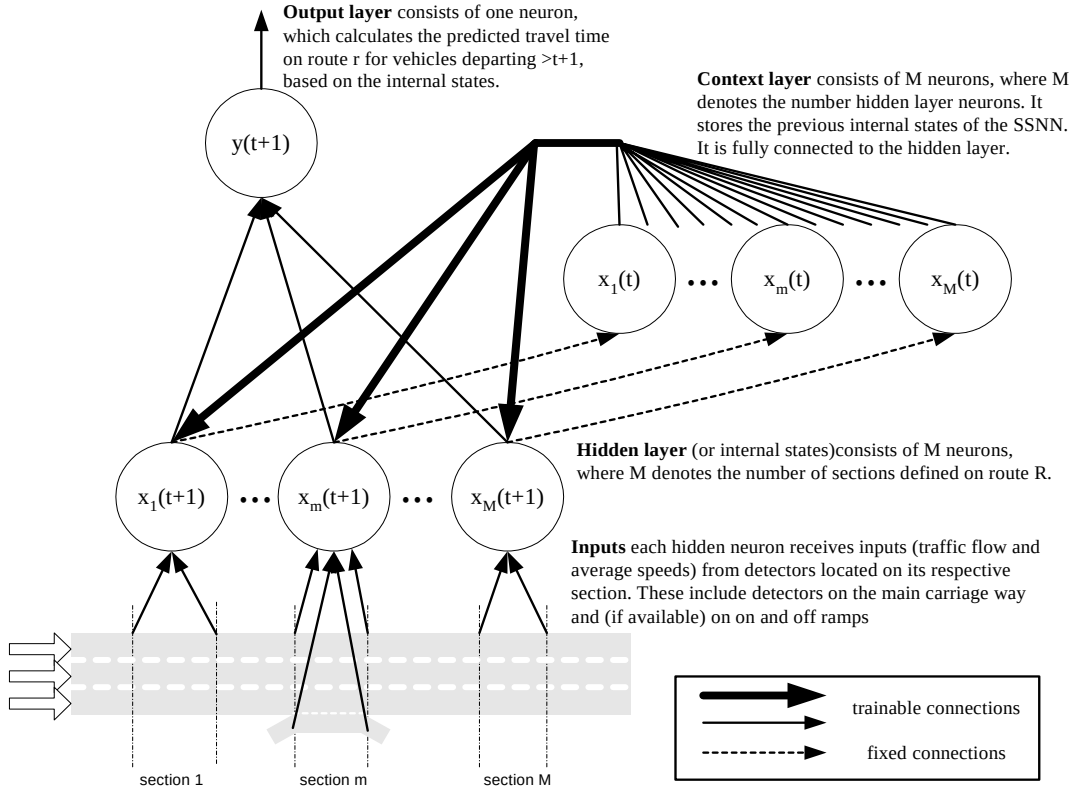


Figure B.1: State-Space Neural Network (SSNN) topology for short term freeway travel time prediction.

B.2 Mathematical Description

We shall refer to the hidden layers' output as the *internal states* of the SSNN. These internal states $x_j(t)$ are calculated as a weighted sum of the inputs and a bias (eqn B.3),

nonlinearly transformed by the well-known sigmoid transfer-function

$$\phi(z) = \frac{1}{1+e^{-z}} \tag{B.1}$$

$$\Theta(\mathbf{z}) = \begin{pmatrix} \phi(z_1) \\ \phi(z_2) \\ \ldots \\ \phi(z_N) \end{pmatrix} \tag{B.2}$$

and can be formulated as

$$I_{hj}(t) = \upsilon_{j0} b_j + \sum_{m=1}^{M} \upsilon_{jm} \mathbf{u}_m(t) + \sum_{m=1}^{M} \nu_{jm} x_m(t-1) \tag{B.3}$$

$$x_j(t) = \phi(I_{hj}(t)) \tag{B.4}$$

where b_j denotes a bias with fixed value 1.0 for hidden neuron j; υ_{j0} denotes the weight associated with that bias; υ_{jm} denotes the weight-vector for hidden neuron j associated with input-vector $\mathbf{u}_m$; and ν_{jm} denotes the weight for hidden neuron j associated with context neuron m. Note that the second term in eqn (B.3) is a vector inproduct, which produces a scalar result.

Similarly, the output neuron transforms its input (the internal states) as follows

$$I_o(t) = \omega_0 c + \sum_{m=1}^{M} \omega_m x_m(t) \tag{B.5}$$

$$y(t) = \phi(I_o(t)) \tag{B.6}$$

where c denotes a bias with fixed value 1.0 for the output neuron j; ω_0 denotes the weight associated with that bias, and ω_m denotes the output weight associate with internal state $x_m(t)$.

Let $\mathbf{u}(t) = (\mathbf{u}_1(t), \mathbf{u}_2(t), ..., \mathbf{u}_M(t))^T$; $\mathbf{x}(t) = (x_1(t), x_2(t), ..., x_M(t))^T$ denote the concatenated input vector of all section specific input vectors, and the SSNN state vector at time instant t respectively. Note the transpose sign depicting both these vectors are treated as column vectors. Finally, let

$$\upsilon_0 = \begin{pmatrix} \upsilon_{10} \\ .. \\ \upsilon_{M0} \end{pmatrix}, \upsilon = \begin{pmatrix} \upsilon_{11} & .. & .. & .. & \upsilon_{1N} \\ .. & .. & .. & .. & .. \\ \upsilon_{M1} & .. & .. & .. & \upsilon_{MN} \end{pmatrix}, \nu = \begin{pmatrix} \nu_{11} & .. & \nu_{1M} \\ .. & .. & .. \\ \nu_{M1} & .. & \nu_{MM} \end{pmatrix} \tag{B.7}$$

denote the bias vector and weight matrices associated with the hidden layer, where N is the total length of the entire input vector $\mathbf{u}(t)$, and

$$\omega = (\omega_1, ..., \omega_M) \tag{B.8}$$

the vector of weights associated with the output layer. The SSNN can now be expressed in a state space form, by using matrix notation:

$$\mathbf{x}(t) = \Theta(\upsilon_0 + \upsilon \mathbf{u}(t) + \nu \mathbf{x}(t-1)) \tag{B.9}$$
$$y(t) = \phi\left(\omega_0 + \omega \mathbf{x}(t)\right) \tag{B.10}$$

By letting $\mathbf{V} = (\upsilon_0, \upsilon, \nu)$, $\mathbf{u}^*(t) = (1, \mathbf{u}(t), \mathbf{x}(t-1))^T$, $\mathbf{x}^*(t) = (1, \mathbf{x}(t))^T$, and $\mathbf{w} = (\omega_0, \omega)$, we can also write

$$\mathbf{x}(t) = \Theta\left(\mathbf{V}\mathbf{u}^*(t)\right) \tag{B.11}$$
$$y(t) = \phi\left(\mathbf{w}\mathbf{x}^*(t)\right) \tag{B.12}$$

B.3 Training the SSNN: Truncated Levenberg-Marquardt

In general terms, the SSNN can be trained (calibrated) in a supervised manner similar to standard feed-forward neural networks. The most widely used technique for supervised training of neural networks is by means of error back-propagation (Hecht-Nielsen 1990). Generally, there are two strands of back-propagation algorithms. The first is often referred to as online or incremental learning. Network performance is evaluated after presenting each input, output data pair $(\mathbf{u}(t), o(t))$ in terms of the neural network output error, and subsequently, weight updates are applied after each performance evaluation.

In the second strand, which is referred to as offline or batch learning, network performance is calculated after a batch of input/output patterns $\{\mathbf{u}(t), o(t)\}_{t=1}^{P}$ have been presented to the network. Only for the latter version a convergence and a universal function approximation theorem exists (Hecht-Nielsen 1990). It is because of this property and the possibility to apply Bayesian regularization techniques to the algorithm (see appendix C) that we use the batch version of backpropagation for training the SSNN. As demonstrated in (Elman 1990), first order context memories like the SSNN can be trained similar general feed-forward structures, that is, without explicitly addressing the functional dependencies of context signals $\mathbf{x}(t-1)$ on $\mathbf{x}(t-2)$ etcetera[1]. This approximation is expressed in the term "truncated" in the title of this section. The next section first describes the backpropagation algorithm, in the sections thereafter the Levenberg-Marquardt version we use for the SSNN, while we conclude with some notes on truncating the algorithm at time period $t-1$ in the last section.

B.3.1 The standard backpropagation algorithm

In the batch version of error back propagation the performance of the neural network is evaluated by means of a SSE function, which depends on the weights (parameters)

[1] There are backpropagation algorithms that do address the time dependency, for example backpropagation through time (BPTT), see e.g. (Werbos 1990)

of the SSNN, and reads[2]:

$$F(\psi) = \frac{1}{2}\sum_{t=1}^{P} F(\psi, t) \tag{B.13}$$

$$F(\psi, t) = (y(t) - o(t))^2 \tag{B.14}$$

where $y(t)$ is calculated with equation B.11, ψ denotes the vector of all weights (parameters) in the SSNN, and P denotes the total number of data pairs in our training data set. For readability we will hereafter use $F_t(\psi) = F(\psi, t)$ for the performance on a single input pattern.

The idea now is to move the weight vector ψ in a direction where F decreases fastest, and stop training as we have reached a (preferably global) minimum of F. Assuming that F is differentiable, this steepest decent direction equals the negative of the gradient of F with respect to ψ

$$-\nabla_{\psi} F(\psi) = -\frac{\partial F}{\partial \psi} = -\left(\frac{\partial F}{\partial \psi_1}, \frac{\partial F}{\partial \psi_2}, ..., \frac{\partial F}{\partial \psi_Q}\right) \tag{B.15}$$

where Q denotes the total number of weights comprising weight vector ψ. Recalling equations (B.11) and (B.12), we let

$$\mathbf{I}_h(t) = \mathbf{V}\mathbf{u}^*(t) \tag{B.16}$$

denote the weighted input vector to the nonlinear transfer function the hidden layer and likewise

$$I_o(t) = \mathbf{w}\mathbf{x}^*(t) \tag{B.17}$$

the weighted input to the output transfer function. Using the chain rule we can write for the output layer neurons' weights

$$\frac{\partial F_t}{\partial \mathbf{w}} = \frac{\partial F_t}{\partial I_o(t)} \frac{\partial I_o(t)}{\partial \mathbf{w}} \tag{B.18}$$

The term $\frac{\partial F_t}{\partial I_o(t)}$ is usually denoted with the delta symbol, and equals the partial derivative of the performance function with respect to the network output times the derivative of the (output) transfer function with respect to its argument:

$$\delta_o(t) = \frac{\partial F_t}{\partial I_o(t)} = \frac{\partial F_t}{\partial y(t)} \frac{\partial y(t)}{\partial I_o(t)} \tag{B.19}$$

$$= \frac{\partial F_t}{\partial y(t)} \phi'(I_o(t)) = \phi'(I_o(t)) \frac{\partial}{\partial y(t)} (y(t) - o(t))^2 \tag{B.20}$$

$$= 2(y(t) - o(t))\phi'(I_o(t)) \tag{B.21}$$

[2]In some publications the output error is defined as $F_t(\psi) = (o(t) - y(t))^2$, which makes no difference for the overall SSE error, but does slightly change the derivation of the learning law below. Specifically, it changes the signs of the delta's and hence of the Jacobian matrix elements in the next sections. Since a single output errors' sign also also swapped, both output error definitions lead to exactly the same results.

Combining (B.18), (B.19), (B.20), and (B.21) then gives the following vector of derivatives of the performance function with respect to the output neurons weights

$$\frac{\partial F_t}{\partial \mathbf{w}} = \delta_o(t)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}\mathbf{x}^*(t)\right) = \delta_o(t)\mathbf{x}^*(t) \tag{B.22}$$

$$= 2(y(t) - o(t))\phi'(I_o(t))\mathbf{x}^*(t) \tag{B.23}$$

In a similar fashion, by using again the (multidimensional) chain rule we can also calculate a vector of delta's and hence gradients for the performance function with respect to the weights of the hidden neurons

$$\Delta_h(t) = \frac{\partial F_t}{\partial \mathbf{I}_h(t)} = \frac{\partial F_t}{\partial \mathbf{x}^*(t)}\frac{\partial \mathbf{x}^*(t)}{\partial \mathbf{I}_h(t)} = \frac{\partial F_t}{\partial \mathbf{x}^*(t)}\Theta'(\mathbf{I}_h(t)) \tag{B.24}$$

Using (B.17) we have

$$\frac{\partial F_t}{\partial \mathbf{x}^*(t)} = \frac{\partial F_t}{\partial I_o(t)}\frac{\partial I_o(t)}{\partial \mathbf{x}^*(t)} = \delta_o(t)\mathbf{w} \tag{B.25}$$

substituting in (B.24) then gives

$$\Delta_h(t) = \delta_o(t)\mathbf{w}\Theta'(\mathbf{I}_h(t)) \tag{B.26}$$

yielding analogously to (B.22) and (B.23) for the derivative of the performance function with respect to the hidden layer weights

$$\frac{\partial F_t}{\partial \mathbf{V}} = \Delta_h(t)\frac{\partial}{\partial \mathbf{V}}\left(\mathbf{V}\mathbf{u}^*(t)\right) = \Delta_h(t)\mathbf{u}^*(t) \tag{B.27}$$

Practically, this means that by starting at the output layer, we can recursively calculate delta's and hence the gradient vector of eqn (B.15). If we let

$$\Delta(t) = (\Delta_h(t), \delta_o(t)) \tag{B.28}$$

denote a vector of all delta's in the SSNN and

$$\mathbf{s}(t) = (\mathbf{u}^*(t), \mathbf{x}^*(t)) \tag{B.29}$$

a vector of all hidden and output neuron inputs, then this gradient becomes

$$\nabla_\psi F(\psi) = \frac{\partial F}{\partial \psi} = \frac{1}{2}\sum_{t=1}^{P}\frac{\partial F_t}{\partial \psi} = \frac{1}{2}\sum_{t=1}^{P}\Delta(t)\mathbf{s}(t) \tag{B.30}$$

Note that the factor $\frac{1}{2}$ disappears if we substitute (B.21) and (B.26) in (B.30). All we have to do to improve the SSNN performance (on the training data set!) is move

weight vector ψ in the direction of $-\nabla_\psi F(\psi)$. The well known and most simple backpropagation weight update rule does this by applying weight updates by

$$\psi^{new} = \psi^{old} - \eta \nabla_\psi F(\psi) \tag{B.31}$$

where η is a small real number called the learning rate. Usually η is valued in the range of $[0.01, 0.1]$. The update rule is generally called the *generalized delta rule* (Hecht-Nielsen 1990).

Algorithm 3 *Back propagation training procedure*

1. *Initialize the SSNN weights ψ (random or with the Nguyen-Widrow method (Nguyen & Widrow 1990))*
2. *Present the batch of input/output data pairs $\{\mathbf{u}(t), o(t)\}_{t=1}^{P}$ to the SSNN, calculate and store F_t, $\Delta(t)$, $\mathbf{s}(t)$ for each pattern and calculate performance (eqn B.13)*
3. *Update weights ψ with eqn (B.31).*
4. *Calculate performance (eqn B.13)*
5. *If convergence criteria met (minimum performance goal, maximum number of steps, or gradient norm reached) then stop, else continue with step 2*

B.3.2 Improved training algorithm: Levenberg-Marquardt

The standard Backpropagation algorithm suffers from a number of both practical and methodological problems. The most important are its slow convergence, and its sensitivity to get stuck in local minima. Therefore in the past decades a number of advanced backpropagation algorithms for neural network training have been developed (see e.g. (Demuth & Beale 1998) for a comprehensive overview), including backpropagation algorithms with momentum and variable learning rate, conjugate gradient algorithms (e.g. Fletcher-Reeves, Polak-Ribiere), line search algorithms (e.g. Brent's and Charalambous algorithm) and Quasi-Newton algorithms. One of the fastest converging and reliable algorithms available is the Levenberg-Marquardt Algorithm (Hagan & Menhaj 1994). For very large models, the algorithm is computationally not feasible, however, for models with up to a few hundred parameters (as the SSNN used here) it is an appropriate choice.

While Backpropagation is a steepest decent algorithm, Levenberg-Marquardt is an approximation to Newtons method, which uses information on the local curvature of the error surface (second derivatives) to obtain the optimal step direction and size. The weight update rule of this algorithm is calculated as follows

$$\psi^{new} = \psi^{old} + \left|\widehat{\mathbf{H}}(\psi) + \mu \mathbf{I}\right|^{-1} \mathbf{J}^T(\psi)\mathbf{e}(\psi) \tag{B.32}$$

where

$$\mathbf{e}(\psi) = (e_1(\psi), e_2(\psi), ..., e_P(\psi))^T \; ; \; e_t(\psi) = y(t) - o(t) \tag{B.33}$$

is a column vector of all output errors on the batch of P input/output data pairs $(\mathbf{u}(t), o(t))$,

$$\widehat{\mathbf{H}}(\psi) = \mathbf{J}^T(\psi)\mathbf{J}(\psi) \approx \nabla^2_\psi \mathbf{e}(\psi) \tag{B.34}$$

is an approximation to the Hessian matrix (second derivative) of the network output errors with respect to its weights, and

$$\mathbf{J}(\psi) = \nabla_\psi \mathbf{e}(\psi) = \begin{pmatrix} \frac{\partial e_1(\psi)}{\partial \psi_1} & .. & \frac{\partial e_1(\psi)}{\partial \psi_Q} \\ .. & .. & .. \\ .. & .. & .. \\ \frac{\partial e_P(\psi)}{\partial \psi_1} & .. & \frac{\partial e_P(\psi)}{\partial \psi_Q} \end{pmatrix} \tag{B.35}$$

is the Jacobian matrix (first derivative) of the network output errors with respect to its weights. If the update (B.32) improves the performance function (eqn B.13), we decrease μ (which is a training parameter similar to the learning rate earlier) by dividing it by some factor κ (e.g. 10). If the update deteriorates performance, it is discarded and μ is multiplied by κ. The update rule (B.32) is approximately Gauss-Netwon for small μ , and approximately steepest descent for large μ.

The clue of the algorithm is in the calculation of the Jacobian matrix. We can use the standard backpropagation algorithm to do so with just one modification, that is, we calculate the delta's based on $e_t(\psi)$,rather than based on $F_t(\psi)$. This means we need to replace the term $(y(t) - o(t))^2$ in eqn (B.20) with $(y(t) - o(t))$ to obtain

$$\frac{\partial e_t(\psi)}{\partial \mathbf{w}} = \frac{\partial e_t(\psi)}{\partial I_o(t)} \frac{\partial I_o(t)}{\partial \mathbf{w}} = \delta_o(t) \frac{\partial I_o(t)}{\partial \mathbf{w}} \tag{B.36}$$

$$\delta_o(t) = \phi'(I_o(t)) \tag{B.37}$$

The sensitivities of the hidden neurons can be now be recursively calculated with equation (B.26), and the Jacobian elements are straightforwardly derived with

$$\frac{\partial e_t(\psi)}{\partial \psi} = \Delta(t)\mathbf{s}(t) \tag{B.38}$$

Each single output error must thus be propagated back through the network, producing one of the P rows of the Jacobian (B.35). The complete Levenberg-Marquardt algorithm now reads:

Algorithm 4 *Levenberg-Marquardt training procedure*

1. *Initialize the SSNN weights ψ (random or with Nguyen-Widrow (Nguyen & Widrow 1990))*

2. *Present the batch of input/output data pairs* $\{\mathbf{u}(t), o(t)\}_{t=1}^{P}$ *to the SSNN, calculate and store* e_t, $\Delta(t)$, $\mathbf{s}(t)$ *for each pattern and calculate performance (eqn B.13)*

3. *Calculate Jacobian (eqn B.32) and approximated Hessian (eqn B.32) and update weights* ψ *with eqn (B.32).*

4. *Calculate performance (eqn B.13)*

5. *If performance improved, divide* μ *by* κ*, otherwise discard weight update and multiply* μ *by* κ*; if after multiplication* μ *exceeds some threshold value stop training, else continue with step 4*

6. *If convergence criteria met (minimum performance goal, maximum number of steps, or gradient norm reached) then stop, else continue with step 2*

B.3.3 Truncated backpropagation / Levenberg-Marquardt

The one thing we have ignored so far is the functional dependence of $\mathbf{x}(t-1)$ on $\mathbf{x}(t-2)$, and thus of $\mathbf{x}(t-2)$ on $\mathbf{x}(t-3)$ and so on. Writing out this dependency would lead to a very elaborate calculation of the hidden layer sensitivities (delta's) (eqn B.24 to B.26), since the recursion through time has an in principle infinite depth. On the other hand, ignoring this dependency, leads to calculating a Gradient (B.30) and Jacobian (B.38) that are at best approximations of the true Gradient and Jacobian.

Nonetheless, for the SSNN training algorithm we choose to ignore the time recursion beyond $t-1$ and truncate calculation of the (true) Jacobian for two practical reasons. First, full calculation yields the above mentioned increased mathematical complexity and computational expense, and second, the Matlab Neural Network Toolbox (Demuth & Beale 1998), which we used for developing and training the SSNN models, only provides the "static" or truncated (but efficient and fast) Levenberg-Marquardt Algorithm. Moreover in literature a truncated version of backpropagation has been widely applied to first order context memories, e.g. (Elman 1990) The same algorithm forms the basis of the Levenberg Marquardt / Bayesian Regularization algorithm, which we have applied to SSNN training throughout this dissertation thesis.

Since our time discretization is relatively course (one time unit equals 60 seconds), we expect that the truncated algorithm is still able to capture most of the dynamics of the (short term freeway) travel time prediction problem. The results in chapters 5 on neuron relevance and memory depth support this hypothesis.

Appendix C

Bayesian Framework for Neural Networks

This appendix is largely based on the work of David MacKay ((MacKay 1992),(MacKay 1994), and (MacKay 1995)), and is provided as a reference for chapters 5, and 7. The aim of this appendix is to illustrate that neural network training, or rather model fitting in general, could be regarded as probabilistic inference producing probability densities in parameter space rather than an exact procedure producing one "optimal" vector of model parameters.

C.1 Background: Probability theory and Occam's Razor

"The language of coherent reasoning is probability theory. All coherent beliefs and predictions can be mapped onto probabilities" (MacKay 1995). As such, predicting mean travel times for vehicles departing on some freeway route given for example current and near-past measurements from some traffic data collection system should be regarded as probabilistic inference. Suppose we would have more than one travel time prediction model (say H_1 and H_2), then also making quantitative statements on which model performs best in a particular situation should be regarded as probabilistic inference.

C.1.1 Basic rules of probabilistic inference

Let the quantity $P(A|B)$ denote the *conditional* probability (a number between 0 and 1) of proposition A given proposition B [1] and $P(A, B)$ the *joint* probability (a number be-

[1] if A, B and H are continuous variables, then probabilities become probability densities and the summations in eqs (C.1), (C.2), and (C.3) become integrals.

tween 0 and 1) of A and B. Furthermore, consider the probabilities of all propositions B add to one and that all propositions B are independent. Under these circumstances, the first basic rule of probability theory is the **product rule:**

$$P(A, B) = P(A|B)P(B) \tag{C.1}$$

which relates the joint probability to the conditional probability. Second, we have the **sum rule**

$$P(A) = \sum_B P(A, B) = \sum_B P(A|B)P(B) \tag{C.2}$$

which relates the so-called *marginal distribution* of A to the joint and conditional probabilities. Therefore, in the Bayesian community, summing (integrating) over joint probabilities is often referred to as *marginalization*.What eqn C.2 effectively does, is that it allows us to get rid of the functional dependence of A on B, by integrating B out of the equation. Finally, **Bayes Theorem** combines both the previous rules:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A|B)P(B)}{\sum_B P(A|B)P(B)} \tag{C.3}$$

and allows us to re-evaluate the probability of B, after having observed A. In eqn C.2 $P(A|B)$ is the conditional probability of A given B. This term can be interpreted as the *likelihood* that A occurs given B. The term $P(B)$ denotes our prior assumptions about B, while the denominator in eqn C.2 serves as a normalization factor.

Suppose for example B denotes the probability Joe has a nasty disease and A the probability of a positive outcome of a medical test for that disease. eqn (C.1) calculates the probability that both A and B are positive, eqn (C.2) calculates the probability of a positive test outcome regardless of Joe having the disease or not , and eqn (C.3) calculates the probability of Joe having the disease given a positive test outcome. Clearly, prior to the test, the probability of Joe having the disease is smaller than after we have observed a positive test result. Similarly, the probability that Joe is healthy decreases after a positive test.

C.1.2 Occam's razor

Occam's razor is a principle, which - like the principle of Parsimony (Box & Jenkins 1976), favors simpler explanations over complex ones. Bayesian inference in terms of comparing models embodies this principle automatically and quantitatively. Moreover, this is not due to the inclusion of subjective prior information, which is often a critique on Bayesian probability theory.

The following example, taken from (MacKay 1995), illustrates the link between Occam's razor and Bayes rule. Suppose we observe some real world process F with some serial measuring device that produces the following sequence of numbers:

$$D \quad -1, 3, 7, 11, .. \tag{C.4}$$

Then suppose our task is

1. to provide mathematical theories on process F, and
2. to come up with a model predicting of the next two values produced by F based on the observed data D.

Suppose we come up with two hypotheses:

Proposition 1 *(H_1): F is an arithmetic progression of the type $x_{n+1} = x_n + k$.Based on the data we find $k = 4$, and the first number of the sequence $x_0 = -1$. The predictions of the next two numbers are hence* 15, 19.

Proposition 2 *(H_2): F is a cubic function of the type $x_{n+1} = -\frac{a}{b}x_n^3 + \frac{c}{d}x_n^2 + \frac{e}{f}$. Based on the data we find $a = 1; b = 11; c = 9; d = 11; e = 23;\ f = 11$, $x_0 = -1$ and the predictions of the next two numbers are hence* -19.9, 1043.8.

We can now use Bayes rule (C.3) to compare the plausibility of H_1 and H_2 given the observed data D. Leaving out the normalization factor implies evaluating

$$\frac{P(H_1|D)}{P(H_2|D)} = \frac{P(D|H_1)P(H_1)}{P(D|H_2)P(H_2)} \tag{C.5}$$

If we apriori (before viewing the data) find both theories equally plausible for process F, we can leave the prior probabilities $P(H_1)$ and $P(H_2)$ out of the equation and specify the first term in the numerator of (C.3), which is the likelihood of the data occurring given both hypotheses. For both tests a reasonable assumption would be that the parameters $k, a, b, c, d, e,\ f, x_0$ are all positive numbers between 1 and 50. The likelihood that this particular data set is produced by model H_1 then equals the probability that $k = 4$, and $x_0 = -1$

$$P(D|H_1) = \frac{1}{50}\frac{1}{50} = 4 \cdot 10^{-4} \tag{C.6}$$

Since there are four ways to express $\frac{1}{11}$ based on our assumptions on the parameter universe ($\frac{1}{11}$, $\frac{2}{22}$, $\frac{3}{33}$, and $\frac{4}{44}$ respectively) and likewise four and two ways to express

$\frac{9}{11}$ and $\frac{23}{11}$ respectively, the likelihood that this particular data set is produced by model H_2 similarly equals

$$P(D|H_2) = \left(4\frac{1}{50}\frac{1}{50}\right)\left(4\frac{1}{50}\frac{1}{50}\right)\left(2\frac{1}{50}\frac{1}{50}\right)\frac{1}{50} \approx 4 \cdot 10^{-11} \qquad \text{(C.7)}$$

In other words, *after observing the data D, theory H_1 is 10 million times more likely to underlie process F than theory H_2*, even if we apriori find theory H_1 and H_2 equally plausible, and assume that the parameters of both models are drawn from the same parameter universe. The reason for this is that the more complex model H_2 has a much wider range (wider but flatter distribution) of data it can produce then the simple model. Nonetheless, H_2 may still be more plausible on the basis of aesthetics, physical considerations or otherwise. But when we base our inference solely on the data, and we make explicit all our assumptions (e.g. on the parameter universe), then Bayes rule gives us solid quantitative evidence in favor of the simple hypothesis H_1. Also note that after observing $x_{n+1} = -19.9$, the odds on hypothesis H_2 improve drastically. Although Bayes rule embodies Occam's razor automatically, it favors simple models only and only if the data warrants it. If the data moves out of the predictive range of a simple model, the more complex one becomes more plausible.

Finally, if we wish to refrain from statements on the appropriateness of either model, the Bayesian approach allows one to infer over all possible theories through marginalization (eqn C.2). For example, on the basis of hypothesis H_1 and H_2, the expected (average) value of the next number process F will produce reads

$$\begin{aligned} x_{n+1} &= \frac{P(H_1|D)\,(11+4) + P(H_2|D)\left(-\frac{1}{11}11^3 + \frac{9}{11}11^2 + \frac{23}{11}\right)}{P(H_1|D) + P(H_2|D)} \\ &= \frac{1}{4 \cdot 10^{-4} + 4 \cdot 10^{-11}}(4 \cdot 10^{-4}15 + 4 \cdot 10^{-11} - 19.9) \approx 11 \end{aligned} \qquad \text{(C.8)}$$

C.2 Neural Networks as Probabilistic Models

Let us denote a neural network as a non-linear parameterized mathematical model mapping from an input(vector) $\mathbf{u}(t)$ to an output $y(t) = G(\mathbf{u}(t), \psi, H)$, where ψ denotes a vector of all parameters in the model and H denotes all other modelling assumptions (the number of layers, the number of weights/connections, the type of transfer functions used, etc.). Note that also $y(t)$ may be a vector; neural networks in principle can approximate any mapping $G \;\; \mathbb{R}^n \rightarrow \mathbb{R}^m$ to an arbitrary accuracy, albeit only under certain conditions (details on this universal approximation theorem can be found on p122 of (Hecht-Nielsen 1990)).

For a function approximation task (e.g. regression), a neural network is trained (calibrated) such that it minimizes some error function $F(\psi)$, usually expressed in terms

of the sum squared output error on a finite training data set $\{o(t), \mathbf{u}(t)\}_{t=1}^{P}$:

$$F(\psi) = E_D(\psi) \tag{C.9a}$$

$$E_D(\psi) = \frac{1}{2}\sum_{P}(y(t) - o(t))^2 = \frac{1}{2}\sum_{P}(G(\mathbf{u}(t), \psi, H) - o(t))^2 \tag{C.9b}$$

In appendix B, it is shown this function can be minimized by taking repeated steps in the direction of the negative gradient of $F(\psi)$ using the backpropagation algorithm or one of its variations. To improve generalization we add a so-called weight decay component (regularizer) to the objective function

$$E_W = \frac{1}{2}\sum_{Q}\psi_j^2 \tag{C.10}$$

where Q denotes the total number of weights in parameter vector ψ. Regularization takes into account that increasing model complexity (larger or more weights) may lead to better performance on the training data, but also to poorer generalization with respect to unseen data. The performance function then reads

$$F(\psi) = \beta E_D + \alpha E_W \tag{C.11}$$

The parameters β and α regulate to which extend the output error (the first term in equation C.11) and the size of the weights (the second term) contribute to the performance function.

Let us now view this minimization problem as probabilistic inference. Let's assume that the (output) data are the product of a Gaussian noise process with noise level $\sigma_D^2 = 1/\beta$ and that the prior distribution of our weights is also Gaussian distributed with noise level $\sigma_W^2 = 1/\alpha$. The error function (C.9b) can then be interpreted is minus the loglikelihood for a noise model

$$\frac{1}{Z_D(\beta)}\exp(-\beta E_D) = \frac{1}{(\pi/\beta)^{N/2}}\exp(-\beta E_D) \tag{C.12}$$

and the regularizer (C.10) as the log prior distribution of the parameters

$$\frac{1}{Z_W(\alpha)}\exp(-\alpha E_W) = \frac{1}{(\pi/\alpha)^{Q/2}}\exp(-\alpha E_W) \tag{C.13}$$

Minimizing the performance function (C.11) then equals minimizing the posterior probability density of the parameters, given the data D, the regularization parameters α and β, and all our other assumptions H

$$\begin{aligned} P(\psi|D, \alpha, \beta, H) &= \frac{P(D|\psi, \beta, H)P(\psi|\alpha, H)}{P(D|\alpha, \beta, H)} \\ &= \frac{1}{P(D|\alpha, \beta)}\left(\frac{1}{Z_D(\beta)}\exp(-\beta E_D)\frac{1}{Z_W(\alpha)}\exp(-\alpha E_W)\right) \\ &= \frac{1}{P(D|\alpha, \beta)}\left(\frac{1}{Z_F(\alpha, \beta)}\exp(-F(\psi))\right) \end{aligned} \tag{C.14}$$

and thus reads

$$\widehat{F}(\psi) = -\log(P(\psi|D, \alpha, \beta, H)) \tag{C.15}$$

The gain of translating neural network training into probabilistic inference is in the fact that, as with the example in the previous section, equation C.14 (which is Bayes Rule) embodies Occam's razor. Popularly speaking, maximizing (C.14) leads to the simplest setting of ψ, α, and β that is still warranted by the observed data (D).

Since we calculate a maximum probability density rather then a fixed point in parameter space, we are provided with two valuable quantities after the training procedure (in which we minimize C.15 with respect to ψ). First we obtain a parameter vector ψ^{MP} that is maximum probable given the data, our training regime and our modelling assumptions, and second we are also provided with quantitative information on the confidence we have in our ψ^{MP} prediction, again by marginalization over ψ, α and β. The predictive distribution of the neural network for a new datum at t^* then reads[2]

$$P(y(t^*)|D, \alpha, \beta, H) = \int d\psi\, P(y(t^*)|\psi, D, \alpha, \beta, H) P(\psi|D, \alpha, \beta, H) \tag{C.16}$$

where we have used the sum rule (eqn C.2) to integrate over ψ, with fixed α and β and fixed neural network architecture H.

C.3 Practical Implementation

C.3.1 Neural network training: Levenberg-Marquardt with Bayesian regularization

The regularization parameters can be updated simultaneously with the network parameters ψ with the algorithm (Levenberg-Marquardt and Bayesian regularization) described by (Foresee & Hagan 1997). For this we have to (only marginally) adapt the Levenberg Marquardt algorithm described in the previous appendix (B).

Algorithm 5 *Levenberg Marquardt and Bayesian Regularization*

1. *Initialize regularization parameters (e.g. $\beta = 1$ and $\alpha = 0$)[3], and the SSNN weights ψ (random or with Nguyen-Widrow (Nguyen & Widrow 1990))*

[2] Bayesian prediction of a new datum $y(t^*)$ should actually involve integration over ALL uncertainty (including our setting of α and β and our particular choice of neural network model H), that is $P(y(t^*)|D) = \sum_H \int d\alpha d\beta \int d\psi\, P(y(t^*)|\psi, \alpha, \beta, H) P(\psi, \alpha, \beta, H|D)$

[3] Setting $\beta = 1$ and $\alpha = 0$ is equivalent to the apriori assumption that the noise in the data is zero mean distributed with variance 1, and that the parameters are all well determined (no noise).

2. *Take a step in the Levenberg-Marquardt algorithm that minimizes performance eqn (C.11 or C.15) with fixed α and β using steps 2 up to 5 of Algorithm 4. Because the performance function (C.11) differs from the original Levenberg-Marquardt performance function (eqn B.13) in that it contains hyperparameters α and β and a regularizer term, the approximate Hessian (compare eqn B.34) is calculated with*

$$\widehat{\mathbf{H}}(\psi) = \beta \mathbf{J}^T(\psi)\mathbf{J}(\psi) + \alpha \mathbf{I} \tag{C.17}$$

where $\mathbf{I}$ *is the identity matrix of size* $Q \times Q$.

3. *Optimize α and β with given updated weights ψ^{new} by maximizing the posterior probability of of α and β given the observed data:*

$$P(\alpha, \beta|D) = \frac{P(D|\alpha, \beta)P(\alpha, \beta)}{P(D)} \tag{C.18}$$

which (if we assume a uniform prior distribution $P(\alpha, \beta)$ of α and β) is equal to maximizing the likelihood $P(D|\alpha, \beta)$ of the data D in the light of α and β. Note that this likelihood in fact is the normalization factor of eqn (C.14)! In (Foresee & Hagan 1997) it is shown that the maximum likelihood estimates for α and β can now be expressed by

$$\alpha^{ML} = \frac{\gamma}{2E_W(\psi^{new})}, \text{ and } \beta^{ML} = \frac{P - \gamma}{2E_D(\psi^{new})} \tag{C.19}$$

where

$$\gamma = Q - \alpha \cdot trace(\widehat{\mathbf{H}})^{-1} \tag{C.20}$$

is the effective number of parameters *in the SSNN model, and $\widehat{\mathbf{H}} = \nabla^2 F(\psi)$ denotes the Hessian matrix of the performance function with respect to the SSNN weights. Conveniently, in the Levenberg-Marquardt Algorithm this approximation to the Hessian matrix is already calculated (eqn C.17) as part of the weight update, so optimizing the regularization parameters does not increase the computational expense of the algorithm more than marginally.*

4. *If convergence criteria met (minimum performance goal, maximum number of steps, maximum value of μ or gradient norm reached) then stop, else continue with step 2*

C.3.2 Error bars

If we assume a Gaussian posterior[4] for the weights $N(\psi^{MP}, \Sigma_\psi)$ and a Gaussian noise model for the targets, we can obtain error bars by local linearization around the SSNN output:

$$y(t) = G(\mathbf{u}(t), \psi, H) \simeq G(\mathbf{u}(t), \psi^{MP}, H) + \mathbf{g}(\psi - \psi^{MP}) \tag{C.21}$$

[4]The Bayesian method does not require Gaussian approximation. The posterior distribution (C.14) may be of any type or form. As a consequence, it can usually only be approximated numerically (e.g. with Monte Carlo Methods). Assuming Gaussian posteriors allows us to proceed analytically.

in which **g** is the sensitivity of the output to the parameters, that is the first derivative of the neural network with respect to its weights $\frac{dG}{d\psi}$. Based on eqn (C.21), expression (C.16) becomes a Gaussian integral with mean $G(\mathbf{u}(t), \psi^{MP}, H)$ and variance

$$\sigma^2_{y(t)|\alpha,\beta} = \mathbf{g}^T \widehat{\mathbf{H}}^{-1} \mathbf{g} + \sigma^2_o \tag{C.22}$$

in which σ^2_o denotes the nose level in the target distribution. Note that the Hessian $\widehat{\mathbf{H}}$ is calculated during neural network training, and is hence obtained automatically. That the variance due to the model parameters is proportional to the inverse of the Hessian (second derivative of performance function with respect to the parameters - eqn C.17) is straightforward, since the Hessian matrix reflects the curvature of the performance function. If the Hessian around the optimal parameter vector contains small values this implies a shallow performance function in that parameter region (a flat distribution), which suggests there are parameter vectors in the neighborhood of the current one that perform almost as well. The inverse of the Hessian then inherently contains large values, reflecting larger uncertainty. Vice versa, if the curvature is large (large Hessian, small inverse Hessian), we are at a distinct peak in parameter space and the uncertainty with respect to this particular parameter vector is small.

The output sensitivities **g** can be calculated similarly to the Jacobian matrix of output errors with respect to the neural network weights (eqn B.35) during training. The only difference is that we do not feed the output error $e(\psi) = y(\psi) - o$ back into the network, but rather just the output $y(\psi)$. Multiplying the Hessian with the square of these gradients ensures that the error-bars reflect uncertainty with respect to those parameters that are well determined (that actually "do" something) only.

Appendix D

LWR / Kalman Filter for Data Cleaning

The Kalman filter can be described as the linear minimum variance estimator of the state vector (refer to for example (Bozic 1979), (Kalman 1960), or chapter 8 of (Brockwell & Davis 1996)) and allows the modeler to combine (linear) state space models and measurement data, given that the variance components in both the model and in the real life process are known. In case of non-linear state and / or output equations (such as in macroscopic traffic flow models), an extended Kalman Filter is required, in which local linearization of both the state dynamics and the system output around the current predicted state is applied. Although there exists no convergence proof for extended Kalman Filters, they have been widely and successfully applied for traffic control and prediction (e.g. (Hoogendoorn 2000), (Wang et al. 2003)).

D.1 The Extended Kalman Filter

Consider a (non-linear) state space model in which the temporal evolution on time instances $p \in \{0, 1, 2, ...\}$ of a state vector $\mathbf{x} \in \mathbb{R}^n$ and the associated output vector $\mathbf{y} \in \mathbb{R}^m$ is defined by

$$\mathbf{x}(p+1) = F(\mathbf{x}(p), \mathbf{u}(p)) + \xi(p) \tag{D.1a}$$

$$\mathbf{y}(p) = G(\mathbf{x}(p)) + \zeta(p) \tag{D.1b}$$

in which F and G are nonlinear mappings $\mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\mathbb{R}^n \rightarrow \mathbb{R}^m$ respectively. The "disturbance" or "control" vector $\mathbf{u}(p)$ is constituted of inputs exogenous to the process. It is assumed that $\xi(p)$ and $\zeta(p)$ are zero mean Gaussian error vectors with

known covariance structure

$$\left\langle \xi(p)\xi(l)^T \right\rangle = \mathbf{S}(p)\delta_{pl} \tag{D.2}$$
$$\left\langle \zeta(p)\zeta(l)^T \right\rangle = \mathbf{R}(p)\delta_{pl} \tag{D.3}$$
$$\left\langle \xi(p)\zeta(l)^T \right\rangle = \mathbf{T}(p)\delta_{pl} \tag{D.4}$$
$$\delta_{kl} = \begin{cases} 1 & p = l \\ 0 & otherwise \end{cases} \tag{D.5}$$

in which $\mathbf{S}(p)$ and $\mathbf{R}(p)$ are nonnegative and positive definite matrices respectively. Furthermore, the state $\mathbf{x}(0)$ is uncorrelated to both $\xi(0)$ and $\zeta(0)$ and considered a Gaussian random variate with mean

$$\widehat{\mathbf{x}}(0) = \langle \mathbf{x}(0) \rangle \tag{D.6}$$

and covariance matrix

$$\Sigma(0) = \left\langle [\mathbf{x}(0) - \widehat{\mathbf{x}}(0)] [\mathbf{x}(0) - \widehat{\mathbf{x}}(0)]^T \right\rangle \tag{D.7}$$

Local linearization of equations (D.1a) and (D.1b) around $\widehat{\mathbf{x}}(p)$ yields

$$F(\mathbf{x}(p), \xi(k)) \approx \mathbf{A}(p)\widehat{\mathbf{x}}(p) + \mathbf{D}(p)\xi(p) \tag{D.8a}$$
$$G(\mathbf{x}(p)) \approx \mathbf{C}(p)\widehat{\mathbf{x}}(p) \tag{D.8b}$$

in which

$$\mathbf{A}(p) = \frac{\partial F}{\partial \mathbf{x}} \mid \mathbf{x} = \widehat{\mathbf{x}}(p); \xi = 0 \tag{D.9}$$
$$\mathbf{C}(p) = \frac{\partial G}{\partial \mathbf{x}} \mid \mathbf{x} = \widehat{\mathbf{x}}(p) \tag{D.10}$$
$$\mathbf{D}(p) = \frac{\partial F}{\partial \xi} \mid \mathbf{x} = \widehat{\mathbf{x}}(p); \xi = 0 \tag{D.11}$$

The (extended) Kalman Filtering algorithm now consists of two steps: the *prediction step* (1) and the *correction step* (2)

1. First, the most likely estimate of the next state is predicted with

$$\widehat{\mathbf{x}}(p+1) = F(\mathbf{x}(p), \mathbf{u}(p)) \tag{D.12}$$

   which is simply the state dynamics equation without noise. Subsequently, the covariance matrix of state estimation errors is estimated with

$$\widehat{\Sigma}(p+1) = \mathbf{A}(p)\Sigma(p)\mathbf{A}(p)^T + \mathbf{D}(p)\mathbf{S}(p)\mathbf{D}(p)^T \tag{D.13}$$

2. Set $p = p + 1$. Based on observations $\mathbf{y}(p)$, the state prediction is corrected with

$$\mathbf{x}(p) = \widehat{\mathbf{x}}(p) - \mathbf{K}(p)\left(\mathbf{y}(p) - G(\widehat{\mathbf{x}}(p))\right) \tag{D.14}$$

in which the so called Kalman Filter Gain

$$\mathbf{K}(p) = \left[\mathbf{A}(p)\widehat{\Sigma}(p)\mathbf{C}^T(p) + \mathbf{D}(p)\mathbf{T}(p)\right]\left[\mathbf{C}(p)\widehat{\Sigma}(p)\mathbf{C}^T(p) + \mathbf{R}(p)\right]^{-1} \tag{D.15}$$

determines the magnitude of each correction. If elements $K_{ij}(p)$ are (near) zero, then the Kalman Filter assigns more confidence in the associated model prediction, whereas large values $K_{ij}(p)$ reflect more confidence in the associated measurements. Finally, the covariance matrix of estimation errors is updated with

$$\Sigma(p) = \left[\mathbf{I} - \mathbf{K}(p)\right]\widehat{\Sigma}(p) \tag{D.16}$$

Starting from $p = 0$, we can recursively apply these equations.

D.2 Lighthill, Witham and Richards Traffic Flow Model

First order traffic flow theory centers on the concept of 'conservation of vehicles' (CoV), analogously to the behavior of fluids in tubes. The resulting kinematic wave model, first introduced by (Lighthill & Whitham 1955), contains a continuous version of the CoV (in this case $t \in [0, \rightarrow]$):

$$\frac{d}{dt}\rho(x,t) + \frac{d}{dx}Q^e(\rho(x,t)) = 0 \tag{D.17}$$

where $\rho(x,t)$ depicts vehicular density (no vehicles per unit space) at location x and time instant t. $Q^e(\rho)$ depicts the equilibrium traffic flow (no vehicles per unit time) as a function of density. Usually $Q^e(\rho)$ (often referred to as fundamental diagram) is a concave function which has a maximum (capacity flow) q_C at the so called critical density ρ_C. This critical density separates free flowing from congested conditions. Finally, mean (equilibrium) speed is calculated through

$$V^e(\rho(x,t)) = Q^e(\rho(x,t))/\rho(x,t) \tag{D.18}$$

The discretized LWR model can be expressed in state space form as follows:

$$\rho_k(t+\Delta t) = \rho_k(t) + \frac{\Delta t}{L_k}(q_k^{in}(t) - q_k^{out}(t) + r_k^{in}(t) - r_k^{out}(t)) \tag{D.19}$$

$$q_k(t) = Q^e\left(\rho_k(t)\right) \tag{D.20}$$

$$u_k(t) = Q^e\left(\rho_k(t)\right)/\rho_k(t) \tag{D.21}$$

where $\rho_k(t)$ depicts the number of vehicles on section k, $q_k^{in}(t)$ and $q_k^{out}(t)$ denote the vehicles entering and leaving section k (over the main carriage way) during time period $[t, t + \Delta t]$, while $r_k^{in}(t)$ and $r_k^{out}(t)$ denote the inflow and outflow at on - ramps and off ramps respectively. Furthermore $u_k(t)$ depicts the space mean speed on k and L_k denotes the length of section k. Note that $q_k(t)$ denotes the (average) flow on section k. We adopt the triangular fundamental diagram proposed by amongst others Daganzo (Daganzo 1997) (Fig D.1).

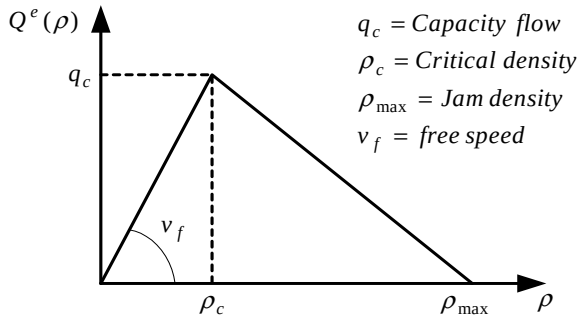


Figure D.1: Triangular fundamental diagram of traffic flow

For numerical stability, a proper choice of time step Δt is crucial. As a general rule, Δt should be smaller than the minimum time required for a vehicle to traverse the smallest section, or more formally: $\Delta t \leq u_k^{\max}(t)/L_k, \forall k, t$. For sections of several hundred metres a safe choice would be $\Delta t = 1\,\text{sec}$. A final note needs to be made about the calculation of $q_k^{in}(t)$ and $q_k^{out}(t)$. For each section the inflow equals the outflow of the previous section (if present). Likewise, outflow equals the inflow into the next section (if present). For a first order traffic flow model the most widely used numerical solution scheme of balancing in- and outflow is the so-called Godunov scheme (e.g. (Lebacque 1996)). In essence the Godunov scheme allows the minimum possible flow to transfer between two adjacent sections. In Fig D.2 a schematic presentation of the Godunov scheme is given.

D.3 LWR / Kalman Filter algorithm

Suppose we consider a route of M adjacent freeway sections each equipped with one (!) inductive loop detector measuring flows and average speeds. Fig. D.3 illustrates how this would be done. Note that the underlying assumption is that on each section homogeneous and stationary conditions apply within a measurement period $p = \{1, .., P\}$.

As a state vector we now define the densities on each of those M freeway sections

$$\mathbf{x}(p) = \left[\rho_1(p), ..., \rho_M(p)\right]^T \tag{D.22}$$

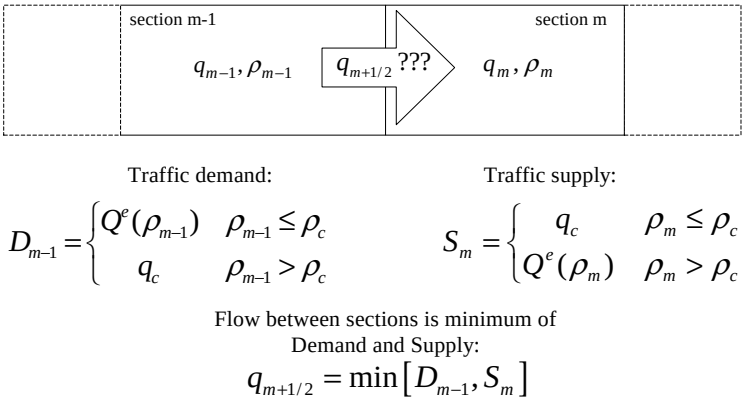


Figure D.2: Schematic representation of the Godunov scheme for first order traffic flow models (for details see e.g. (Lebacque 1996))

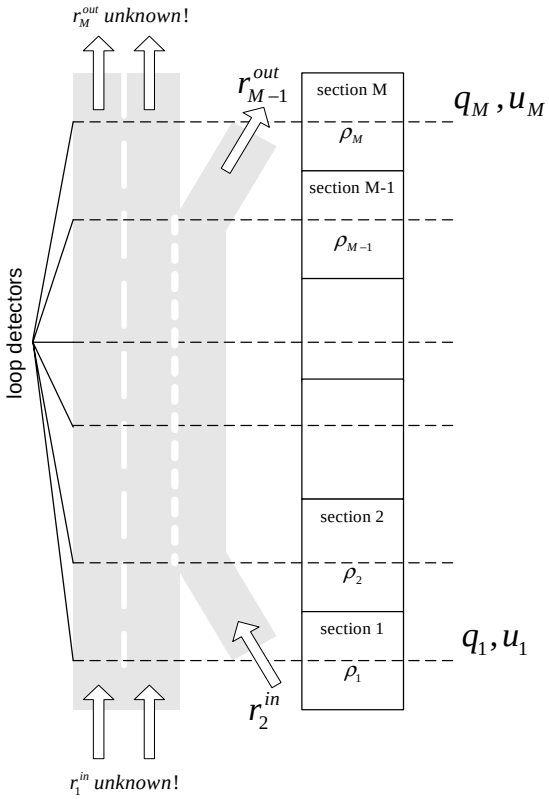


Figure D.3: Network setup for the LWR / Kalman Filter data cleaninbg procedure. A section contains one inductive loop detector, meaning there are as much sections as there are detectors.

As outputs we define the flows and speeds (which we measure) on each of these sections, that is

$$\mathbf{y}(p) = \left[q_1(p), ..., q_M(p), u_1(p), ..., u_M(p)\right]^T \tag{D.23}$$

The state vector $\mathbf{x}(p)$ hence is a $M \times 1$ column vector, the output $\mathbf{y}(p)$ a $2M \times 1$ column vector. Finally, as exogenous inputs to this system we take the in- and outflows at on and off ramps (including traffic demand at the first section and outflow at the last)

$$\mathbf{u}(k) = \left[r_1^{in}(p), ..., r_M^{in}(p), r_1^{out}(p), ..., r_M^{out}(p)\right]^T \tag{D.24}$$

which is again a $2M \times 1$ column vector. The problem to overcome is that the LWR model requires small time steps Δt for numerical approximation, while the Kalman Filter can only be applied at time steps p (of typically 60 seconds) where measurements are available. If we assume constant in- and outflows at on- and off ramps within each measurement period k of 60 seconds, and a time discretization for the LWR model of 1 second we can run the LWR/Kalman Filter with a discretization of $p = 1$ seconds, simulating as if every second a measurement becomes available.

Recalling equations (D.19), (D.20) and (D.21), the state equation can be written in the desired form

$$\mathbf{x}(p+1) = F(\mathbf{x}(p), \mathbf{u}(p)) + \xi(p) \tag{D.25}$$

in which F is the CoV equation (D.19) and is solved by the Godunov scheme as outlined in fig. D.2. Note that the elements of $\mathbf{u}(p)$ (eqn D.24) have the dimension veh/s. The output equation reads

$$\mathbf{y}(p) = G(\mathbf{x}(p)) + \zeta(p) \tag{D.26}$$

in which G is the fundamental diagram (eqn D.20) for the flow elements in $\mathbf{y}(p)$ and equals (eqn D.21) for the speed elements in $\mathbf{y}(p)$. For the Kalman filter to be completed we still need to determine (approximate) the matrices $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{D}$. These are obtained by local linearization of equations (D.19) and (D.20) around $\widehat{\mathbf{x}}(p)$.

$$\begin{aligned}
\mathbf{A}(p) &= \frac{\partial F}{\partial \mathbf{x}} \mid \mathbf{x} = \widehat{\mathbf{x}}(p); \xi = 0 && \text{(D.27)}\\
&= \left\{A_{ij}(p)\right\},\ i,\ j \in \{1,..,M\}\\
A_{ij}(p) &= \begin{cases} 1 & i = j \\ 0 & otherwise \end{cases}, \forall p\\
\mathbf{D}(p) &= \frac{\partial F}{\partial \xi} \mid \mathbf{x} = \widehat{\mathbf{x}}(p); \xi = 0 && \text{(D.28)}\\
&= \mathbf{A}(p)\\
\mathbf{C}(p) &= \frac{\partial G}{\partial \mathbf{x}} \mid \mathbf{x} = \widehat{\mathbf{x}}(p) && \text{(D.29)}\\
&= \left\{C_{ij}(p)\right\},\ i \in \{1,..,2M\},\ j \in \{1,..,M\}\\
C_{ij}(p) &= \begin{cases}
\upsilon_f & \rho_j(p) < \rho_c & , i \leq M,\ i = j \\
-\frac{q_c}{\rho_{\max}-\rho_c} & \rho_j(p) \geqslant \rho_c & , idem \\
\frac{\upsilon_f}{\rho_j(p)} - \frac{Q^e(\rho_j(p))}{(\rho_j(p))^2} & 0 < \rho_j(p) < \rho_c & , i > M, , i - M = j \\
\frac{-\left(\frac{q_c}{\rho_{\max}-\rho_c}\right)}{\rho_j(p)} - \frac{Q^e(\rho_j(p))}{(\rho_j(p))^2} & \rho_j(p) \geqslant \rho_c & , idem \\
0 & & otherwise
\end{cases}
\end{aligned}$$

Note that the matrices $\mathbf{A}, \mathbf{D}$ are of size $M \times M$, and $\mathbf{C}$ of size $2M \times M$ respectively. The LWR / Kalman data cleaning algorithm involves the following steps

Algorithm 6 *LWR Kalman Data Cleaning procedure*

1. *Preparation*
   (a) *Estimate the parameters of the (in our case triangular) fundamental diagram, based on observed flows and densities (occupancies), that is (again in our case), υ_f (free speed), ρ_c (critical density) and $\rho_{\max}$ (jam density). The capacity flow follows from these three: $q_c = Q^e(\rho_c)$.*
   (b) *Make an initial estimate for the variance covariance matrix of the density estimation errors $\Sigma(0)$.*
   (c) *Set $p = 0$, and Initialize state vector $\mathbf{x}(p)$ (e.g. by means of $\rho = Q^e(q)/u$)*
2. *Algorithm*
   (a) *Estimate all variance and covariance matrices, that is $\mathbf{S}(p), \mathbf{R}(p)$ and $\mathbf{T}(p)$.These may be constructed based on specifications of the MONICA inductive loop detectors, or by engineering judgement. In our case we assume that each detector produces random errors of 1 veh/s and 1 m/s for their flow and speed measurements respectively and that the covariance between detectors vanishes linearly with their respective distance. In case these ,matrices are not considered time dependent, they may be estimated on beforehand and remain constant.*

(b) Set traffic demand upstream $r_1^{in}(p) = q_1(p)$ *and likewise traffic supply downstream* $r_M^{out}(p) = q_M(p)$

(c) Prediction step: predict next state (without noise components!)

$$\widehat{\mathbf{x}}(p+1) = F(\mathbf{x}(p), \mathbf{u}(p), 0) \tag{D.30}$$

and update covariance matrix of estimation errors ($\mathbf{A}(p)$ *and* $\mathbf{D}(p)$ *are left out of the equation since both are regarded unity matrices in this case)*

$$\widehat{\Sigma}(p+1) = \Sigma(p) + \mathbf{S}(p) \tag{D.31}$$

(d) Correction step: set

$$p = p + 1$$

and calculate Kalman Filter Gain (leaving out unity matrices $\mathbf{A}$ *and* $\mathbf{D}$*).*

$$\mathbf{K}(p) = \left[\widehat{\Sigma}(p)\mathbf{C}^T(p) + \mathbf{T}(p)\right]\left[\mathbf{C}(p)\widehat{\Sigma}(p)\mathbf{C}^T(p) + \mathbf{R}(p)\right]^{-1} \tag{D.32}$$

Note: if measurements $y_i(p)$ *are missing, then the measurement noise associated with those inputs is set to infinite*[1]*, that is* $R_{mn}(p) = \infty, \forall\, m = n = i;\ T_{mn}(p) = \infty, \forall\, m = i$*. Alternatively, one could set the appropriate measurement errors to zero, that is* $\mathbf{y}(p) - G\left(\widehat{\mathbf{x}}(p)\right) = 0$*. Now correct state vector*

$$\mathbf{x}(p) = \widehat{\mathbf{x}}(p) - \mathbf{K}(p)\left[\mathbf{y}(p) - G\left(\widehat{\mathbf{x}}(p)\right)\right] \tag{D.33}$$

and covariance matrix of estimation errors

$$\Sigma(p) = \left[\mathbf{I} - \mathbf{K}(p)\right]\widehat{\Sigma}(p) \tag{D.34}$$

(e) Continue with step 2a until $p > P$ *(the number of periods)*

[1] in some programming environments a large positive number would also suffice

Appendix E

Regiolab Delft

This text is taken from (Van Zuylen & Muller 2002)

RegioLab Delft is a traffic laboratory with many participating organizations. Road authorities (the Netherlands Ministry of Transport, Public Works and Water Management the province of Zuid-Holland, municipality of Delft), representatives of the traffic industry (Siemens and Vialis) and research and educational institutes (the Test center for traffic systems of the Ministry of Transport, CONNEKT, Research school TRAIL and the Delft University of Technology) work together. The RegioLab Delft aims to collect traffic data from the region of Delft, to analyze the data and to integrate the information. Existing detection equipment, such as the motorway loop detection system Monica of the Ministry and loop detection at controlled intersections in Delft, are extended with new and sometimes experimental means to detect traffic.

Based on these means of detection it will possible to recognize traffic patterns, to find origin destination relations, to measure road user's reaction on dynamic traffic measurements, and to measure and predict travel times. RegioLab Delft provides the participating organizations and if requested also the national and regional traffic information centers (TIC) with combined information about the traffic condition in the total area (managed by different road authorities). Based on this information, existing dynamic models can be validated and new models developed. As an example, all results reported in chapter 8 are based on data from the Regiolab Delft laboratory.

Summary

This dissertation focusses on short term travel time prediction for freeways for Advanced Traffic Informations Systems (ATIS), such as variable message signs (VMSs). The ultimate purpose of ATIS systems is to enable more informed individual driver behavior e.g. in terms of departure time and route choice. In turn it is hypothesized these more intelligent individual choices also lead to benefits in terms of collective traffic operations. There are however a few key requirements for these individual and collective benefits to actually occur. In terms of the traffic information the following criteria are relevant

Unambiguity Drivers should be able to understand and unambiguously (at least predictably) interpret the information. We argue this implies that travel time is most appropriate choice for information on VMSs, instead of for example queue lengths

Validity If drivers understand the information (travel time on route r for departure time p), they have to conceive it as complying with their own experiences (for example the travel time they believe is to be expected on route r). The information (travel time on route r for departure time p) should also objectively comply with what actually occurred (the actual travel time on route r for departure time p)

Objective validity is also a requirement for models producing the traffic information (e.g. predicted travel times). For both traffic information and these underlying models a set of objective criteria can be identified.

Accuracy The difference between what actually happened and what was displayed should be as small as possible (in this sense accuracy is closely related to validity). Small as possible is of course an indicator that may be location and application specific[2]

Robustness The model producing the information should be able to deal with different (traffic) conditions (free flowing, congested, incidents, etcetera). Also, if the data

[2]On a journey of three hours a 5 minute travel time prediction error is negligeable, however, for a trip 5 minutes a model making a five minute error would be considered very inaccurate.

to a model is corrupt, the model should still be able produce reasonable outcomes (which could even be a message indicating something is wrong).

Adaptivity Traffic processes are characterized by constant change, due to (structural) changes in both traffic demand patterns as wel as traffic supply characteristics. The model should be able to track these changes and adapt (its parameters) accordingly to preserve its robustness, accuracy and validity.

Reliability Although interpreted in many different ways, we view reliability here as "the mother of all qualities". Hence a reliable model is adaptive, robust, valid and accurate under all prevailing (traffic) conditions.

As such, in this thesis we present a reliable framework for short term freeway travel time prediction (fig. E.2). This framework consist of a number of components (A-F), which will be introduced below.

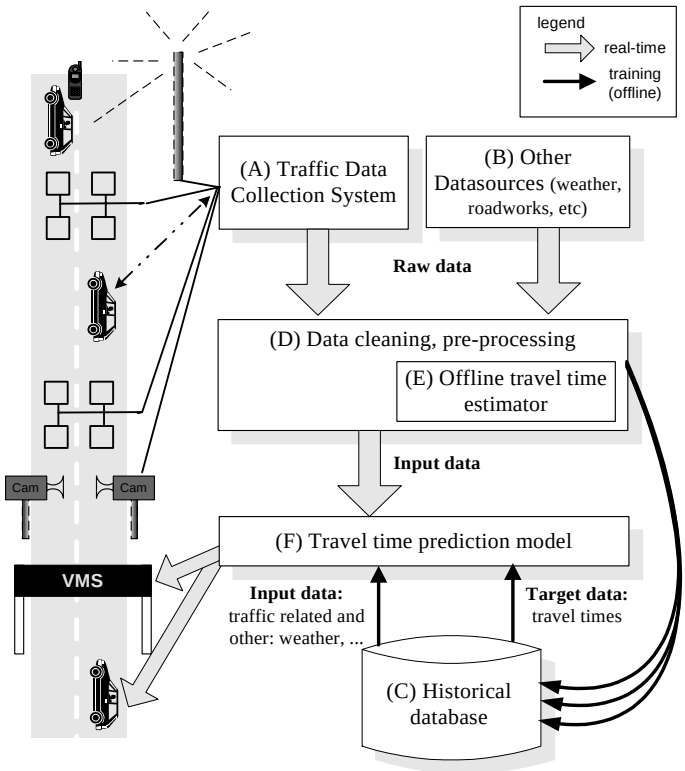


Figure E.2: Reliable framework for short term freeway travel time prediction

Predicting travel time requires that we somehow are able to measure travel time, such that the travel time prediction model can be calibrated and validated. Travel times can for example be measured with camera based systems in conjunction with license plate recognition software, or be obtained from Automated Vehicle Identification (AVI) systems operational on some toll facilities. If no travel time measurement system is

available, travel times can still be deduced from the quantities that are measured. This technique is referred to as *travel time estimation* and is a tool that can only be used *offline.* The requirement for a travel time prediction model thus is that there is some traffic data collection system (fig. E.2 (A)) installed which either measures travel times directly, or which measures quantities from which we can estimate travel times offline.

A typical example of such a traffic data collection system is the so-called MONItoring CAsco (MONICA) system, with which most Dutch freeways are equipped. The system comprises of so-called inductive loop detectors measuring speeds and flows on cross sections every 500 - 2000 metres on a minute basis. Since there is a straightforward mathematical relationship between the *harmonic* mean speed in a time period p at a detector and the travel time vehicles experience passing a section k (of say 500 metres) around that detector during period p it is possible to offline estimate travel times from speeds. Unfortunately, in MONICA local *arithmetic* mean speeds are recorded, which is computationally more inefficient and, which is worse, leads to a biased estimate (underestimation) of travel time on section k. We therefore propose an algorithm to correct for this bias, which is based on estimating speed variance through local density and a differenced time series of arithmetic mean speeds. It turns out this algorithm effectively removes the bias and enables us to estimate section level travel times fairly accurately. Next, we introduce an algorithm that uses these section level travel times to derive travel times on entire freeway routes, constituted of adjacent sections, each equipped with inductive loops. This so-called Piece-wise Linear speed based (PLSB) trajectory algorithm (fig. E.2 (E)) then enables us to gather a large historical database of travel times (fig. E.2 (C)). This historical database contains records for a large number departure time periods p with in each record the travel time for vehicles departing in p on freeway route r and the measured (and corrected) mean speeds and flows (no vehicles passing per period) of that same period and also info from other datasources (fig. E.2 (B)).

With this historical database we can now calibrate and validate a model that predicts mean travel time on route r in a particular period p. In contrast with travel time estimation, which is merely the translation of other traffic variables (e.g. speeds) into travel time, travel time prediction is a highly complex and dynamic problem, since travel times are determined by complex non-linear interactions of heterogeneous groups of driver-vehicle combinations, each characterized by their own specific technical and behavioral properties, such as vehicle dimensions and acceleration characteristics, drive-style (aggressive, conservative), and motive. To predict the mean travel time $\tau_r(p_0)$ of vehicles starting a particular route r at some departure period p_0, we need to know whether or not they will encounter delays (congestion) during the period $\left[p_0, p_0 + \tau_r(p_0)\right]$ along their route. The problem, however, is that it is exactly that quantity $\tau_r(p_0)$ (= time spent on the route = travel time), which we wish to predict in the first place. In this sense, travel time prediction is a "chicken and egg" type of problem. In literature this problem is solved either with sophisticated traffic flow models and prediction algorithms for the boundary conditions, or intelligent inductive (data

driven) models that are able to learn the complex traffic dynamics from data on the route of interest directly.

We propose a recurrent neural network model which learns these complex spatiotemporal dynamics from data (historical database). Choosing a neural network type of model, intrinsically satisfies the adaptivity criterion, since such models can be recalibrated if circumstances require this. The structure of this so-called state space neural network (SSNN) model (fig. E.2 (F)), however, is based on traffic flow theory. First of all, the model is formulated in state space form, analogously to traffic flow simulation models. This allows the model to predict travel time based on current input in the context of its previous internal states. Secondly, the SSNN topology is based on the geometry and detector lay-out of the route of interest, which makes the model generic and applicable on all freeway stretches. Since we do not make apriori assumptions on which data from which detector on the route may be relevant for the travel time prediction problem, we require a training (i.e. calibration) algorithm that is capable of selecting the appropriate inputs. Moreover, since the SSNN is a non-linear model with many degrees of freedom there exist (in principle) infinite parameter settings that would fit a particular travel time prediction problem well. We adopt the Bayesian framework for neural network training for this purpose. This framework consistently succeeds in distinguishing between relevant and non-relevant parameters and guarantees a parameter setting which is warranted by the data. For example, from the 735 parameters in the SSNN model we applied in a real case, only 20-25% were dubbed effective after the Bayesian regulated training scheme, a reduction in model complexity of up to 80%. It turns out that due to this training algorithm and the state space structure of the SSNN, the internal workings of the SSNN are in fact closely related to the actual traffic processes on the route interest. The parameter setting after training (consequently!) indicates the SSNN classifies traffic patterns in a way that makes sense from a traffic theory point-of-view. We also introduce a measure to determine the relevance of each of the SSNNs internal states. It appears only neurons connected to regularly heavily congested freeway sections (on which vehicles encounter the most delay) are relevant to the SSNN model, while other neurons contribute only marginally.

Besides theoretically and scientifically interesting, the SSNN model also is capable of accurate forecasts of travel times both on synthetic data as well as in a real case, a densely used 13 kilometer freeway stretch in The Netherlands equipped with the MONICA inductive loop system. In this case study we trained the SSNN with a historical database of two years of (8-hour) afternoon peaks (approximately $375,000$ records) and tested the model on a separate data set of four months of afternoon peak sequences ($41,000$ records). On that test data set the model produces almost unbiased results with a standard error of 8%. This is twenty times more accurate than current (instantaneous) approaches installed on the Dutch freeway network. Compared to some state-of-the-art models from literature, the SSNN proves equally or slightly more accurate. Moreover, in conjunction with simple data correction and completion algorithms, the SSNN model is robust with respect to missing and / or corrupt data. In illustration,

in MONICA the average percentage of data corruption for each departure time period on this stretch (with 27 detectors) is 12%, with regular extremes to as much as 25%.

It is also possible to quantify (in a statistical sense) the uncertainty associated with the short term freeway travel time prediction framework. We identified three sources that contribute to that uncertainty. The first component (model confidence) is due to the parameters of the SSNN travel time prediction model, which as said are by no means unique (certainly not apriori). The second uncertainty source is due to the fact we train the SSNN with estimated rather than real travel times, while the third relates to the distribution of travel times itself. In a departure time period not all vehicles will experience the same travel times due the for example differences in drive style.

Model confidence is obtained automatically by using the Bayesian method for SSNN training. We argue the main value of these confidence intervals is not in their (nominal) coverage of mean observations but in their quality as indicator for the amount of uncertainty associated with a particular prediction. Uncertainty in this sense reflects in fact the magnitude of the error the SSNN makes. We show that in case of missing data, but also in case the SSNN is confronted with traffic conditions that are partially outside its input-domain (it hasn't "seen" these during training), these confidence intervals grow larger, indicating the SSNN is less certain of its predictions. In these cases, indeed the prediction errors are larger. Practically this means that in real-time operation, confidence levels provide an automatic and quantitative "warning" mechanism for traffic managers, with which the model as well as the data with which it is fed can be monitored without having to measure actual travel times[3].

Finally, there are a number of limitations to the SSNN framework in its current form. Some of these, however, can be solved by straightforward extensions to the model. For example, one could account for the effect on travel time of traffic processes elsewhere on the network, the effect of weather or other external factors and for example the effect of traffic control (e.g. automated speed limiting or intersection control). We do recommend however, a larger scale study to further improve the offline travel time estimation procedure. Such a study would require larger scale travel time measurements on a number of different freeway stretches. We also illustrate how the SSNN framework could be applied with different traffic data collection systems, such as automatic vehicle identification systems (AVI) and floating car data (FCD).

In conclusion, in this dissertation thesis we present a reliable, valid, accurate and robust framework for short term freeway travel time prediction, which is generic in the sense that it can be applied on any freeway stretch equipped with traffic data collection systems. The central component is a state space neural network (SSNN), which is also generic in the sense that its design not location specific but related to the geometry and detector configuration. In many respects this SSNN model operates like a regular traffic flow model, with the key difference that it learns to model complex traffic processes directly from data.

[3]Even in case actual travel times *are* measured, they are available only afterwards, that is after vehicles have traversed the route

Samenvatting

Dit proefschrift beschrijft een betrouwbare methode voor online reistijdvoorspelling op snelwegen, te gebruiken voor geavanceerde reisinformatie systemen (GRS), zoals bijvoorbeeld dynamische route informatie panelen (DRIPs), of actuele verkeersinformatie websites. De achterliggende veronderstelling van dat soort systemen is dat *geïnformeerde* reizigers slimmer kunnen kiezen tussen de voor hen beschikbare alternatieven, in termen van bijvoorbeeld route of vertrektijdstip. In potentie zijn er ook positieve *collectieve* gevolgen te verwachten op basis van al die individueel slimmere beslissingen. Voorbeelden daarvan kunnen zijn vermindering van congestie (op de korte termijn) en een toename in de reistijdbetrouwbaarheid (ook op de langere termijn).

Er zijn echter wel een aantal voorwaarden waaraan een GRS zou moeten voldoen om die positieve collectieve effecten teweeg te brengen. Voor wat betreft de verkeersinformatie gaat het met name om de volgende eisen

Begrijpelijkheid Reizigers moeten in staat zijn de aan hen verstrekte informatie te begrijpen en deze (op de door de verstrekker bedoelde of voorspelde) wijze te interpreteren. Dat betekent ons inziens dat *reistijd-informatie* sterk de voorkeur heeft boven *file-informatie*.

Validiteit Als reizigers de informatie begrijpen (bijv. de reistijd op een snelweg route r bij vertrek in periode p), is het noodzakelijk dat die overeenkomt met datgene wat men verwacht. M.a.w. de informatie moet mogelijk geacht worden vanuit de ervaring en perceptie van het individu. Behalve subjectief valide moet de informatie ook objectief valide zijn, dat wil zeggen, ze moet overeenkomen met datgene wat werkelijk plaatsvond (bijv. de werkelijk opgetreden (gemiddelde) reistijd op route r bij vetrek in periode p).

Objectieve validiteit is ook een randvoorwaarde voor het model dat de reistijden voorspelt. Daarnaast zijn voor zo'n model van belang:

Nauwkeurigheid Het verschil tussen dat wat werkelijk optrad en datgene wat voorspeld is moet zo klein mogelijk zijn, waarbij "zo klein mogelijk" natuurlijk re-

latief is aan de lengte van de route, de gemiddelde reistijd op die route en het soort applicatie dat bediend wordt[2].

Robuustheid Het model dat de informatie produceert zou in staat moeten zijn goed te functioneren onder verschillende condities (vrije afwikkeling, congestie, incidenten, ongelukken, etc.). Ook moet het model om kunnen gaan met het feit dat online verkeersgegevens nogal wat fouten bevatten en vaak zelfs gedeeltelijk ontbreken.

Adaptiviteit Verkeerspatronen zijn onderhevig aan continue korte en lange termijn veranderingsprocessen, onder invloed van bijvoorbeeld infratrstructerele verbeteringen, veranderend rijgedrag, snellere en veiliger voertuigen. Een model dat reistijden op snelwegen voorspeld moet mee veranderen, dwz. aangepast kunnen worden als de omstandigheden daartoe aanleiding geven.

Betrouwbaarheid Alhoewel in de literatuur veel verschillende definities van betrouwbaarheid bestaan, interpreteren wij betrouwbaarheid als "de moeder van alle eisen", dat wil zeggen, een betrouwbaar model is valide, nauwkeurig, robuust en adaptief onder de meest voorkomende condities.

Daartoe is in dit proefschrift een raamwerk voor betrouwbare reistijdvoorspelling op snelwegen ontwikkeld dat schematisch is weergegeven in figuur E.2. De componenten in dat raamwerk worden hieronder besproken.

Om reistijden te voorspellen is het allereerst noodzakelijk reistijden te *meten*, zodat een voorspellend model kan worden gecalibreerd en gevalideerd op basis van die metingen. Reistijden meten kan bijvoorbeeld met camerasystemen en geavanceerde beeldherkenning methoden. Voertuigen "gezien" op een locatie kunnen dan worden teruggevonden op de volgende, waarna de reistijd van dat voertuig tussen die twee locaties kan worden afgeleid. Andere voorbeelden zijn GSM of GPS gebaseerde systemen waarmee individuele voertuig trajectorieen en daarmee reistijden kunnen worden afgeleid. Indien er geen reistijd meet systeem aanwezig is, kunnen reistijden ook worden afgeleid uit ander soort gegevens, bijvoorbeeld snelheden en intensiteiten. Deze techniek wordt *reistijd schatten* genoemd en is per definitie alleen *achteraf* mogelijk. Om een reistijdvoorspelsysteem te kunnen ontwikkelen is dus ofwel een reistijdmeetsysteem nodig, ofwel een meetsysteem op basis waarvan reistijden (achteraf) kunnen worden geschat.

Een typisch voorbeeld van zo'n laatste meetsysteem is het MONItoring CAsco (MONICA), dat geinstalleerd is op een groot gedeelte van het Nederlandse snelwegen netwerk. Aan de basis van MONICA staan zgn. inductie lussen, die om de 500-2000 meter per minuut gemiddelde snelheden en voertuigintensiteiten meten. Er bestaat een eenvoudige wiskundige relatie[3] tussen de harmonisch gemiddelde snelheid in tijdsperiode p op een bepaalde detectorlocatie x_d en de reistijd op een wegvak k rondom die

[2]Op een reis van 3 uur is een fout van 5 minuten te verwaarlozen. Op een trip van 10 minuten is dezelfde fout waarschijnlijk onacceptabel.

[3]$\tau = L_k/\widetilde{u}_{kp}$, waarbij L_k de lengte van wegvak k voorstelt en $\widetilde{u}_{kp}$ de harmonisch gemiddelde snelheid op een cross sectie op k gedurende p.

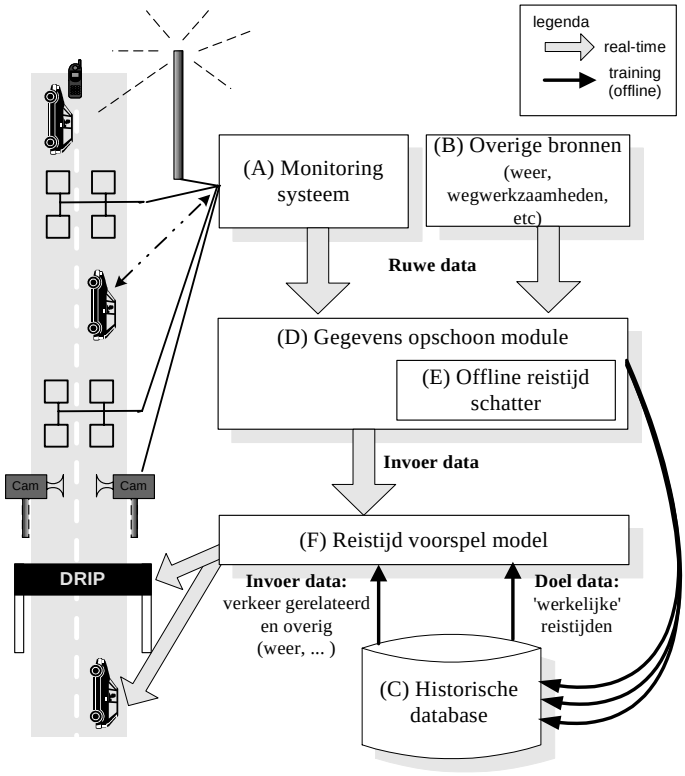


Figure E.2: Betrouwbaar raamwerk voor korte termijn reistijdvoorspelling op snelwegen.

detector, mits men een stationaire en homogene verkeerstoestand veronderstelt op k gedurende p. In MONICA wordt echter de rekenkundig gemiddelde snelheid berekend, waardoor structurele (significante) afwijkingen in reistijdschattingen op basis van die snelheden ontstaan. In dit proefschrift wordt een algoritme ontwikkeld om die structurele afwijking te compenseren, op basis van uitsluitend die gegevens die gemeten worden, namelijk tijdreeksen van rekenkundig gemiddelde snelheden en voertuigintensiteiten. De methode slaagt erin de afwijking nagenoeg volledig te compenseren. Met die gecorrigeerde snelheden kunnen vervolgens reistijden over routes van aaneengeschakelde wegvakken worden geschat. Daartoe introduceren we een nieuwe reistijdschatter gebaseerd op de in de praktijk veel gebruikte trajectoriën-methode. Met deze zogenaamde piece-wise linear speed based (PLSB) reistijdschatter[4] (figuur E.2 (E)) kan men vervolgens een grootschalige database (figuur E.2 (C)) opbouwen met historische reistijden per vertrektijdstip period p voor een bepaalde route r. Per periode p wordt behalve de gemiddelde reistijd (voor dan vertrekkende voertuigen) ook de metingen uit MONICA (en / of een vergelijkbaar systeem) van detectoren op route r opgeslagen.

Met die database is nu een reistijdvoorspelsysteem te calibreren en valideren. Terwijl een reistijd*schatter* in feite slechts verkeers gegevens achteraf "vertaalt" naar reistijden,

[4] stuksgewijs lineaire op snelheid gebaseerde reistijdschatter.

probeert een reistijd*voorspeller* op basis van huidige (en historische) verkeersgegevens de meest waarschijnlijke reistijd te berekenen voor voertuigen die nu of in de nabije toekomst de route r gaan afleggen. In niet-stationaire en congestieve verkeerscondities is dit een complex probleem. Immers, om de gemiddelde reistijd $\tau\ (p_0)$ van voertuigen op een route r vertrekkend in periode p_0 te kunnen berekenen, moeten we weten of deze voertuigen al dan niet vertraging gaan oplopen in de periode $\left[p_0,\ p_0 + \tau\ (p_0)\right]$, bijvoorbeeld ten gevolge van (het ontstaan van) file. Het probleem is dat de duur van die periode precies datgene is wat we wilden voorspellen, namelijk de reistijd $\tau\ (p_0)$. Modelleren en voorspellen van de dynamica van verkeerstromen is een complexe aangelegenheid, gestuurd door niet-lineaire en stochastische interactie van vele individuele voertuigen, elk bepaald door specifieke individuele eigenschappen als voertuigdimensies en -karakteristieken, rijstijl, aggressiviteit, reismotief, attentie niveau, etcetera. Het reistijdvoorspelprobleem is aan te pakken met ofwel gedetailleerde verkeersafwikkelingsmodellen ofwel generieke zgn. data-gestuurde modellen die de verbanden kunnen leren uit data.

We stellen een model voor uit de laatste categorie, een zgn. feedback neuraal netwerk, dat in staat is die verkeersdynamica in zowel de ruimte als tijd uit data te leren en daarmee reistijden te voorspellen. Het voordeel van een dergelijk data-gestuurd model is dat gegeven de wiskundige structuur, de parameters gemakkelijk kunnen worden aangepast mocht dit noodzakelijk zijn. Die wiskundige struktuur van dit zogenaamd state space neuraal netwerk (SSNN - figuur E.2 (F)) is echter ontleend aan verkeerstroom theorie. In de eerste plaats is het model geformuleerd als een state space model analoog aan veel verkeer afwikkelingsmodellen. Dit zorgt ervoor dat het SSNN reistijden voorspelt op basis van huidige metingen maar in de context van wat het in het verleden voorspelde. Daarnaast is het model gestructureerd op basis van de detector configuratie voorhanden op de route. Gegeven die configuratie is er geen invoerselectie meer nodig, bijvoorbeeld welke gegevens van welke detector van welke tijdstappen moeten worden gebruikt. Dit maakt het model geschikt om op een willekeurige snelweg (uitgerust met een meetsysteem) te worden toegepast. Wel lokatie specifiek is het resultaat van het calibratie proces, ook wel *training* genoemd. Uit dat proces volgen de specifieke route optimale parameters in het SSNN. Een belangrijk probleem met generieke geparameteriseerde modellen is dat gedurende calibratie de parameters prima kunnen fitten met de training (calibratie) data set (uit de eerder genoemde historische database), terwijl het model met de gevonden parameters op andere data heel slecht presteerd. Dit probleem (over fitting) is (onder andere) te voorkomen door gebruik te maken van Bayesiaanse regulering tijdens de training. Bij Bayesiaanse regulering wordt de *a posteriori* kans op een bepaalde parameterset gegeven de trainingsdata gemaximaliseerd, maar wordt tegelijkertijd gezorgd dat de parameters in het model zo klein mogelijk blijven. Dat laatste zorgt ervoor dat het SSNN een veel "gladdere" functie simuleert en veel beter generaliseert ook in geval van nieuwe (nog ongeziene) data. Bijvoorbeeld, het SSNN model dat is ontwikkeld voor de test case voor de zuidbaan van de A13 tussen Den Haag en Rotterdam bevat 735 parameters. Na Bayesiaanse gereguleerde training met MONICA-data en op MONICA gebaseerde

reistijdschattingen bleken daarvan slechts 20 tot 25% daadwerkelijk effectief; een reductie in modelcomplexiteit van bijna 80%. We introduceren daarbij ook een maat voor de relevantie van de verschillende neuronen en invoergegevens gebaseerd op het zogenaamde backpropagation trainingsalgoritme. Op basis van deze maat voor relevantie blijkt dat de gevonden (gereguleerde) parameterset nauw gerelateerd is aan de verkeersprocessen op de A13. De relevante parameters zijn afkomstig van detectoren op wegvakken waar de meeste vertraging ontstaat. Andere parameters "doen niets" of nauwelijks iets.

Behalve theoretisch en wetenschappelijk interessant, is het SSNN ook zeer geschikt voor toepassing als reistijdvoorspelmodel voor bijvoorbeeld DRIPs. De gemiddelde (structurele) fout over een grote testdataset (van de eerder genoemde A13) bestaande uit 118 namiddag-spitsperiodes (14:00-20:00) is minder dan 0.5% met een standaard deviatie van ca. 8%. In vergelijk met de huidige methode, een zogenaamd instantane reistijdvoorspeller[5] is dat structureel *twintig maal* zo nauwkeurig en neemt ook de residuele fout (de spreiding) af met 300%. Vergeleken met een aantal andere state-of-the-art modellen uit de literatuur, is het SSNN ongeveer even nauwkeurig als, of zelfs beter dan de als beste gemarkeerde modellen. Een belangrijk verschil is wel dat de resultaten van het SSNN behaald worden op soms sterk gecorrumpeerde (soms zelfs ontbrekende) data, terwijl in andere studies corrupte data vaak buiten beschouwing wordt gelaten.

In dit proefschrift wordt daartoe een aantal routines gepresenteerd om het op het SSNN gebaseerde systeem (figuur E.2) robuust te maken in geval van ontbrekende of gecorrumpeerde data. Speciaal in het geval van MONICA is dit van cruciaal belang. Ter illustratie: van de 27 detectoren gebruikt in het A13 model blijkt dat gemiddeld 12% van de gegevens uit MONICA onbetrouwbaar zijn of ontbreken met regelmatige uitschieters van 20 to 25%. In de zogenaamde "gegevens-opschoon-module" (figuur E.2(D)), is een aantal eenvoudige algoritmes geimplementeerd voor het online repareren van corrupte of ontbrekende gegevens. Deze eenvoudige algoritmes, gebaseerd op interpolatie en exponentieel smoothen, blijken in samenhang met het SSNN een robuust geheel te vormen. Zelfs bij percentages van 30 to 40% willekeurig ontbrekende gegevens is het totale systeem nog in staat nauwkeurige reistijdvoorspelling te produceren. Als detectoren (en zeker detectoren op locaties waar congestie ontstaat) structureel falen wordt die nauwkeurigheid sneller minder; nochthans produceert het SSNN model ook dan nog aannemelijke reistijden. Het blijkt ook mogelijk een SSNN te *leren* hoe om te gaan met datacorruptie. Ook dit vergroot de robuustheid spectaculair, zei het tegen de prijs van verminderde nauwkeurigheid.

Naast algoritmes om met ontbrekende gegevens om te gaan, stellen we ook methoden voor die expliciet de betrouwbaarheid van de voorspellingen berekenen en de onzek-

[5]Deze instantane voorspeller is feitelijk een reistijdschatter, die de veronderstelling maakt dat datgene wat nu gemeten wordt voor onbepaalde tijd constant blijft. Op basis van die constant gedachte gegevens worden dan reistijden geschat. In vrije afwikkeling (’s nachts) is dit een prima veronderstelling, in veranderlijke en dynamische condities (file vorming) loopt de veronderstelling spaak.

erheid daarin (statistisch) kwantificeren. Wij veronderstellen dat die onzekerheid is opgebouwd uit drie (onafhankelijke) componenten. De eerste heeft betrekking op de verdeling van reistijden in een gegeven vertrekperiode *p*, bijvoorbeeld omdat bestuurders verschillende rijstijlen en wenssnelheden hebben. De tweede heeft te maken met het feit dat we het SSNN getraind hebben met PLSB *geschatte* reistijden, in plaats van gemeten reistijden. Gegeven dat die PLSB-schatter (zowel structurele als residuele) fouten maakt, leren we derhalve het SSNN feitelijk om ook die fouten te reproduceren. Tenslotte is er onzekerheid in de parameters van het SSNN zelf. Omdat dit model in hoge mate niet-lineair is, is de parameter vector na training zeker geen unieke oplossing. In het Bayesiaanse paradigma levert de gevonden parametervector een maximum op in de performance van het SSNN als functie van die (hoog-dimensionele) parametervector, gegeven de training data en de specifieke structuur van het SSNN. Als men veronderstelt dat de parameters Gaussisch verdeeld zijn[6] kan een symmetrische onzekerheidsmarge om die piek worden gedacht die analytisch te berekenen is en een kwantitatieve maat voor de betrouwbaarheid van de uitvoer van het SSNN oplevert, als functie van de invoergegevens.

Met die betrouwbaarheidsmaat is het mogelijk om online de betrouwbaarheid in de SSNN voorspelling te kwantificeren door middel van zogenaamde betrouwbaarheidsintervallen. Dit is een krachtige middel, die de operator van een GRS dat gebruik maakt van een SSNN in staat stelt online de kwaliteit van de voorspellingen te monitoren, zonder ooit een werkelijke reistijd te meten (hetgeen toch alleen achteraf kan). Het blijkt dat de betrouwbaarheids intervallen breed worden - en daarmee de onzekerheid groot - in het geval dat

- er sprake is van ontbrekende of corrupte invoer
- het SSNN onbekend is met de invoer, bijvoorbeeld omdat er iets fundamenteels veranderd is op de route (snelheidslimieten, extra rijstroken, etc)
- in geval van extreme situaties, bijvoorbeeld ongelukken en incidenten

Met alle bronnen van onzekerheid tezamen - reistijdverdeling, PLSB-fouten en SSNN-parameters, kan men predictie-intervallen[7] construeren, die een beeld geven van de mogelijke spreiding van individuele reistijden rond de door het SSNN voorspelde waarde. In geval de voorspelling erg ver naast de werkelijke (gemiddelde) reistijd zit, zijn logischerwijs deze predictie-intervallen niet meer als onder- en bovengrenzen voor individuele reistijden te interpreteren.

[6] Deze (Gauss) veronderstelling is niet noodzakelijk, maar levert op dat de betrouwbaarheid analytisch kan worden berekend. In andere gevallen kan men door bijv. Monte-Carlo-simulatie de uitvoer verdeling benaderen.

[7] Predictie-intervallen omvatten noodzakelijkerwijs de eerder genoemde betrouwbaarheids intervallen

Tenslotte zijn er een aantal limitaties in het SSNN-systeem (figuur E.2) in de huidige vorm. Deze kunnen evenwel, met een aantal voor de hand liggende aanpassingen worden ondervangen. Bijvoorbeeld, het effect van weer of andere externe factoren op de reistijd kan expliciet worden meegenomen in het model. Ook zou men in het model expliciet verschillende dynamisch verkeersmanagement maatregelen (zoals dynamische snelheidslimieten) kunnen opnemen. Om dergelijke aanpassingen en verbeteringen goed te kunnen analyseren bevelen we aan om voor een aantal verschillende snelwegtrajecten grootschaliger reistijdmetingen te verrichten. Met die gemeten reistijden kan de PLSB-schatter verder worden geoptimaliseerd en kunnen de effecten van externe factoren beter in kaart worden gebracht. Tenslotte kan het SSNN-model ook worden toegepast op basis van andere verkeersdata-verzamelsystemen, zoals automatische voertuigidentificatie (AVI) systemen en floating car data (FCD), waarin voertuigen uitgerust met bijvoorbeeld GPS als voortbewegende detectoren fungeren.

De conclusie is dat het ontwikkelde reistijdvoorspelsysteem op basis van het SSNN, nauwkeurig, valide, adaptief en robuust is, en daarmee betrouwbaar. Het systeem biedt bovendien tools die deze betrouwbaarheid expliciet kwantificeren. Het systeem is generiek, in de zin dat het op een willekeurig snelwegtraject kan worden toegepast, gegeven (a) dat dit traject is uitgerust met een monitoring-systeem en (b) dat een historische database beschikbaar is van gemeten dan wel (achteraf) geschatte reistijden. Het centrale model in het systeem, het SSNN, is ook generiek, in de zin dat de wiskundige structuur gebaseerd is op de lay-out van het snelwegtraject en de aanwezige detectoren. Een model-ontwikkelaar hoeft zich niet bezig te houden met de tijdsdynamica, die intrinsiek in de structuur van het SSNN aanwezig is. In een aantal opzichten gedraagt het SSNN zich als een macroscopisch verkeer afwikkelingsmodel, met dat verschil dat het SSNN de complexe dynamica direct vanuit de data leert.

About the Author

Hans van Lint was born on April 15 1971 in Delft. After finishing a masters in civil engineering at the Delft University of Technology in 1997, with speciality civil engineering informatics, he started working for a small Delft-based software development company I.T. Works, specialized in tailor-made ICT solutions for consultants in the environmental engineering market. In a two year period, in which the company expanded from three to ten employees, his main activities were design, implementation and maintenance of a wide variety of software and database products, but also consultancy, group-communication techniques, acquisition and sales. Thereafter he joined the traffic management and telematics group of the traffic and transport department of consultant DHV Environmental & Infrastructure, Amersfoort (The Netherlands). Examples of projects in which he was involved include development of traffic information websites, evaluation of new traffic monitoring techniques, and project management of the Rotterdam regional traffic information centre (RegioTIC).

In May 2000 he continued part-time as technical project coordinator for RegioTIC. In that capacity he was co responsible for the day-to-day project management of the soft- and hardware developments connected to the RegioTIC, for example P+R dynamic routing system, RDS-TMC services, DATEX based traffic information management software and website, and more. The other half of the time he was affiliated as a researcher at the transport and planning department of the faculty of Civil Engineering and Geosciences of the Delft University of Technology, persuing a Ph.D. entitled "Reliable travel time prediction for freeways". From March 2001 until April 2004, Hans van Lint is affiliated full-time at the Delft University of Technology. Next to his Ph.D. research he was (and is) amongst other things involved in the HELENA project, in which a multilane/multiclass macroscopic traffic model is developed, and in the Regiolab Delft project. From May 2004 onwards he will remain at the transport and planning section in the capacity of assistent professor traffic flow theory.

TRAIL Thesis Series

A series of The Netherlands TRAIL Research School for theses on transport, infrastructure and logistics.

Nat, C.G.J.M., van der, *A Knowledge-based Concept Exploration Model for Submarine Design*, T99/1, March 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Westrenen, F.C., van, *The Maritime Pilot at Work: Evaluation and Use of a Time-to-boundary Model of Mental Workload in Human-machine Systems*, T99/2, May 1999, TRAIL Thesis Series, Eburon, The Netherlands

Veenstra, A.W., *Quantitative Analysis of Shipping Markets*, T99/3, April 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Minderhoud, M.M., *Supported Driving: Impacts on Motorway Traffic Flow*, T99/4, July 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoogendoorn, S.P., *Multiclass Continuum Modelling of Multilane Traffic Flow*, T99/5, September 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoedemaeker, M., *Driving with Intelligent Vehicles: Driving Behaviour with Adaptive Cruise Control and the Acceptance by Individual Drivers*, T99/6, November 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Marchau, V.A.W.J., *Technology Assessment of Automated Vehicle Guidance - Prospects for Automated Driving Implementation*, T2000/1, January 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Subiono, *On Classes of Min-max-plus Systems and their Applications*, T2000/2, June 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Meer, J.R., van, *Operational Control of Internal Transport*, T2000/5, September 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Bliemer, M.C.J., *Analytical Dynamic Traffic Assignment with Interacting User-Classes: Theoretical Advances and Applications using a Variational Inequality Approach*, T2001/1, January 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Muilerman, G.J., *Time-based logistics: An analysis of the relevance, causes and impacts*, T2001/2, April 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Roodbergen, K.J., *Layout and Routing Methods for Warehouses*, T2001/3, May 2001, TRAIL Thesis Series, The Netherlands

Willems, J.K.C.A.S., *Bundeling van infrastructuur, theoretische en praktische waarde van een ruimtelijk inrichtingsconcept,* T2001/4, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Binsbergen, A.J., van, J.G.S.N. Visser, *Innovation Steps towards Efficient Goods Distribution Systems for Urban Areas,* T2001/5, May 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Rosmuller, N., *Safety analysis of Transport Corridors*, T2001/6, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Schaafsma, A., *Dynamisch Railverkeersmanagement, besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*, T2001/7, October 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Bockstael-Blok, W., *Chains and Networks in Multimodal Passenger Transport. Exploring a design approach*, T2001/8, December 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, T2002/1, February 2002, TRAIL Thesis Series, The Netherlands

Vis, F.A., *Planning and Control Concepts for Material Handling Systems*, T2002/2, May 2002, TRAIL Thesis Series, The Netherlands

Koppius, O.R., *Information Architecture and Electronic Market Performance*, T2002/3, May 2002, TRAIL Thesis Series, The Netherlands

Veeneman, W.W., *Mind the Gap; Bridging Theories and Practice for the Organisation of Metropolitan Public Transport*, T2002/4, June 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Van Nes, R., *Design of multimodal transport networks, a hierarchical approach*, T2002/5, September 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Pol, P.M.J., *A Renaissance of Stations, Railways and Cities, Economic Effects, Development Strategies and Organisational Issues of European High-Speed-Train Stations,* T2002/6, October 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Runhaar, H., *Freight transport: at any price? Effects of transport costs on book and newspaper supply chains in the Netherlands*, T2002/7, December 2002, TRAIL Thesis Series, Delft University Press, The Netherlands

Spek, S.C., van der, Connectors. *The Way beyond Transferring,* T2003/1, February 2003, TRAIL Thesis Series, Delft University Press, The Netherlands

Lindeijer, D.G., *Controlling Automated Traffic Agents*, T2003/2, February 2003, TRAIL Thesis Series, Eburon, The Netherlands

Riet, O.A.W.T., van de, *Policy Analysis in Multi-Actor Policy Settings. Navigating Between Negotiated Nonsense and Useless Knowledge*, T2003/3, March 2003, TRAIL Thesis Series, Eburon, The Netherlands

Reeven, P.A., van, *Competition in Scheduled Transport*, T2003/4, April 2003, TRAIL Thesis Series, Eburon, The Netherlands

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, T2003/5, June 2003, TRAIL Thesis Series, The Netherlands

Soto Y Koelemeijer, G., *On the behaviour of classes of min-max-plus systems*, T2003/6, September 2003, TRAIL Thesis Series, The Netherlands

Lindveld, Ch..D.R., *Dynamic O-D matrix estimation: a behavioural approach*, T2003/7, September 2003, TRAIL Thesis Series, Eburon, The Netherlands

Weerdt, de M.M., *Plan Merging in Multi-Agent Systems*, T2003/8, December 2003, TRAIL Thesis Series, The Netherlands

Langen, de P.W, *The Performance of Seaport Clusters*, T2004/1, January 2004, TRAIL Thesis Series, The Netherlands

Hegyi, A., *Model Predictive Control for Integrating Traffic Control Measures*, T2004/2, February 2004, TRAIL Thesis Series, The Netherlands

Lint, van, J.W.C., *Reliable Travel Time Prediction for Freeways*, T2004/3, June 2004, TRAIL Thesis Series, The Netherlands