COOPERS

Co-operative Networks for Intelligent Road Safety

WP 3800

IR 3800

Report on new functions, requirements and algorithm for direct control and decision support

D. Grether, A. Bergsten, D. Zetterberg, Y. Chen, D. Strippgen, K. Nagel

Version
1.3

Status	Date
Draft	21.04.2008
Reviewed	
Approved	

Release	Function	Approval
K. Nagel, D. Strippgen	Name	
TU Berlin, VSP	Organisation	
+49 30 314-29552	Phone	
+49 30 314-26269	Fax	
Strippgen@vsp.tu-berlin.de	E-Mail	

Contract Number:

FP6-2004-IST-4 Nr. 026814

Acronym:

COOPERS

Title:

Co-operative Networks for Intelligent Road Safety

Distribution:

Part.-Nr.	short name	Participant name	Nationality
1	ATE	AustriaTech – Gesellschaft des Bundes für technologiepolitische Maßnahmen	Austria
2	HIT	Vereinigung High Tech Marketing	Austria
3	SEI	ARC Austrian Research Centers GmbH	Austria
4	ARS	ARS Traffic and Transport Technology B.V.	Netherlands
5	SWA	Swarco Europe GmbH	Austria
6	EYI	Ernst and Young Financial – Business Advisors S.p.A.	Italy
7	ASF	ASFINAG - Autobahnen- und Schnellstraßen-Finanzierungs- Aktiengesellschaft	Austria
8	FAV	Forschungs- und Anwendungsverbund Verkehrssystemtechnik Berlin	Germany
9	ORF	Österreichischer Rundfunk	Austria
10	-	Left intentionally blank	
11	TUW	Technische Universität Wien	Austria
12	ASC	Ascom Switzerland Ltd	Switzerland
13	TRG	University of Southampton	United Kingdom
14	PWP	pwp-systems GmbH	Germany
15	OBB	Oberste Baubehörde im Bayerischen Staatsministerium des Innern	Germany
16	DOR	Dornier Consulting GmbH	Germany
17	GEW	GEWI Hard- und Software Entwicklungsgesellschaft mbH	Germany
18	BRE	Autostrada del Brennero	Italy
19	VEG	VEGA Informations-Technologien GmbH	Germany
20	-	Left intentionally blank	
21	LOD	Politechnika Lodzka	Poland
22	LUC	Lucent Technologies Network Systems GmbH	Germany
23	TRA	TRANSVER GmbH	Germany
24	FHG	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung E.V.	Germany
25	EFM	EFKON mobility GmbH	Germany
26	EFK	EFKON AG	Austria
27	VTI	Statens väg- och transportforsknings-institutet	Sweden
28	KTH	Kungliga Tekniska Högskolan	Sweden
29	NET	TeamNet International S.A.	Romania
30	STH	S&T Hermes Plus	Slovenia
31	INO	INESC Inovação – Instituto de Novas Tecnologias	Portugal
32	APP	LGAI Technological Center S.A.	Spain
33	ICI	National Institute for Research Development in Informatics	Romania
34	TUC	Technical University of Crete	Greece
35	KYB	Kybertec, s.r.o.	Czech Republic
36	JAS	JAST Sàrl	Switzerland
37	PHI	Philips Innovative Technology Solutions NV	Belgium
38	BMW	Bayerische Motoren Werke Aktiengesellschaft	Germany
39	NAV	Navteq B.V.	Netherlands
40	-	tbd1	
41	-	TBD2	
42	ARC	Austrian Research Centers GmbH	Austria
43	ASA	ASFA – Association professionelle des Sociétés Françaises concessionnaires ou exploitantes d'Autoroutes ou d'ouvrages routiers	France

Document History:

Version	Date	Released by	Description
0.5	11.04.2008	D. Strippgen	VSP, TU Berlin, Report
0.9	27.04.2008	D.Strippgen	Introduction revised
1.0	27.04.2008	D. Grether	Reworked scoring chapter
1.1	02.05.2008	K. Nagel	Revised version
1.2	02.05.2008	E. Kosmatopoulos	Revised and rewritten Chapter 2.6
1.3	16.06.2008	K. Nagel	Changes from peer-review

Table of Contents

Executive Summary **15**

1 Introduction **17**

1.1 Related work from SWP 4200 17

1.2 Aim and hypotheses 19

1.3 Method 19

1.4 Structure of this report 19

2 Background **21**

2.1 Route guidance theory 21

2.2 Multi-Agent Transport Simulation (MATSim) as a telematics test bed 28

3 The performance of different prediction models for travel times and traffic jams **36**

3.1 The basic theory used in the models 36

3.2 Evaluation measures 37

3.3 The Single Bottleneck Model 38

3.4 The Distribution Model 45

3.5 Disturbance Model 51

3.6 Incident detection 56

3.7 Multi Bottleneck Model 61

4 Testing the approach in a real world scenario: The Berlin simulations **66**

4.1 Merging the models 66

4.2 Scoring 66

4.3 Subpopulation score 66

4.4 Evaluation of Travel Times 67

4.5 Economic evaluation 82

4.6 Equipment ratio 83

5 Discussion **85**

6 Conclusions **86**

7 References **87**

List of Figures

Figure 1. Feedback flowchart of classic feedback control 22

Figure 2. Illustration of the system time lag. 23

Figure 3. Nash times with reactive control. 24

Figure 4. Feedback flowchart with prediction. The dotted arrow is the measurements from the system, i.e. number of agents on the links, etc. 25

Figure 5. Diagram for multiple route guiding. 26

Figure 6. The MATSim simulation process. 29

Figure 7. *The* main route (red) and the alternative route (blue) of the Tegel test location. (Google Maps) 31

Figure 8. The test network. Green colour represents free flow conditions and red represents congestion.The accident location is indicated by the arrow. 33

Figure 9, An UML–representation of the controller and prediction classes. The UML standard for illustrating class structures can be found for example on the web page: http://www.agilemodeling.com/artifacts/classDiagram.htm (29 Jan 2008). 35

Figure 10. The test network used for the primary evaluation round. 40

Figure 11: Comparison of the predictions of the Single Bottleneck Model and the actual travel times. No guidance control is applied. 41

Figure 12: Nash times Comparison between reactive control and Single Bottleneck control 43

Figure 13. Travel times for the main route without an accident. It can be seen that not all agents travel with free speed even though there are capacity left. Maximum capacity of a route is 3000 agents/hour and in this run, there are 2000 agents/hour. 44

Figure 14. Sparse traffic on the part before the accident. The arrow indicates the bottleneck. 45

Figure 15. The two steps in the algorithm of the Distribution model. 46

Figure 16: Main route predictions with basic SB-control of pulsating traffic. 47

Figure 17: Alternative route predictions with basic SB-control of pulsating traffic. 47

Figure 18: Alternative route predictions using Distribution Model on pulsating traffic. 48

Figure 19: Main route predictions using Distribution Model on pulsating traffic. 49

Figure 20: Nash time plot for the distribution model on a scenario with pulsating traffic. 51

Figure 21: Measured and SB-predicted travel times on the main route when there is an additional inflow. 54

Figure 22: Measured and Disturbance-predicted travel times on the main route when there is an additional inflow. 55

Figure 23: Nash times for the disturbance and the regular single bottleneck model, respectively. 56

Figure 24: Travel times on main route with automatic incident detection. 59

Figure 25: Scenario for developing the Multi Bottleneck Model. Accidents are reducing the capacity on link 14 to 77 percent and on link 11 to 63 percent. Test population: PlansInflow (as PlansNormal but with additional inflow on link 31 of 500 agents per hour). 61

Figure 26: The regular SB with incident detection on a scenario with two queues. 63

Figure 27: SB, incident detection and Disturbance compensation on a scenario with two queues. 64

Figure 28: Multi Bottleneck model on a scenario with two queues. 65

Figure 29. The no accident case (left) and the accident case (right) at 17:20. The accident causes a long queue on the highway, but the on the alternative route the demand is low. 68

Figure 30. Travel times in the accident – no control scenario. Purple graphs are observations; black are values predicted by the Single Bottleneck Model. There are two outliers in the upper diagram. 69

Figure 31. Route guidance based on observed travel times for the main route (above) and the alternative route (below). Purple represents the real value; black represents value used in feedback. 70

Figure 32. The Basic Single Bottleneck model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback. 71

Figure 33. System at 17:20. Left: no control; right: control with Basic SB model. Compared to the open loop system, the queue on the highway is significantly reduced and the demand on the alternative route is considerably higher. 72

Figure 34. Traffic situation at 16:20 (just before the accident occurs). No control (left) and Basic SB Control (right). 73

Figure 35. Distribution model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback. 74

Figure 36. Disturbance model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback. 75

Figure 37. Prediction of the Basic SB model when the highway has normal capacity and is not subject to route guidance. Purple represents the real travel times; black represents predicted values. 76

Figure 38. Incident detection. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback. 77

Figure 39. SB model with distribution, disturbance and incident detection functionality active. Purple represents the real travel times; black represents predicted value applied in feedback. 78

Figure 40. Observed Nash times for the accident scenario with the SB model with all features. The Nash times are centered around –100, which corresponds to the prediction error on the main route, subtracted with the prediction error on the alternative route. 79

Figure 41. Multi Bottleneck Model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback. 80

Abbreviations

3G	3rd Generation Networks
4G	4th Generation Networks
AADT	Annual Average Daily Traffic
AASHTO	American association of state and highway transportation officials
AB	Abbiege-Unfall
ABS	Antilock Braking Systems
ACC	Automated Cruise Control
ADAS	Advanced Driver Assistance Systems
AETR	European Agreement Concerning the Work of Crews of Vehicles Engaged in International Road Transport
AG	Aktiengesellschaft
AIDE	adaptive integrated driver-vehicle interface
AIDER	Innovative Vehicle-Infrastructure Telematics for Rescue Operations
AIS	Abbreviated Injury Scale
AIS/ISS	Abbreviated Injury Scale/Injury Severity Scores
AKTIV	Adaptive und Kooperative technologien für den Intelligenten Verkehr
AP	Action Point
API	application programming interface
ART	Article
ARTS	Advanced Road Telematics in the South-West
ASRB	automotive safety restraints bus
ASV	Advanced Safety Vehicle
AUTOSAR	Automotive Open System Architecture
AVCSS	Advanced Vehicle Control and Safety Systems
AVI	Automatic Vehicle Identification
BASt	Bundesanstalt für Straßenwesen
BMVBW	Bundesministerium für Verkehr, Bau- und Wohnungswesen
BMVIT	Bundesministerium für Verkehr, Innovation und Technologie
BS	British Standard
BSI	Bundesamt für Sicherheit in der Informationstechnologie
CA	Collision Avoidance
CA	Consortium Agreement
CALM	Communication Air-interface Long and Medium range
CALM M5	Continuous Air interfaces - Long and Medium Range - Microwave 5 GHZ
CAN	CAN-Bus (Controller Area Network)
CARE	Community database on Accidents on the Roads in Europe
CAREPLUS	Citizens Active in Reading Education Plus
CARTALK-2000	Safe and comfortable driving based upon inter-vehicle communication
CCTV	Closed Circuit Television
CDC	Collision Deformation Classification
CDMA	Code division multiple access
CDRG	Centrally Determined Route Guidance
CE	Communauté Européenne
CEN	European Committee for Standardisation
CENELEC	European Committee for Electro technical Standardisation
CENTRICO	Central European region transport telematics implementation co-ordination
CHAMEL-EON	Pre-crash application all around the vehicle
CICAS	Cooperative Intersection Collision Avoidance Systems

CO	Coordinator
COOPERS	Co-operative systems for Intelligent Road Safety
CORBA	Common Object Request Broker Architecture
CORVETTE	Co-ordination and validation of the deployment of advanced transport telematic systems in the Alpine area
COST	Coopération européenne dans le domaine de la recherche scientifique et technique
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CS	Cost statement
CSMA/CA	Carrier Sense Multiple Access - Collision Avoidance
CT	Communication Tool
CVIS	Cooperative Vehicle-Infrastructure Systems
CW	Collision Warning
D	Deliverables
D&E	Dissemination and exploitation
DAB	Digital Audio Broadcasting
DAB/ DVB	Digital audio broadcasting/ digital video broadcasting
DARC	Data Radio Channel
DART	Dutch Accident Research Team
DATEX	Data exchange
DC	Dissemination committee
DG	Direction General
DG INFSO	Directorate General Information Society and Media
dGPS	Differential Global Positioning System
DIS	Driver Information System
DML	Demonstration management leader
DOT	departments of transportation
DSRC	Dedicated Short-Range Communication
DVB	Digital video broadcasting
DVR	Deutscher Verkehrssicherheitrat
e.g.	for example
E/E/PE	Electrical/Electronic/Programmable Electronic
EASIS	Electronic architecture and system engineering for integrated safety systems
EC	European Commission
eCall	emergency Call
ECE	Economic Commission for Europe
ECU	electronic control unit
EDIFACE	Electonic Data Interface
EEA	European Environment Agency
EEC	European Economic Community
EEG	Electroencephalogram
EES	Equivalent energy speed
EFC	Electronic Fee Collection
EFCD	Enhanced floating car data
EFTA	European Free Trade Association
EK	Einbiegen-/Kreuzen-Unfall
EMC	Electromagnetic Compatibility
E-MERGE	European Mountain lake Ecosystems: Regionalisation, diaGnostics and socio-economic Evaluation
EMI	Electromagnetic Interference
EN	European Norm

ERI	Electronic Registration Identification
ESA	European Space Agency
ESC	Electronic Stability Control
ESP	Electronic stability program
ESS	Environmental Sensor Stations
ETA	Estimated time of arrival
ETSI	European Telecommunications Standards Institute
EU	European Union
EUC	Equipment Under Control
EVTA	Event tree analysis
EWG	Environmental Working Group
FCD	Floating Car Data
FM	Frequency Modulation
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Modes, Effects, and Criticality Analysis
FMSCA	Federal Motor Carrier Safety Administration
FP	Framework Programme
FRAME	Framework architecture made for Europe
FStrAbG	Fernstraßenausbaugesetz
FSV	Forschungsgemeinschaft Strasse und Verkehr
FTA	Fault Tree Analysis
FTDMA	Flexible Time Division Multiple Access - bandwidth partitioning by time slicing
G	generation
GDF	Geographic Data Files
GHR	Gazis-Herman-Rothery
GIS	Geographical information system
GM	General Motors
GNSS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global system for mobile communications
GSSF	Galileo system simulation facility
GST	Global System for Telematics
HANNIBAL	High Altitude Network for the Needs of Integrated Border-Crossing Applications and Links
HAZOP	Hazard and Operability analysis
HGV	Heavy Goods Vehicles
HL	high level
HMI	Human Machine Interface
HOV	High Occupancy Vehicle
HUDs	Head-Up Displays
HW+SW	Hardware + Software
I	Infrastructure
I/O	input/output
I2V	Infrastructure to Vehicle
ICD	International Classification of Diseases
ICT	Information and Communications Technology
ICTSB	ICT Standard Board
ID	Identification
IEC	International Electronical Commission
IEEE	Institute of Electrical and Electronics Engineers

INS	Institut national de Statistique
INVENT	Infrastructure for Virtual Enterprises
IOG	Infrastructure operator group
IP	Integrated project
IR	Internal report
IRTAD	International Road Traffic and Accident Database
ISA	Intelligent Speed Adaptation
ISDN	Integrated services digital network
ISO	International Organization for Standardization
ISP	Industry and Component Suppliers Panel
IST	Information society technologies
ISTAT	Istituto Centrale di Statistica
IT	Information Technology
ITS	Intelligent Transport Systems
ITSSG	Intelligent Transport Systems Steering Group
ITU-T	International telecommunication Union – Terminals for telematic services
IVHW	Inter Vehicle Hazard Warning
IVI	Intelligent vehicle infrastructures
J2EE	Java 2 platform enterprise edition
J2SE	Java 2 platform standard edition
JK	Jahreskarte
KAREN	Keystone architecture required for European networks
KD	Unfallkostendichte
KFV	Kuratorium für Verkehrssicherheit
KL	Unfallsbelastungskosten
KPI	Key performance indicators
KR	Unfallkostenrate
kW	kiloWatt
LACOS	Large Scale Correct Systems
LAN	Local area network
LATERAL-SAFE	Lateral driver assistance applications
LCS	Line Control Systems
LDRG	Locally Determined Route Guidance
LDW/A	Lane Departure Warning/Avoidance
LDWS	Lane Departure Warning Systems
LED	Light Emitting Diode
LIN	Local interconnect network
LOS	Level of Service
LV	Unfall durch Längsverkehr
LVD	Low Voltage Directive
M	Milestone
MALSO	Manoeuvring Aids for Low Speed Operation
MATSim	Multi-Agent Transoprt Simulation (Toolkit)
MOST- Bus	Media oriented systems transport bus
MOT	Multimedia object transfer protocol
MT	Management team
MTM	Methods Time Measurement
NGOs	Non-Governmental Organizations
OBU	Onboard Unit
OEM	Original Equipment Manufacturer/Manufacturing

OSGi	open services gateway initiative
PAC	Policy advisory panel
PAD	Portable Application Description
PATH	Program for Advanced Transit and Highway
PC	Project Coordinator
PCI	peripheral component interconnection
PCMCIA	personal computer memory card international association
PDA	Personal digital assistant
PDAC	plan-do-act-control
PDT	Peripheral Detection Task
PM	Person months
PMT	Project Management Team
PPP	Public private partnership
PReVENT	Preventive and Active Safety Applications
PROSPER	Project for Research on Speed adaptation Policies on European Roads
PSAPs	Public Safety Answering Points
PT	public transport
PTPS	Public Transportation Priority System
R	Reports
R&D	Research & development
RACM	Reasonably available control measures
RAMSS	Reliability, Availability, Maintainability, Safety & Security
RDCW	Road Departure Crash Warning
RDS	Radio Data Systems
RDS-TMC	Radio Data System - Traffic Message Channel
RFID	Radio Frequency Identification Device
RMI	Road monitoring infrastructure
RM-ODP	Reference Model – Open Distributed Processing
RPN	Risk Priority Number
RPU	Robust Positioning Unit
RRS	Road Restraint Systems
RSE	roadside equipment
RSU	roadside unit
RTA	Road Traffic Advisor
RTD	Round trip delay
RTLX	Raw Task Load indeX
RTTT	Road Transport and Traffic Telematics
RV	Unfall durch ruhenden Verkehr
RX	Receiver
SA	System architecture
SAE	System Architecture Evolution
SAE	Society of Automotive Engineers
SAFELANE	Situation Adaptive system For Enhanced LANE keeping support
SafeSpot	Cooperative vehicles and road infrastructure for road safety
SARTRE	Social Attitudes to Road Traffic Risk in Europe
SBAS	Satellite Based Augmentation System
SCB	Statistics Sweden
SCOM	Steering committee
SERTI	Southern European Road Telematic Implementations
SIG	Special Interest Group

SIKA	Statens Institut för KommunikationsAnalys
SIL	Safety Integrity Level
SMS	Short message service
SNRA	Swedish National Road Administration
SO	Sonstiger Unfall
SRA	Swedish Road Administration
SRB	Safety research board
STRADA	Swedish Traffic Accident Data Acquisition
STREETWISE	Seamless Travel EnvironmEnt for the Western Isles of Europe
STVO	Straßenverkehrsordnung
StVUnfStatG	Straßenverkehrsunfallstatistikgesetz
SVD	Selective Vehicle Detection
SWOV	Stichting Wetenschappelijk Onderzoek Verkeersveiligheid
SWP	Sub Work Package
SWPL	Sub-work package leaders
TCC	Traffic Control Centres
TCT	Technical co - ordination team
TDMA	Time Division Multiple Access - bandwidth partitioning by time slicing
TEN	Trans European network
TEN-MIP	Trans European network-multi annual programme
TEU	Traffic eye universal
TIC	Traffic Information Centre
TICS	Traffic Information and Control Systems
TISP	Traffic information service provider
TIWS	Traffic Impediment Warning Systems
TLT	Thematic leader team
TMC	Traffic Message Channel
TMIC	Traffic management and information centres
TMT	Thematic leader teams
TNO	Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek
TPEG	Transport Protocol Experts Group
TRMM	Trunk Road Maintenance Manual
TTI	Tactical traffic image
TTI	Traffic and Traveller Information
TTP(/C)	Time-Triggered Protocol (/ Dependability Level C)
TX	Transmitter
U	Unfälle
UD	Unfalldichte
UDP	user datagram protocol
UL	Unfallbelastung
UML	Unified modelling language
UMTRI	University of Michigan Transportation Research Institute
UMTS	Universal mobile telecommunications system
UR	Unfallrate
US	United States
ÜS	Überschreiten-Unfall
USDOT	United States department of transportation
UTMS	Universal Traffic Management Society
V	Vehicle
V2I	vehicle to infrastructure

V2V	Vehicle to Vehicle
VAS	Value added service
VEESA	vehicle e-safety architecture
VII	vehicle infrastructure integration
VIKING	Co-ordination of ITS implementation in northern Europe
VDS	Variable Direction Sign
VMS	Variable Message Sign
VMT	Vehicle mile traveled
VRUs	Vulnerable Road Users
VSL	Variable Speed Limit
VTPI	Voorhees Transportation Policy Institute
VTTI	Virginia Tech Transportation Institute
WBS	Work breakdown structure
WBT	Web based training
WILLWARN	Wireless Local Danger Warning
WLAN	Wireless local area network
WP	Work Package
WPL	Work Package Leader
WüStV	Wiener Übereinkunft über den Straßenverkehr
XFCD	Extended Floating Car Data
XGDF	eXtended Geographic Data Files
XML	eXtensible Markup Language
ZIP	Zone Improvement Plan

Executive Summary

This report is an addition to the results of our first report “**IR 4200-2: Strategies and requirements for automated traffic management**“ to be found on the COOPERS portal in the corresponding working folder of WP 4200. The outcome of the WP4200 report was that, although controlling a traffic scenario, as given on the Berlin motor highway, is feasible even without an underlying model of the traffic, this turned out to be rather cumbersome. Numerous parameters had to be fitted for every de-tour, which emerged as a rather time consuming process and leads to a high degree of mutually dependent variables, all to be maintained correctly to give satisfying results.

We therefore investigated a more sophisticated model, to better adopt the traffic behaviour. This report summarizes our effort to research strategies for controlling traffic with an underlying traffic model. Again the Berlin demonstrator was used as an example to put our theories to work.

Traffic congestion constitutes a problem in many large cities. Congestion can be handled by reducing the network demand, expanding the infrastructure, or by utilizing the road network more efficiently. This report presents a methodology for route guidance, based on automatic feedback control from the current traffic situation. Through variable direction signs or individual in-car (coopers) devices, all vehicles with a certain origin and destination (both normally intermediate) are guided to take the currently fastest route. In this report the traffic is guided over one of two alternative routes with the goal of a Nash equilibrium, i.e. that every guided agent travels the fastest path. Nash equilibrium occurs when the routes have equal travel times, or when all agents use a route with shorter travel time.

Predictive data about how the system reacts to the control measures is fundamental to control the traffic in an optimal way. Feedback of observed travel times results in an unstable system with oscillating travel times. Given this background, the task of this report has been to present a model that predicts route travel times, with the purpose of improving the performance of traffic route guidance. The approach used is automatic feedback control, and therefore some basic terminology from control theory is used throughout the article.

The model introduced in this report needs no parameter estimation but uses only static information about traffic network, together with on-line counting of vehicles. It is shown that with reliable travel time predictions, optimal control can be achieved by bang-bang control, which also needs no parameter estimation. This results in a guidance system that works on any location without prior estimation of location specific parameters.

A microscopic traffic simulator, MATSim (Multi-Agent Transport Simulation), is used for developing the prediction model and evaluating its system effect with route guidance. Simulated in-car COOPERS (Co-operative Systems for Intelligent Road Safety) devices are used for data collection and for transmitting the guidance to vehicles. The advantage of a microscopic simulator is that it is able to correctly handle situations with inhomogeneous driver populations, caused for example by drivers/vehicles with many different destinations.

The prediction models and the feedback control are evaluated in two different traffic networks: a topologically simple test network and a reduced version of the full Berlin network. The results of the simulations are promising; guidance with predictive models results in shorter average travel time than guidance based on observed travel times. Evaluation with the *score measure* indicates that drivers benefit economically from predictive route guidance.

1 Introduction

The area of traffic guidance is currently developing fast. The background is both increasing problems with traffic congestions in major cities and introduction of new traffic management technology such as GPS devices. Congestions are naturally due to several causes. If the demand on the road network is much bigger than its capacity, the situation is clearly hard and cannot be solved without physical, and generally expensive, extension of the infrastructure. But this is not always the case. Often, only some of the streets are congested at a given time while the demand on other streets are far beneath the maximum capacity. In such cases, it can be fruitful to guide vehicles into less crowded streets in order to use the available network optimally.

It is a common assumption that people take the best route in a normal traffic situation. But when the traffic fluctuates due to incidents and unexpected demands, people often take the slower alternative.

How should this guiding be carried out? There are many ways of estimating the traffic situation. Public radio information about traffic is often based on phone calls, cameras or observing helicopters. Route guidance can also be given through variable road signs with direction advices or directly to in-car GPS devices. Measurements of travel times from point A to point B via two alternative routes can be used to guide vehicles at A to take the faster alternative. The theory applied is classic feedback control that is commonly used in many industrial processes to achieve a certain predefined control goal. The control goal in our case is to direct vehicles into the currently fastest route.

This approach has been tested with good results; see for example Diakaki et al. (1997). However, measuring the travel times at B generally sends a control signal that is too late, since there is a time lag between A and B. What is really wanted, is a reliable prognosis of the time it will take a vehicle to travel to B, starting from A in this moment – not the travel time for the latest measured vehicle that reached B. These adequate travel times must be predicted. The idea of this report is to calculate the travel times based on the present traffic situation. The development and testing of the prediction model were carried out in the traffic simulating software MATSim.

1.1 Related work from SWP 4200

In our last report IR 4200-2 we summarized strategies for automated traffic management in the context of the COOPERS project. We started out with the list of services that are envisaged for the COOPERS device. Those services were then partitioned into those that will be feasible with the first generation device (i.e. without explicit back channel of the COOPERS device to the infrastructure), and those that will be not. For some of those services, in particular travel time prediction and route guidance, it turned out that the back channel can be replaced by other channels of information flow from the traffic system to the traffic management centre, such as, e.g., traffic state estimation from conventional induction loops. To consider a specific set-up, the SWP 4200 report as well as this report is oriented towards the Demostrator 3 (Berlin).

We then discussed evaluation technologies for telematics applications (of which COOPERS is an example). In particular, we pointed out that traditional transportation planning software, most notably the four-step process with equilibrium assignment, is not able to evaluate telematics projects since the very advantage of telematics approaches, to react in real time to fluctuations and incidents, is not picked up by that software. In addition, many existing (software) tools tend to aggregate the driver population into macroscopic flow rates. Such approaches have their advantages especially for control theory applications, since they are mathematically tractable. However, when a system with a large diversity of drivers, e.g. with many different destinations, is considered, such systems reach their limits. Instead, we tried a microscopic (sometimes called agent-based) approach – such approaches have no problems with diversity in the system, but are in general not mathematically tractable.

In Report IR 4200-2 we discussed specific examples of automated traffic management that are possible with the COOPERS setup. In particular, simplified feedback control for route guidance is investigated in the context of the microscopic simulation. Such approaches have in fact been investigated previously with underlying macroscopic (or so-called mesoscopic) models, but it was important to show that control strategies can also be defined and investigated for the microscopic models.

A shortcoming of the SWP4200 Report algorithm and models was, that fine-tuning the simple control mechanisms could get rather tedious. With several on- and off-ramps to the highway, as given especially on the A100/A110 city highway we had to cope with in the Berlin demonstrator, this could lead to a huge amount of tuning to be done. In this report we will try to find ways of improving the technology, so that manual parameter fitting might be no longer needed. This would hopefully increase the feasibility to implement our algorithm into real-world systems.

Apart from improving the technology part of the implementation of a COOPERS device into our MATSim framework, we use the integrated capability of MATSim to calculate a score, which is the overall performance an agent has experienced during his day. This score is a measure of contentment of the agent or satisfaction that the agent has aggregated over the time of the day. We improved the MATSim scoring algorithm with respect to the ability to filter out the scores of sub-populations, e.g. those parts of the Berlin population equipped with a COOPERS device. The score is measured in Euro, and meant to be comparable with the monetarized output from standard economic appraisal. Inside MATSim, however, the score does not only react to plain travel time gains, but ultimately will allow us to differentiate between, say, a traveller who is in a hurry and another one who has time. This will ultimately allow to perform an economic evaluation of telematics devices in direct comparison to more conventional measures, such as capacity expansion.

So with respect to our former report IR 4200-2 our ongoing research was twofold:

- Implementing new, more sophisticated ways of automatic travel control, essentially to avoid the costly parameter tweaking of the context-free algorithms that we implemented in our former report

- Integrating the powerful scoring mechanism build into MATSim with the automatic control mechanisms simulation runs, to be able to obtain meaningful economic scores from both the whole population as well as relevant subpopulations of the scenarios rendered.

1.2 Aim and hypotheses

The overall aim of this reports work is to develop travel time prediction models that improve the feedback control compared to using measured data. The models will be implemented and evaluated in MATSim. A general approach will be taken when developing the models so that they work independently from the guidance location, meaning that no parameter tuning will be necessary.

In focus are accident scenarios. When there is a queue under normal conditions – for example in the morning peak hours – the road network is normally used to its full capacity and no better alternatives can be found. Hence there is no potential for guidance control. But when there is a temporary capacity reduction caused by an accident or road maintenance work, the traffic can be rerouted on an alternative route that still has a capacity buffer.

Throughout this report we will use an onboard COOPERS-device to give detour recommendation to the driver. Therefore our system could be used for a wide range of possible COOPERS services. It could be clearly be utilised for these services, planned in the COOPERS context:

- S1a: Accident warning
- S1b: Incident warning
- S6: Traffic congestion warning
- S11: Recommended next link

Furthermore it could be used to be able to calculate the estimated journey time (S10) in a more meaningful way.

1.3 Method

Our approaches for predicting travel times are based on physical principles of traffic flows. We have implemented the models in JAVA code using the MATSim structure, integrating them with the pre-existent classes for drivers, streets, intersections, etc. In the development phase and in the primary evaluation phase a small test network was used that consists basically of two routes between point A and point B. For a final and more realistic testing scenario we identified a location in Berlin that seemed suitable for this kind of guidance control, and evaluated the models further by running them on the large Berlin MATSim network on that location.

1.4 Structure of this report

Chapter 1 and 2: Introduction, theory and reactive control

The first chapter gives an overall introduction, whilst the second chapter introduces feedback control as a traffic guidance method and presents the aim of the reports work. It also introduces the theoretical framework and the problems with reactive feedback control, i.e. using a measured output. It also presents the MATSim framework, the system studied, its dynamics and limitations. Also, the networks for testing the models, an introduction to the COOPERS project and a brief overview of the JAVA implementation is found here.

Chapter 3: The prediction models

Chapter three contains the basis for all predictive models in this reports work. It also describes the models and the evaluations of them on the test network.

Chapter 4: Evaluations on a Berlin scenario and conclusions

Chapter 4 presents evaluations from simulations on a Berlin network, chapter six comes up with conclusions from all evaluations and with ideas for further research that would push the development forward.

Chapter 5: Conclusion

2 Background

2.1 Route guidance theory

To introduce the relevant theoretical framework, the basic guidance scenario is here presented. Basically, drivers want to go from the point A to the point B and there are two different routes available. In accordance with the MATSim nomenclature, drivers will hereafter be called *agents* and point A and point B will be referred to as the *sign link* and the *destination lin*k, respectively. The agents have learned from experience what route has the shortest travel time. Optimally this will result in one of two possible cases. The first case is that all agents take the fastest route. The second case is that both routes are equally fast, and agents travel over both routs. Two routes can be equally fast if they have the same length and speed limit, or if one of them is shorter but currently contains more agents that make it slower than normal. At this situation every agents takes the fastest alternative; no individual agents would travel faster if he had chosen the other route. We say that the system is at *Nash equilibrium*, with an expression from game theory. The Nash equilibrium is the control goal for our route guidance. As mentioned earlier, the interesting scenario for route guidance is when something extraordinary happens, for example an accident. Since the agents do not know about the accident further down the route, many of them will take the slower route.

2.1.1 Guiding the agents with feedback control

The route guidance idea used in the preceding reports work of WP 4200 (IR-4200-2, 2007) was a classic feedback control. The system output $y(t)$, i.e. the difference in travel times for the routes, is called the *Nash time*. This was measured and used as feedback signal. The difference between this output and the wanted output y_{ref} is then sent to the controller that calculates the next input (guidance direction) to the system.

In other words, basic feedback control works by calculating the control input $u(t)$ based on the size of the control error $y – y_{ref}$. In our case the control error is the difference between current Nash time and the Nash time defined by the control goal, which is zero, corresponding to Nash equilibrium with equal travel time on both routes. A flowchart representation of the closed loop control system is displayed in Figure 1.

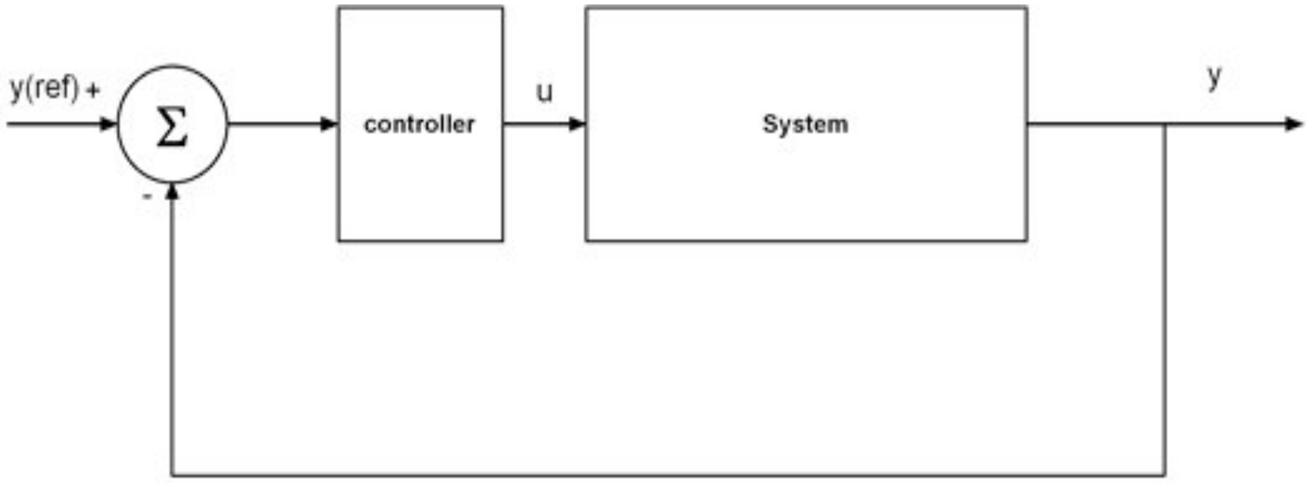

Figure 1. Feedback flowchart of classic feedback control

2.1.2 The feedback controller

There are many different controllers that can be used for feedback control. In the work reported here, we will solely use bang-bang control, or relay control as it is also called. This is an all-or-nothing approach. In every time instance, all the passing traffic is directed into the faster of the two routes. Mathematically, one can write the strategy in the following way. Here $y(t) = TT_{main}(t) – TT_{alt}(t)$, where TT_i means travel time for route i.

If $y(t) > 0$ *then* $u(t) = 1$ the traffic is directed into the alternative route

If $y(t) < 0$ *then* $u(t) = 0$ the traffic is directed into the main route

If $y(t) = 0$ *then* $u(t) = u_0$ no guidance is given, vehicles take either route

A bang-bang controller is robust in terms of that it is system independent. It has no control parameters that must be tuned for the specific routes, unlike for example the PID-controller. This is a major advantage, especially when no system models are at hand. One drawback is that such control tends to be too strong. If the control output is very close to zero, the bang-bang controller still controls in the same manner as if the control error were very big. This leads to an oscillatory behaviour. The oscillations can be avoided with a dead-zone on the output signal so that no control signal is given when the Nash time is within an interval close to zero.

The oscillatory behaviour of a bang-bang controlled system also depends on how often a new control signal is applied. Assume that the control signal is updated every five minutes. If the demand is high and all agents are directed into the same route for five minute, this route will have become considerably slower by the time the control signal changes. This obviously generates oscillations, which would be mitigated if the control signal would be proportional to the control error. However, if the control signal changed as soon as the system output changed, no oscillations would be generated in first place. But then one must have non time-delayed information about when the sign of $y(t)$ changes. Having such information, and in-car devices instead of variable direction signs on the roads, every car can be guided individually according to the latest calculated control signal. We return to the practical aspects of the communication of the control signal in the next chapter.

2.1.3 Problem with reactive control based on measured system output

Observations of the time difference between the two routes do not give us reliable information about what travel time an agent at the sign link can expect. The time lag from input (the control signal) to output (the Nash time), results in oscillatory system behaviour and a guidance signal that will never be up to date. To illustrate the time lag and the oscillatory behaviour, two graphs are shown below. We can see the time lag in the travel time diagram in Figure 2. Here, no control is switched on. The dashed black graph is the measured travel time on which the control signal is based. For example, at 4000 seconds simulation time the expected travel time for an agent entering the main route was 800 seconds. But when that vehicle reaches the destination link, its travel time is measured to 1000 seconds. The two curves contain the same values (in this test case all agents travels a complete route), but shifted depending on the length of the queue. If the travel time is 500 seconds, then the measured value will show up 500 seconds “too late”. The longer the travel time, the larger is the time lag.

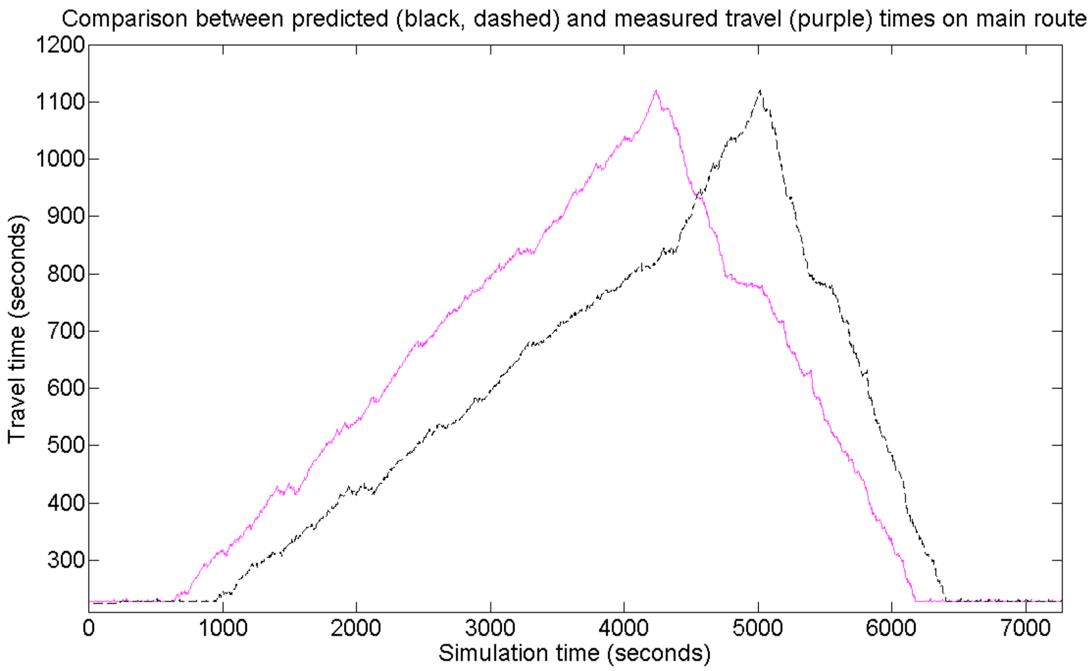

Figure 2. Illustration of the system time lag.

As the control signal is derived from “old” information, agents are directed into a route also after it has become the slower alternative, generating the oscillating Nash times showed in Figure 3. Unlike the last diagram, the Nash time plot was generated from a simulation with the route guidance activated.

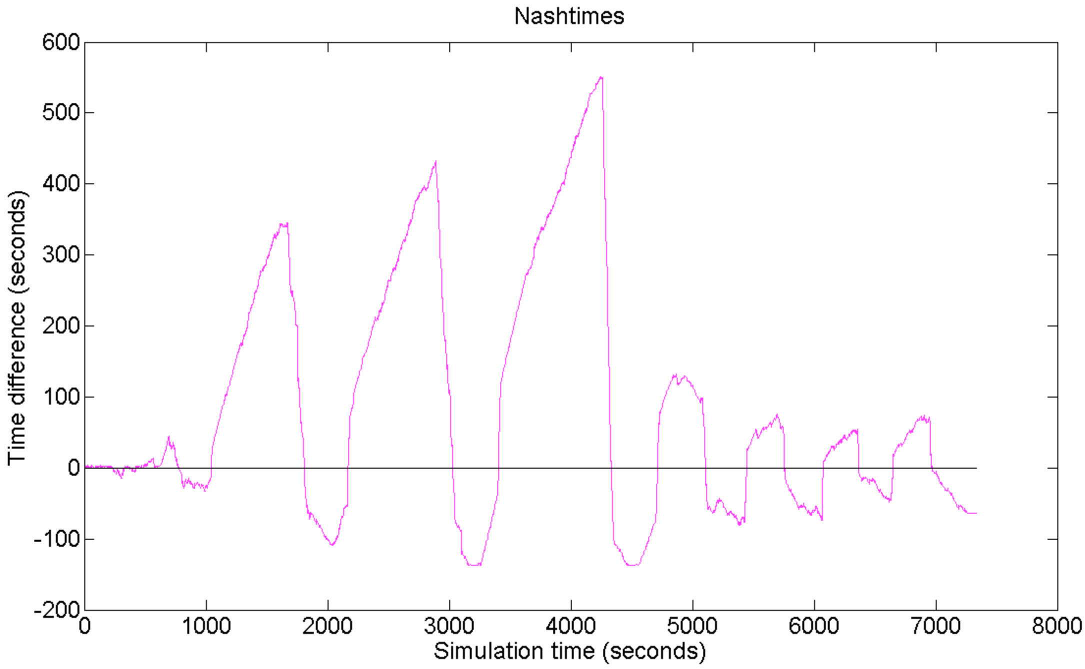

Figure 3. Nash times with reactive control.

The cause of the oscillating behaviour is the inadequate system output that we use for the control. As seen in Figure 2, there is a discrepancy between travel time values used for control and the travel times that are later measured for the guided vehicle. Therefore, if a queue is discovered, i.e. the measured travel times are bigger on one route; the controller starts sending vehicles in the other direction. But it is not until these vehicles reach the end of the route that the controller can notice the change that the control achieved. Thus, the system output needed is not observable, as the measurements are too old.

Our primal idea was to compensate for this by identifying the time delay of the system; i.e. after how many seconds a change in the control signal causes a change in the measured Nash time. This is problematic as the time lag is not static but represents one of the fundamental dynamics in the system. As showed in the travel time graph, the longer it takes to travel the route, the greater the difference between the current values and the measured. This is intuitively comprehensible as the time lag is determined completely by travel time on the routes, which of course depend on the traffic situation, e.g. queues.

2.1.4 Modelling theory – predicting the system output

Instead of using outputs that are "too old" we want to predict the output for the agents before they are guided. For that we need a model that predicts the travel times according to the present traffic situation. With predicted system output, the flow chart of the system will look like in Figure 4.

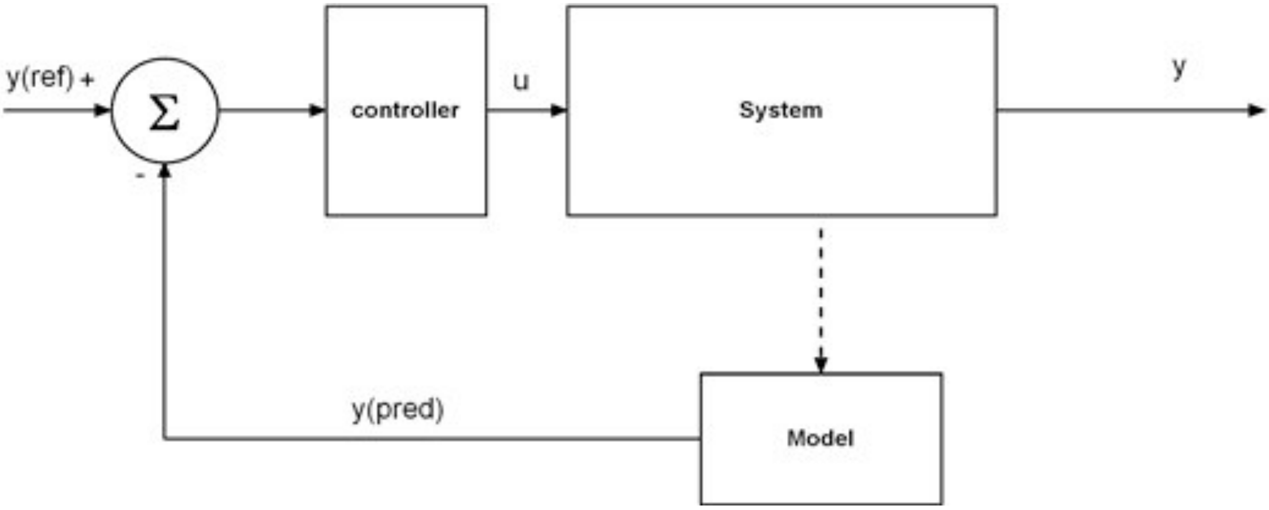

Figure 4. Feedback flowchart with prediction. The dotted arrow is the measurements from the system, i.e. number of agents on the links, etc.

There are different approaches to model dynamic systems. Physical modelling is often preferred for simple systems where the dynamics follow well-established physical theories. When no system knowledge is at hand, there are a number of black-box approaches. When identifying a system with such methods, parameters in differential equations are mathematically optimized to suite the relationship between input and output as good as possible. Time lags are often included in such models, but the problem for the traffic guidance is that the time lag between input (left or right for an agent) and the output (difference in travel time between that agent and another agent travelling on the other route) is varying over time, depending on current traffic conditions. If there is presently a long queue, it will take a long time before a change in guidance will be noticed at the end of the routes. Due to this and also for the reason that it is always a good idea to try the simplest things first, we choose the approach of physical modelling.

One could say that Figure 4 does not represent a feedback control loop, using a strict definition of "system feedback". There is no feedback from the system output *y(t)* – in fact the observed output plays no part at in the control. Nevertheless is there an obvious feedback from the system to the control signal, and therefore we will continue to use the term "feedback control" for the route guidance.

2.1.5 Guidance over more than two alternative routes

Our scenarios – and our modelling methodology – limit the controller to using two alternative routes for guidance. It is easy to see that a road network could be used more efficiently if the guidance control could direct traffic via several different routes. Later in this report we discuss the difficulties we had finding a suitable test location in Berlin. As we focus on highway accidents, it is hard to find one single alternative route that has capacity enough to handle a substantial part of a highway traffic flow – and still having a travel time short enough to be a reasonable alternative.

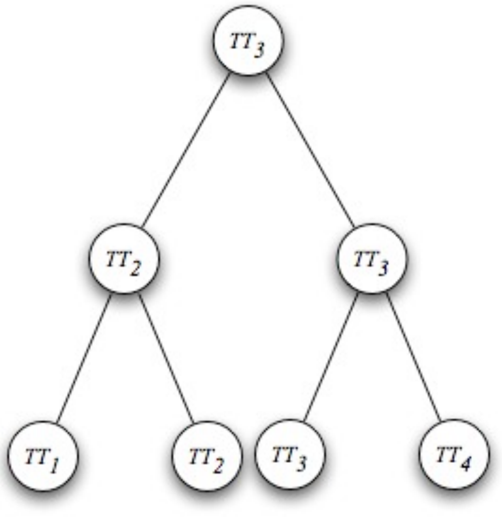

Figure 5. Diagram for multiple route guiding.

These limitations in the road network can be dealt with by using more than one alternative route to compensate for a capacity reduction on one big main route. It is possible and theoretically quite simple to connect many two-route controllers of the kind previously described, so that a traffic management system can guide traffic over many alternative routes. Figure 5 is a binary tree where every leaf represents the current travel time for a route, predicted with the models developed within this reports work. Every parent displays the travel time for the fastest of its children routes. The current structure of the tree in Figure 5 informs us that route 2 is currently faster than route 1, but route 3 is the fastest of all four alternatives. All agents going from A to B are therefore directed into route 3. As soon as any of the other routes become faster route 3, that route travel time will become the root of the tree. The guidance will shift, keeping the system at Nash equilibrium.

2.1.6 Modelling Traffic Systems for Route Guidance: Current Research

In order to be able to implement and evaluate route guidance algorithms, appropriate traffic models should be incorporated so as to be able to predict the vehicles' future travel times along alternative routes as well as the effect of route guidance commands to the overall traffic conditions. In the simplest case, conventional forecasting algorithms may be implemented, which – based on historical as well as real-time data – attempt to predict the travel times along pre-specified routes. However, such an approach, although proven quite efficient in providing travel time information and prediction, is not appropriate for the design of efficient route guidance algorithms since it does not incorporate a model of the traffic dynamics (and thus it cannot model and predict the effect of route guidance commands to the overall traffic conditions).

An alternative approach would be to use appropriate traffic models that incorporate – and appropriately capture – the effect of various traffic-related operations (such as traffic behaviour at motorway bifurcations, on-ramps and off-ramps) to the overall traffic conditions. Macroscopic, mesoscopic and microscopic traffic flow models may be used for such a purpose. The choice of the particular traffic flow model to be used depends on its accuracy on effectively modelling and predicting the so-called traffic state and, moreover, on its computational requirements; several traffic flow models that can model and predict traffic flow conditions quite accurately are not appropriate to be used within route guidance algorithms due to their heavy computational requirements that prevent their usage in real-time.

Due to the real-time computational requirements of route guidance algorithms, the majority of traffic models that have been used in route guidance algorithms applied in real-life, were of macroscopic type: for instance, the route guidance system AMOR (Messmer et al., 1998) which has been successfully implemented in the Scottish Highway Network and in the network of Aalborg, Denmark, uses an over-simplified macroscopic model (based mostly on store-and-forward traffic models) for predicting travel times as well as the effect of route guidance commands to the overall network traffic conditions. A more elaborate macroscopic model than the one used in AMOR has also been proposed and implemented by the same team that developed AMOR, within the traffic surveillance and prediction tool RENAISSANCE (Wang and Papageorgiou, 2005). RENAISSANCE uses a – validated using real-life data – second-order traffic flow model for modelling traffic behaviour at road segments and bifurcations; additionally, RENAISSANCE incorporates an extended Kalman filter approach for “tuning” in real-time traffic state and travel time predictions based on real-time traffic data. The application of RENAISSANCE in traffic networks of quite large size (e.g., the Amsterdam motorway network system) produced quite satisfactory results.

Despite the success of macroscopic-based models such as AMOR and RENAISSANCE in particular applications, macroscopic models cannot be as accurate and reliable in modelling and predicting the traffic state behaviour as microscopic models; microscopic models, since they model the traffic behaviour up to the individual vehicle level, can capture and reproduce traffic phenomena that cannot be modelled by macroscopic models (such as the effect of lane changing and merging in the overall traffic conditions; or the effect of individually different route guidance, where every vehicle has a potentially different guidance). Thus, if microscopic models can meet the real-time computational requirements of route guidance algorithms, it is preferable to use such models instead of macroscopic ones since the former can model more accurately and thus predict more reliably traffic state, travel times as well as the effects of the route guidance commands to the overall traffic conditions. It has to be emphasized that microscopic models are expected to be significantly more accurate in predicting of travel times over long routes (and thus over long time intervals), contrary to macroscopic models whose accuracy deteriorates when the route length becomes long. The microscopic model inside MATSim – which was adopted in this work – is an example of a microscopic model that can quite efficiently meet the real-time requirements for route guidance while preserving better modelling and prediction capabilities than macroscopic models.

The general idea adopted within the present work is to use physical conditions of the roads such as capacity, length, intersections, together with speed limits and sensors that can give information about current velocities, number of agents and traffic densities on a road segment. A reasonable assumption used in our model development, is that in case of queue, agents are served in this maximum rate. One could say that the simple queuing model used in this work, is similar to classic queuing theory, involving servers and agents. Bottleneck links are the servers and vehicles are the agents. Little or no attention is paid to the agent-to-agent dynamics.

2.2 Multi-Agent Transport Simulation (MATSim) as a telematics test bed

The abbreviation MATSim stands for *Multi-Agent Transport Simulation*. MATSim is a micro-simulator toolbox for demand modelling, simulation, iteration and analysis of transportation scenarios. “Micro-simulator” means that the simulation is agent-based, simulating on a per-car basis. Unlike other traffic simulating software, MATSim generates individual activity plans for all agents, who in turn dynamically generate a demand on the road network.

2.2.1 The street network

A traffic network consists of links that represent roads, or parts of roads, tied to each other at nodes, which represent intersections. A link has three basic properties that constitute its traffic dynamics; its free speed velocity, its flow capacity (number of cars that can leave the link per time unit) and a storage capacity (the maximum number of cars that can occupy the link at the same time) (Gawron, 1998; Cetin, 2003, page 8).

The intersections follow a three step logic: A car will move from one link to another if 1) it has arrived at the end of a link, 2) it can be moved according to the flow capacity of the link and 3) there is free space on the link that it is about to enter. If two cars from different links want to enter a third link at the same time step, the simulation will choose which car randomly proportional to the capacity of the incoming links (Cetin, 2003, page 9).

The links in MATSim have only one lane with a capacity that represents all lanes on the road segment in the real world. This simplification has become a problem on highway ramp locations in our simulations, a problem we will come back to in Section 3.6. Another simplification of the Berlin traffic network itself is that no traffic lights have been implemented.

2.2.2 Population, the agents

Simulations in MATSim need a number of interacting agents. The agents constitute a population that uses the network for different activities. Every agent has a plan that he tries to follow during the day. A plan is a sequence of activities such as home, work, leisure and travels that connect the different places where the activities take place. An activity has a start and end time and the agents know over which links they should travel to get to each activity. The plans are derived from real data. In other words, the population used in the Berlin simulations represents the real people in Berlin, having similar activities. To make the simulation runnable on standard personal computers, the population file normally used in Berlin simulations has approximately 350 000 agents, constituting a 10 percent sample of the total number of agents in Berlin. The capacity of the network is reduced to a corresponding level in order to keep the simulation realistic.

2.2.3 Replanning

To develop traffic situations that resemble real world scenarios, the agents can re-plan from day to day (Cascetta and Cantarella, 1991; Raney and Nagel, 2006). In MATSim, altering an agent’s behaviour is possible only through modification of its plan. The system remembers several plans for every agent and measures the performance of each plan. In this way agents learn which plans work well, and discard plans that do not work well. In the context of the present study, the most important aspect of a plan are its *routes*, and the most important aspect of the performance is the *travel time*.

In general it makes sense to consider additional choice dimensions as well, such as (departure) time choice or location choice.

The route replanning intuitively behaves in the following way: If the links that they travelled were congested yesterday, agents might try another route today. After a number of iterations of this sort, the population learns to take the approximate best route alternatives as long nothing unexpected happens, such as an accident. In the Berlin simulation, these “relaxed plans” are the result of 80 iterations of the same day. After these 80 opportunities to re-plan, it is expected that the agents take close to optimal routes between their activities, and no further day-to-day replanning is made. This implies that there is no need to actively guide the traffic unless something extraordinary happens that causes queues to build up unexpectedly. This results in the so-called “no control (base) case” for the remainder of this text: Travellers have pre-computed and fixed departure times, and pre-computed routes. These are nearly optimal under “normal” conditions. But even if an incident occurs, travellers are assumed to stick to these plans except if some control measure is switched on. As all Berlin simulations were made from that same iteration, the day-to-day replanning was never active within this report’s work.

Agents can also do *within-day replanning* (Cascetta and Cantarella, 1991; Rosetti and Kiu, 2005; Illenberger et al., 2007). Under certain conditions, e.g. if a link ahead looks congested, the agent can modify its day plan spontaneously and choose a faster route. It is this feature, which makes it possible for agents to react to unforeseeable incidents, that is used for the online traffic guidance that we develop in this report’s work. According to predictions of the travel times on different alternative routes, the fastest way is provided to the agents, which will listen to the recommendation with a certain probability, a compliance rate. Other within-day replanning features than the route guidance as investigated in this report are deactivated in our simulations to increase transparency.

2.2.4 The COOPERS project and the test scenario.

From its start in 2006, the COOPERS project has focused on developing telematics applications between vehicles and infrastructure in order to provide infrastructural and safety related status information. Several services were defined in the COOPERS project to improve traffic safeness and flow. As we have already indicated in our introduction, some of these COOPERS services could be improved with our work. Namely the services "accident/incident warning" (S1a/b), "traffic congestion warning" (S6) and "recommended next link" (S11) could benefit from our research. But also the "estimated journey time" (S10) could be calculated in a more meaningful way. Therefore improvements could go into these two fields of interest:

- Selective traffic jam warning and guidance based on current traffic situation on highway segments
- Short term ETA (Estimated Time of Arrival) based on current traffic situation on the road network

The present text is primarily concerned with the former, although an investigation of the latter would also be possible with the presented simulation technology.

2.2.5 Berlin test location

For the evaluation of route guidance using the prediction models that were to be developed, we chose a Berlin highway section relevant according to the aims with the COOPERS project. In addition to these we had the following criteria on the test locations:

- The main route must have a high demand during simulation hours. When an accident reduces the capacity of the road temporarily and partially (e.g. to 50 percent for an hour), this should generate a queue that increases the travel time on the main route so that the alternative route becomes faster.
- The alternative route should have capacity enough to handle the extra traffic that it receives due to the accident on the main route. If it is not big enough to take any additional traffic, then it will not be a real alternative.
- The alternative route is generally longer than the main route, but it should not be too long. Under normal conditions agents take the main route because its cost, in terms of travel time, is lower than for any of the alternatives. Nevertheless, the alternative should not be as costly that you stick to the main route; no matter how much time you will spend queuing.

The location finally chosen is on the northbound highway 111 in north-western Berlin, close to the Tegel airport. After analyzing MATSim simulations, this location seemed to fulfil the criteria above and to be suitable for our purposes. The highway constitutes the main route and has a capacity of 3967 agents/hour, while the alternative route has a capacity of 1858 agents/hour. To make the simulations run faster, the population of Berlin is reduced to 10%. This is compensated for as the capacities of the links are also reduced, but to 13%, which is something that has been tried out by VSP to be feasible. This must be kept in mind when studying the tables that contains number of guided agents. An accident that reduces the capacity by 50% is set far north on Highway 111, red on the map in Figure 7. The blue route is the alternative route. The traffic dynamics of the alternative route is quite different from that on the main route in terms of having many intersections and therefore a lot of additional traffic. Predictions on the alternative route should therefore be harder to do than for the main route that only has one additional on/off-ramp location.

We experienced some problems with the road network while trying the route guidance initially. The on and off ramps of the highway were congested immediately when just a few agents were directed off the highway and onto the alternative route. The flow capacities of these links were doubled, and in this way it was made possible to evaluate route guidance on the Tegel location. Increasing the capacities is not necessarily equivalent to widening the road in the real world. The effect can also be achieved by a telematics modification of the intersection, i.e. giving the guided agents a special treatment (e.g. a green light).[1]

[1] The capacity modifications (link number and modification factor) can be found in the file TrafficManagementConfiguration.xml. Naturally, the capacities are constant over all simulations.

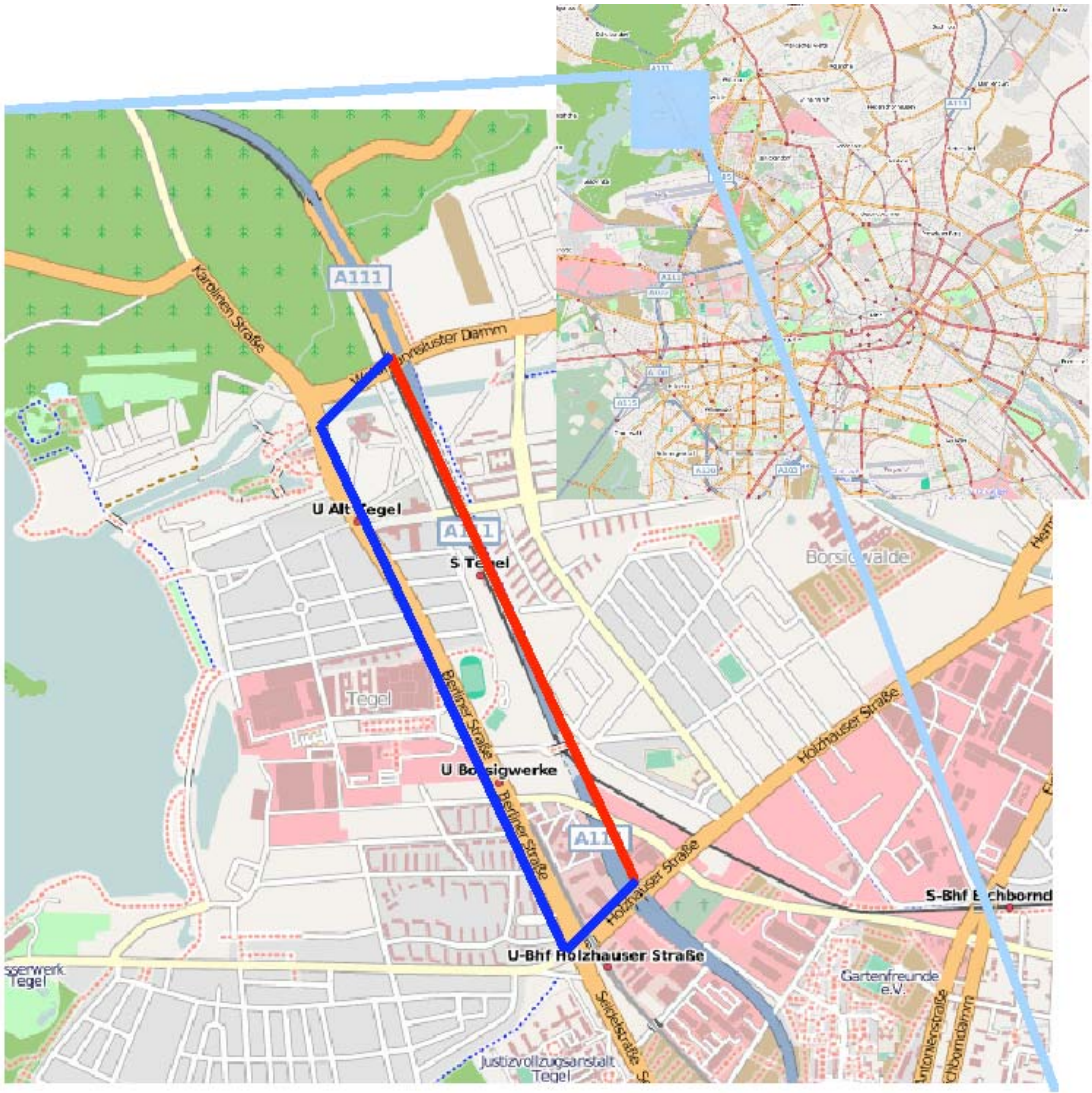

Figure 6. The main route (red) and the alternative route (blue) of the Tegel test location. The traffic that we consider moves towards the top. (Open Street Map)

2.2.6 Infrastructure needed. Coopers-devices and VDS

Our models and implementations are fairly flexible. They require sensors that count vehicles at certain measurement points. Unlike when using reactive control based on measured travel times, the prediction models presented in this report do not require that the vehicles be tagged, just counted. The sensors must provide information about number of vehicles on a road segment ("links" in MATSim) as well as the current in- and outflows to the route.[2] The route guidance sent to in-car devices, or to a stationary variable direction sign, a *VDS,* at the sign link. In this report's work is assumed that individual in-car devices are available such as the RDS-function of the car radio or a GPS, and we refer to them as *COOPERS-devices*.

---

[2] It should be noted that for the benchmark case, called "reactive control", it would be necessary to have sensors that are able to measure travel times from one part of the network to another, which could be best achieved by tagged vehicles.

For the simulations we assumed that 80 percent of the receivers of the guidance actually follow the advice. The rest stick to their original plans, perhaps thinking that their own experience of the traffic is more reliable than the automatic traffic management system. The 80 percent compliance rate was used in the preceding work of WP4200 (IR 4200-2) and to us it seemed to make the simulations more realistic. The positive effect of the guidance (measured in travel times, Nash times and also model fit) is expected to be lower when not all agents comply with the control signal. Still, the indications should point towards the same conclusions, regardless of the compliance rate.

COOPER-devices should be able to supply the relevant information, but if they only will be able to receive and not send information, sensors at certain locations could be used. Instead of using a COOPERS infrastructure, the models developed in this reports work can be used with variable direction signs. The biggest difference would be that the direction sign could not be updated too often since that would confuse the drivers. Further discussion about VDS implementations can be found in the COOPERS report IR 4200-2.

Concerning the differences between in-vehicle systems and road-side-systems, one has to look at incoming as well as outgoing communication. For the incoming communication it has been pointed out by the Report from WP4200 that an in-vehicle control could give an individual re-route recommendation to every driver. If the recommendation is to be given by a VDS, this information needs to be quantized to a certain length of time, e.g. by half a minute as being done in the WP 4200-2 report, for supplying a by passing driver with not too many different recommendations. Therefore the sensibility of the control system decreases. In this report (WP 3800), the calculation of this individual information is being done second-wise, that is with the highest possible time resolution the simulation system was run.

It terms of outgoing information the simulation system is robust against whether the information is given by an roadside system, like an induction loop or traffic eye, or given by the coopers device itself. For a successful operation the system needs information about the ingoing traffic and the outgoing traffic for a distinctive pieces of the road, namely the "matsim links" which are between each intersection or off- and on-ramps. In the simulation system, this information is extracted from so called "events", which are defined as observation of the --synthetic-- real world. These observations could be gathered in an actual real-world scenario through several different channels, as pointed out above

2.2.7 The test network

The simulations on the Berlin network are very time consuming and the traffic situations are complex. To develop our models, a test network was designed together with populations suited for trying the features of the models. The test network is shown in figure 8.

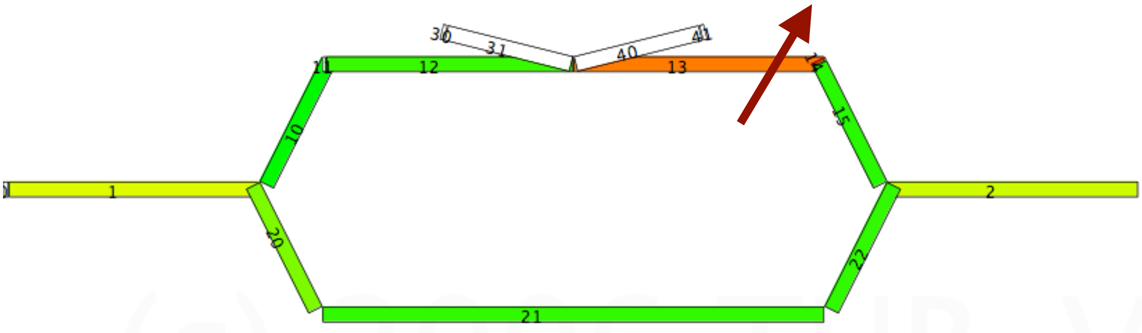

Figure 7. The test network. Traffic moves from left to right. *Green colour represents free flow conditions and red represents congestion.The accident location is indicated by the arrow.*

The test network is designed so that the capacity of one link equals the sum of the capacities of the incoming links. This seemed like a reasonable assumption and suitable for testing the features of the models. Agents go from left to right and the guiding takes place at the first crossing. The upper route is referred to as the main route and the lower as the alternative route. The accident is set on link 14, which is the short link immediately after link 13, which is the long red link in figure 8. On the main route, there are two additional links, one for incoming and one for outgoing traffic if such scenarios are needed. Traffic from or to these link will not be guided and is looked upon as a disturbance. The colour scheme represents the traffic density on the links. White links are not trafficked at all, green links have low density, yellow represents medium density, and red means that the link is completely congested.

2.2.8 Implementation

MATSim is a large and complex simulator and during the work with this report, most of the architecture was perceived as a black box whose inner dynamics was known nor to us, nor to the models. To facilitate the reader's understanding of MATSim and how the prediction models were integrated, this chapter presents the JAVA-classes that are the closest and most relevant to the prediction models.

2.2.8.1 VDSSign: From Prediction to Guidance Signal

The name `VDSSign` is a remnant from the earlier work of WP4200, where we initially used Variable Direction Signs to communicate with the agents. This class, as well as some other implementations, was refactored during our work, but the name remained.[3]

When the route guidance is activated as a traffic management module, an instance of `VDSSign` is created for every two-route guidance system. The task of `VDSSign` is to ask the `ControlInput` object for the current system output and transform this into a control signal. When the control input is an object of `ControlInputImpl1`, VDSSign gets the measured Nash time (reactive control). Otherwise it gets a system output predicted by either `ControlInputSB` or `ControlInputMB` (more about these classes in Part II).

The VDSSign sends the Nash time a `FeedbackControler`, which returns a control signal. The `BangBangControler` returns –1 or 1, indicating the fastest route. The `PControler` – which is not used as argued for earlier in this report – returns a value in the interval [–1, 1], depending not only of the sign of the Nash time but also the size. VDSSign then communicates this value to the MATSim system for withinday-replanning, which makes the agents replan and change their route. If the Nash time is zero, the VDSSign does not give any guidance and the agents stick to their original plans.

2.2.8.2 ControlInput: Making the Predictions

The Interface `ControlInputI` defines the different ControlInput implementations, i.e. the prediction models that we developed, by demanding some necessary methods like the `getNashTime()`. A more important class is `AbstractControlInput`, which contains the objects that are common for all prediction models. For instance, this is where all the information about the network is read when the simulation starts, like free speed travel times, inlinks and outlinks, natural bottlenecks, etc.

The abstract class also contains the important code that measures travel times. This is done by calling the MATSim method `handleEvent()`, which can be considered to represent sensors that detect when a vehicle leaves or enters a link. In this way flows and agents are measured and the models get the data they need for predicting the travel times for an agent at the sign link. In the ControlInput implementations, `getPredictedTravelTime()` predicts the travel time for every route, and the difference is being returned by `getPresictedNashTime()`. The class variables messageHoldTime defines how often a new guidance signal derived from the system output.

---

[3] As said before, a big difference between using a VDS-sign and in-car COOPER devices is that the guidance signals going out to the agents can be updated every second having COOPER devices, while updating the guiding on a big road sign every second will confuse the drivers quite severely. The java class VDSSign is still the class communicating with the agents, but in our implementation it can do it without paying any respect to the update intervals.

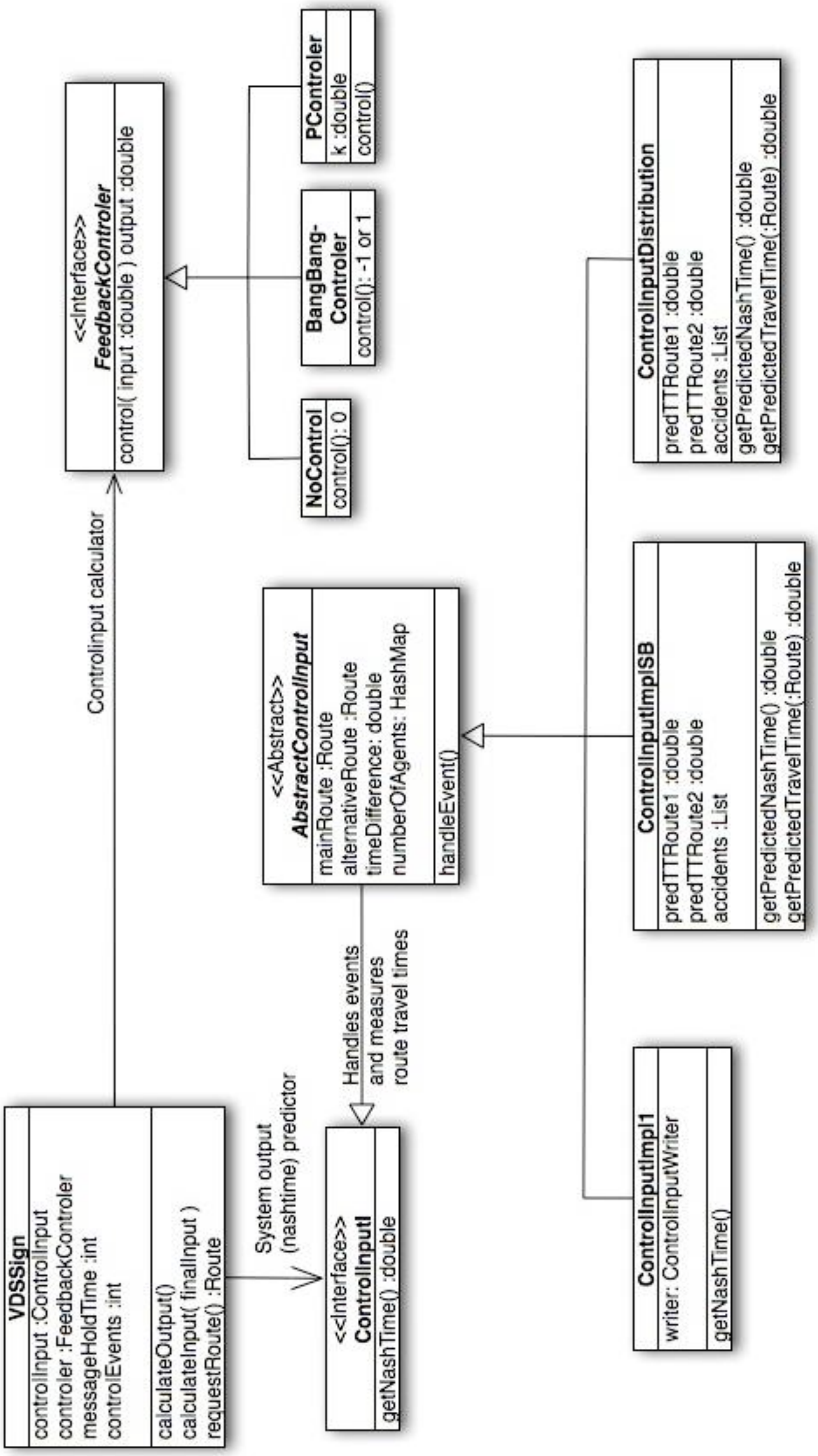

Figure 8, An UML–representation of the controller and prediction classes. The UML standard for illustrating class structures can be found for example on the web page: http://www.agilemodeling.com/artifacts/classDiagram.htm (29 Jan 2008).

3 The performance of different prediction models for travel times and traffic jams

3.1 The basic theory used in the models

Starting off from a simple flow based queuing model, we need the number of cars on a road and the capacity of that road to calculate the travel time. As said earlier, in MATSim roads are represented by so called links. If the current link has a capacity of letting *c* vehicles out per second, a maximum of *c* vehicles can exit the road in *t* seconds. Thus, *c* is the outflow rate from a link when a queue is present. Assuming that all agents travel at the maximum allowed speed, called *free speed*, the time it takes to travel a link if no queuing is needed is the length of the link divided by the free speed. This time is called TT_{fs} , which strand for Travel Time at Free Speed.

If we count the number of vehicles that currently occupy a link, we can calculate the time it would take for all the agents to be released off the link, using the maximum outflow, *c*. This time could be called the queuing time, TT_q. In queuing theory terms, we calculate the total *serving time* for the agents. Comparing TT_{fs} and TT_q and choosing the greater of the two gives us an estimate of the link travel time. If $TT_q < TT_{fs}$, this can mean one of two cases, that we treat in the same way. Either there is a queue short enough to be completely served before an agent about to enter the segment arrives at its end. Else the agents presently on the road segment are even fewer, generating no queue at all. An agent entering the road segment will in both cases travel the whole segment by free speed, arriving at the end of the segment in TT_{fs} seconds.

Doing this kind of predictions for an entire route, consisting of several links, our first approach was to find one bottleneck link of the route and using this link's capacity for the predictions. This first model is called the *Single Bottleneck Model*, or just *the SB,* and will be described in detail in the next chapter. Focusing on one single bottleneck seemed as a good methodology a priori, based on the assumption that there are no big inflows or outflows that influence the traffic situation to great extent.

It is important to realize that TT_q is not just the time the queue needs to dissolve, but also our predicted travel time for an agent entering the route. Even if the agent spends most of the time on the route driving free speed and only a little time queuing at the bottleneck, his total travel time to the bottleneck will be determined by when the agent in front of him pass the bottleneck.

The assumption used above, that the maximal outflow of a link occurs when it has a queue might seem intuitively strange. One might think that a queue would mean lower velocity, and therefore that the outgoing flow would be beneath the maximum. Our definition of a queue states, however, that a queue arises when the flow of agents onto a link is higher than the outflow capacity of that link. This means that when agents are queuing on a link, the actual outflow has reached the maximum allowed by the physical characteristics of the link. Only when the capacity is reduced – as in the accident test case – will the presence of a queue imply that the outflow is beneath the normal maximum.

3.2 Evaluation measures

There are different ways of evaluating a model. The model itself can be evaluated through *fit* values. The fit value measures how much of the travel time variance that is explained by the model, in other words its prediction accuracy. It is calculated by comparing predicted and measured travel times on the routes, as in the expression below. Matlab functions were written to do this evaluation. The prediction error $(y_i - \hat{y}_i)$ is calculated every second so index *i* corresponds to the simulation time. The simulation is set so that a new $\hat{y}_i$ is calculated every second. This should be compared the travel time observed at the end of the route y_i seconds after $\hat{y}_i$ was calculated (when the agent entered the route). Therefore, the data set with predicted values was pre-processed so that $\hat{y}_i$ is the travel time predicted for the agent whose observed travel is y_i. Another interesting value is the Nash times, since minimizing them is the control goal used. The Nash time is, as explained earlier, the travel time difference between the routes. Instead of “time difference” we will continue to use “Nash time” for the reason that this is the term used in the MATSim code. With the following formula we get a scalar value reflecting the size of the Nash deviation for an entire simulation (the value is hereafter referred to as *standard Nash deviation)*.

$$fit = 100 \cdot \sum_i \frac{(y_i - \hat{y}_i)}{y_i} \qquad \sigma_N = \sqrt{\frac{\sum_i^N (y_i - \hat{y}_i)^2}{N}}$$

y_i is the measured Nash time, $\hat{y}_i$ the predicted Nash time, σ the standard deviation of Nash times and *N* the number of predictions.

It is worth mentioning that the predictions are done every time step, in our case every second, but there are not always agents passing and receiving the guidance. Also the measured times are extracted every second, which means that the latest measured travel time will be used in every step. In a time step when no car left the route, we compare the travel time measured for the latest vehicle that left the route with the travel time predicted *this time step*. If the traffic situation changes (e.g. number of agents on the route links) and the predictions reflect that, the fit value will be affected negatively even though the model is doing a good job, providing the controller with a more up to date input. This deviation is normally very small compared to the value of *y* and cannot even be seen in the travel time plots presented n the report. The error becomes clear in a scenario where only very few agents travel on one of the routes, since predictions still are made every second. Another interesting way of evaluating the models is to see how the *average travel times* are changed for the agents when using a certain model compared to using reactive control or no control. The travel times for every agent were extracted from the simulations. The average travel time for each of the two routes and the average travel time for both routes are calculated and used as measure of the system performance. Also the number of agents travelling on each route will be given as an evaluation measure. This measure allows us to compare the number of redirected agents for different models, and informs about how many agents “suffer” from travelling on the slower route, and respectively how many benefit from taking the faster alternative.

We use one more evaluation measure, the *score*. The score translates changes in travel time to economic effects (measured in euros per agents). This measure works best with a realistic population, and was therefore used only in the Berlin scenarios. The scoring functions will be explained further in chapter 10.

3.3 The Single Bottleneck Model

3.3.1 Description of the model

The model is based on the following strategy. We identify the bottleneck on the route, which is either the link with the lowest outflow capacity or, more interestingly, a link where an accident partially reduces the normal capacity. All the agents that are on the part of the route before the bottleneck are counted and so the travel time is predicted. The free speed time needed to reach the bottleneck is then compared with the time it takes for the cars on the route to pass the accident (calculated with the reduced capacity outflow in case of a known accident). If not all agents ahead of the agent currently being guided (the guidance object) will have passed the bottleneck within the free speed driving time; the guidance object will have to queue. In this case the travel time for the guidance object is predicted to be the queuing time up to the bottleneck. This is added to the time it takes to finish the remaining part of the route, which is assumed to be travelled free speed. Finally, after the travel time has been predicted in this manner for both routes, the predicted time difference is returned to the guidance system.

In the end of this chapter we discuss some traffic dynamics that could cause problems for this rather simple prediction model. We expect, however, the route guidance to perform better with this model than with observed travel times. Recalling the background given in chapter 3, we can expect the effect to be greatest when the accident is far away from the sign link, with much congestion in between.

It is a prerequisite that the accident is “known” to the model, i.e. that the reduced capacity can be used in the calculations instead of the normal maximal capacity of the link. Naturally also the location of the accident must be known. This information is accessed easily in MATSim, but in the real world it is needed that the temporary capacity reduction is identified manually, for example by traffic management personnel. If the reduction is due to road work or similar, such information can be acquired beforehand. Later in the report a methodology for detecting capacity reductions automatically is presented.

It is important to realize why the travel time is predicted as the queuing time, and not the (free speed) travel time needed to reach the queue + queuing time. This would be incorrect because the queue will have partly dissolved when the agents reaches it. In fact, the predicted queuing time will be reduced with exactly as many seconds as it took for the agent to reach the queue, driving free speed. In other words, the TT_{fs} part is “eaten” by the TT_q part.

As discussed previously, when the traffic situation on the route is not affected significantly by inflows or outflows, i.e. additional traffic coming onto or going off the routes from intersections along the route between A and B, there is no reason to argue for that there is more than *one* bottleneck on a route. If the bottleneck is the narrowest part of the route, there must be significant additional inflows after it in order to build up a queue at a subsequent location. If there are two bottlenecks that are equally narrow, the model chooses the latter on the route since that bottleneck could swallow some of the time it takes to queue on the earlier, but never the other way around. When the background noise of inflows and outflows is large (in the Berlin scenario this background noise sometimes exceeds the proportion of controlled agents on the alternative route), it becomes problematic and should be handled. The multi-bottleneck model presented later was developed for this reason.

3.3.2 Mathematical description

The notations used in the expressions are described below. The sums' index refers to the sequence of links on the route, where $i = 0$ refers the first link on the route.

tt_i free speed travel time for link i

x_i number of agents on link i

b index of the bottleneck link (the link where the accident has occurred)

n index of the last link on the route

c_b bottleneck capacity (vehicles/second), assumed to be known to the model

The free speed travel time for the entire route is the sum of the free speed travel time of all the links that constitute a route:

$$TT_{fs} = \sum_{i=0}^{n} tt_i$$

The travel time in case of queue, TT_{queue}, is expressed by dividing the number of agents up to the bottleneck with the bottleneck capacity and last adding the free speed part.

$$TT_{queue} = \frac{\sum_{i=0}^{b} x_i}{c_b} + \sum_{i=b+1}^{n} tt_i$$

The largest value of the two calculations is taken as the travel time prediction of the route:

$$TT = \max\left(TT_{fs}; TT_{queue}\right)$$

These calculations are made every second for both routes. The Nash time is calculated as the difference between the two routes (called the "main route" and the "alternative route"):

$$\hat{y} = TT_{main} - TT_{alt}$$

This is the value used as input to the feedback controller, which guides the agents into the faster of the two routes.

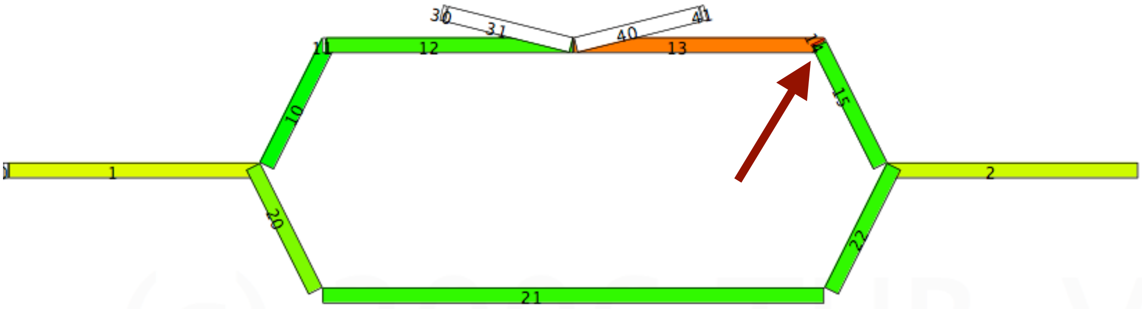

Figure 9. The test network used for the primary evaluation round.

3.3.3 Evaluation

In this chapter, the model performance will be evaluated on the test network described in section 3.8. Figure 10 helps us recall the scenario. According to the agents' plans, the traffic flows steadily and with equal density over the two alternative routes. The accident indicated by the arrow reduces the capacity, which the guidance system compensates for by directing a larger proportion of the agents into the alternative (the lower) route. An accident occurs at 7:15 in the morning and the road capacity is restored at 8:15. Between those hours, the capacity is reduced to 50%, which can illustrate roughly the real world case where one or two lanes are shut off due to the accident.

First, a scenario with no control was run, but with the model predicting the travel times. In figure 11 we can compare the measured travel times (purple line) with the predicted (black, dashed line).

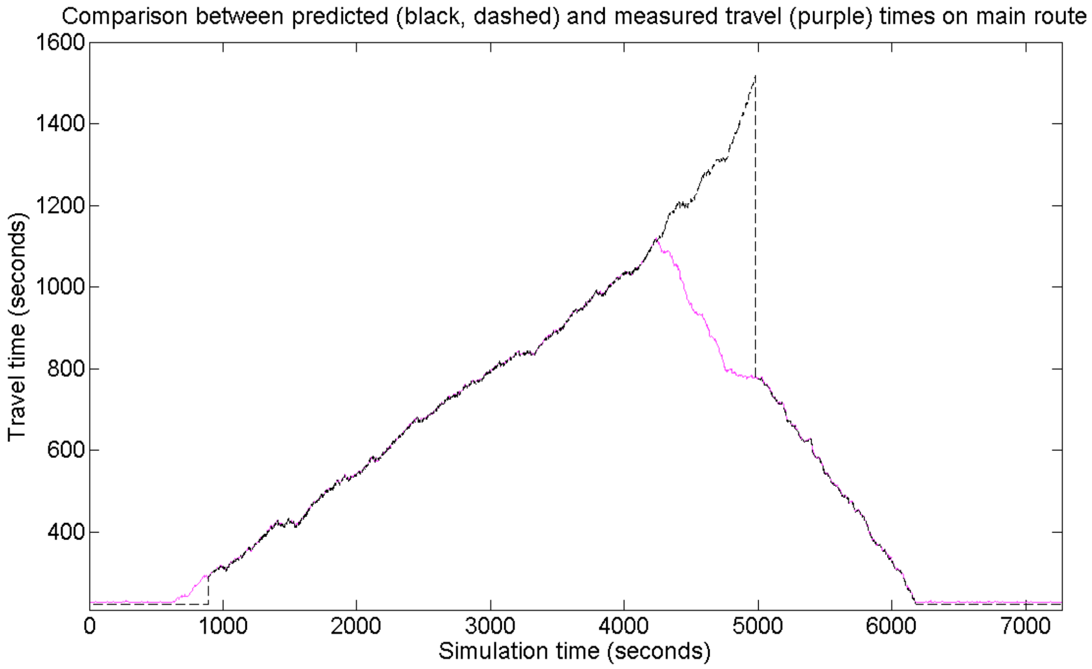

Figure 10: Comparison of the predictions of the Single Bottleneck Model and the actual travel times. No guidance control is applied.

As shown, the model predicts almost perfectly except for when the accident starts and ends. To deal with the first error, one would need to know that an accident will occur at a certain time and where and how much it reduces the capacity, so that the model can predict the future travel times correctly and the guidance control can guide pro-actively. Similarly, the restoration back to normal capacity must be known some time before, so that one could guide the traffic as if the accident had already ended before it actually does.

Since the model predicts the travel times accurately according to figure 11, travel time improvements are expected when switching on the control. In table 1, the travel times achieved with the Single Bottleneck Model (abbreviated the “SB”) are compared with the travel times of a non-accident case scenario, an accident scenario without control and with reactive control, i.e. the control using measured system outputs.

Table 1: Fit values and travel time improvements of the SB model. Accident reduces the normal capacity to 50 percent on link 14.

	Normal Case (no accident)	Accident Case, No Control	No Model (reactive control)	Single Bottleneck Model
Main route:				
TT	229	568	359	232
Agents	3999	4000	3258	3405
Fit	-	-	58.8	96.8
Alternative route:				
TT	227	227	276	229
Agents	3999	3999	4741	4594
Fit	-	-	83.4	98.3
Total				
Std. Nash deviation	2.6	425.8	187.5	8.6
Average TT	228	397	310	230

The average travel time increases by 397 – 228 = 169 seconds when the accident occurs, which is a percentage increase of 74%. This increase in travel time is smaller with reactive control, which reduces it by 87 seconds. The percentage increase is thus 36%. Using the model, the average travel time decreases to 230 seconds, only 2 seconds worse than the case without an accident. From these runs, we can also see that the fit values are much higher using the model compared to using measured travel times. The fit on the main route improves from 58.8 to 96.8 on the main route, and from 83.4 to 98.3 on the alternative route.

In the table, we can also see the standard Nash deviation. Our control goal is a Nash time of zero, although in reality, a Nash deviation that is similar to the deviation without an accident is quite enough. The table shows that the reactive control cuts the standard Nash deviation in half compared to when no control is applied. The Single Bottleneck Model further mitigates the effects of the accident and brings the standard Nash deviation down to one fiftieth of the no control case deviation. As expected, it can be seen in figure 12 that the only severe deviation is in the beginning, when the accident occurs.

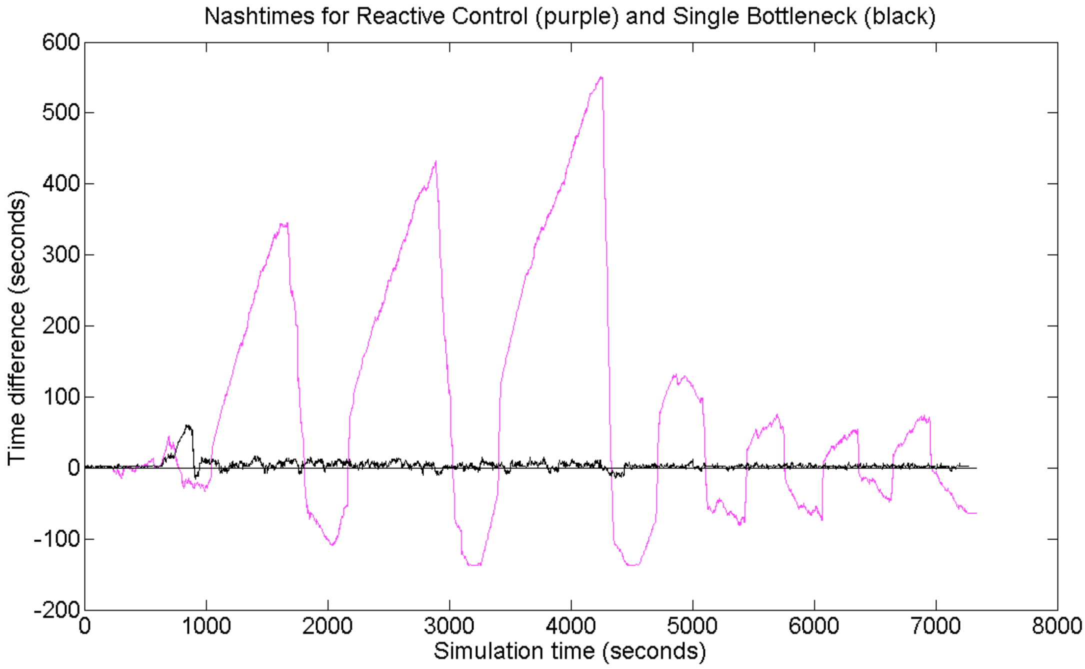

Figure 11: Nash times Comparison between reactive control and Single Bottleneck control

3.3.4 What does the Single Bottleneck model not handle?

Due to delays caused by technical difficulties, the performance of the prediction model was not evaluated on the Berlin scenario until in a late phase of the reports work. Before that, however, the effect of some traffic dynamics on the prediction model were analysed theoretically. The following problems were pinpointed in order to develop the model further:

- Non-homogenous traffic density on routes. To the model, the route consists of two parts; the part before (= upstream of) the bottleneck and the part after the bottleneck. When the traffic is concentrated to the beginning of the pre-bottleneck part, the Single Bottleneck model can falsely predict that an agent at the sign link will not queue at the bottleneck. The longer the route and the more heterogeneous distribution of agents, bigger will the prediction error be.
- Inflows and outflows during the prediction time horizon. The Single Bottleneck model only takes into account the amount of traffic that is *on* the route when the prediction is done. In case agents leave or enter the route between the sign link and the bottleneck, the queuing time will change.

Handling these problems is the subject of the two following chapters. Another feature that we wanted to develop in order to make the models less dependent of human operation was *automatic incident detection.* At a late stage, when the Berlin scenario was running, the effect of yet another model assumption was detected – there were two separate queues on the alternative route. In the Single Bottleneck model the travel time is assumed to be determined by not more than one queue. The topography of the test network had been too simple to generate a situation with multiple queues. Nevertheless, the last chapter in this part of the report is dedicated to the *Multi-Bottleneck model*.

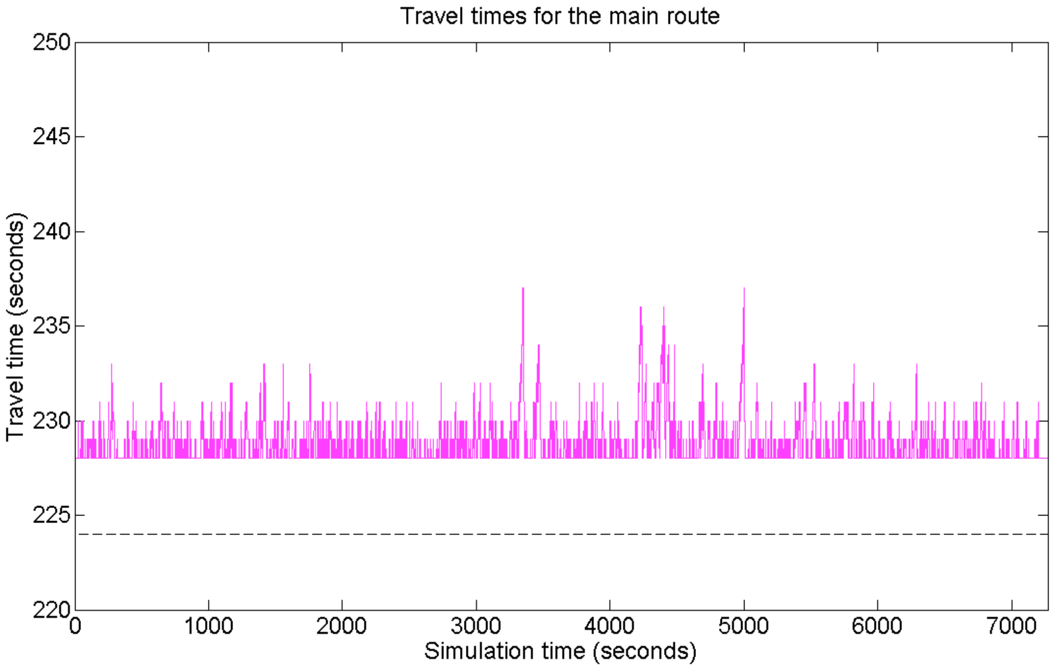

Figure 12. Travel times for the main route without an accident. It can be seen that not all agents travel with free speed even though there are capacity left. Maximum capacity of a route is 3000 agents/hour and in this run, there are 2000 agents/hour.

Figure 13 contains two prediction errors that will not be subject to further problem solving in this reports work. The noise in the observed curve is due to non-homogenous traffic density on links. There are often micro-queues on single links that have much traffic, but not enough traffic to generate a "real" queue.[4] This randomness makes the simulation more realistic since not all vehicles in a real world scenario drive at the speed limit at all times (of course, some vehicles would also drive faster than that, which is not possible at all in MATSim). No stochastic influence on an agent's travel time will be modelled. However, one should have this random noise in mind when the prediction models are evaluated.

Apart from the noise, figure 13 also shows a static prediction error (four seconds in this plot). Knowledge about the MATSim dynamics tells us that this error stems from the fact that it takes one second for an agent to go over a node. This could of course be compensated for in the models to achieve better predictions, but since this simplification in the simulator does not reflects the intersection dynamics in real traffic very well, we chose not to compensate for such errors.

[4] On probable cause of the noise is that the plans for the agents are created with a randomized method. Two agents trying to leave the same place simultaneously will cause a instantaneous demand peak, even though the average demand is below the maximum capacity. On of the agent will have to wait and try again in the next simulation step (which is by default one second later).

3.4 The Distribution Model

3.4.1 The problem with unevenly distributed traffic

The problem with *heterogeneous or unevenly distributed of traffic* arises when there is much traffic on the beginning of the route and only sparse traffic on the rest. Consider the situation in figure 14. There is heavy traffic on the first part of the route and after that almost empty up to the bottleneck. The basic Single Bottleneck predicts that the agent now being guided will not have to queue, based on the number of agents on all links before the bottleneck. The sparsely trafficked part lowers the average below the critical number of agents needed for a queue to build according to the SB model. Nevertheless, when the "heavy charge" in the beginning of the route reaches the bottleneck there will instantly be a congestion containing all the vehicles in the charge. Thus, the guided vehicle will queue even though it was expected drive free speed. If the prediction was correct, the vehicle would not have been guided into the lower route.

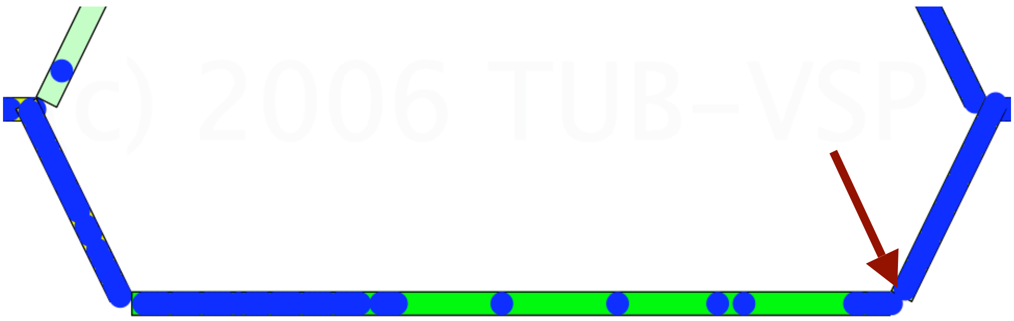

Figure 13. Sparse traffic on the part before the accident. The arrow indicates the bottleneck.

Because of the distribution of the traffic is not taken into account, the standard Single Bottleneck model makes prediction errors for the scenario described above. To detect heterogeneous traffic patterns we added a distribution check functionality to the Single Bottleneck model.

3.4.2 The SB model with distribution check

To solve the problem, multiple queuing calculations are done, i.e. the basic Single Bottleneck model approach is used a number of times, but for several subparts of the route. Important is that the same bottleneck capacity is used in the calculations for all parts. As in the SB model, agents are presumed to travel with free speed on all links after the bottleneck. Thus, the links after the bottleneck are added to what is called the *free speed part* of the route. The approach is then to go through the links, one by one, starting from the bottleneck and proceeding backwards against the beginning of the route. For every link, we check if the agents on the link are enough to cause congestion at the bottleneck. This is done with the basic SB method, i.e. the free speed travel time of the link is compared with the hypothetical queuing time; the time it will take for all agents on the link to pass the bottleneck using the number of agents and the bottleneck's outflow capacity. This is a way of finding out if a dense group of cars on a part of the route will queue at the bottleneck. If the calculations state that they will not queue, the link is added to the *free speed part*, but if they will, one more calculation must be done. This time, the same basic Single Bottleneck calculation is done, but for the part of the route consisting of the links from the start of the route, up to the link in question; whose vehicles will make up the queue. Still, the same bottleneck is used. This calculation is done to see if that particular queue has vanished by the time the vehicle currently being guided has travelled the distance to the first vehicle of the dense group, which is the sum of the link lengths of that part of the route. Here, the first vehicle of that group is assumed to be at the end of the link that it is travelling.

Again, if the queue is predicted to have dissolved, we continue to check the next link. After a while, the entire route will belong to the *free speed part*, or there will be a *free speed part* and a *queuing part*. These are added to make up the prediction for the whole route.

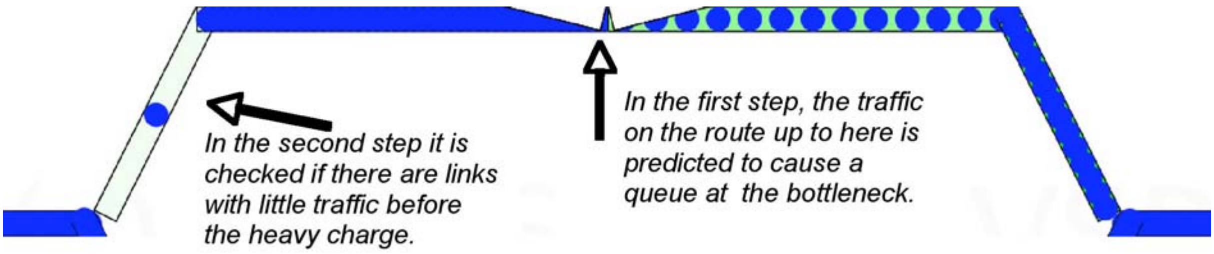

Figure 14. The two steps in the algorithm of the Distribution model.

3.4.3 Evaluation: Prediction abilities

When using the standard test population the problems described in the beginning of this chapter do not occur because the agents come in a steady flow. To evaluate the distribution model a new population was created which generates a steady base flow but also pulses that arise in reality, for example, because many agents leave home at the same time. Another example this can represent is the influence of traffic lights. In this evaluation scenario, the steady base flow is lower than it was with the normal population used for the basic SB evaluations, but pulses of heavy traffic comes on both routes approximately every ten minutes.

First, the new population was guided using the regular SB model. The graphs of the predictions of the travel times on both routes exemplify the problem faced by the basic Single Bottleneck model. The guiding is activated.

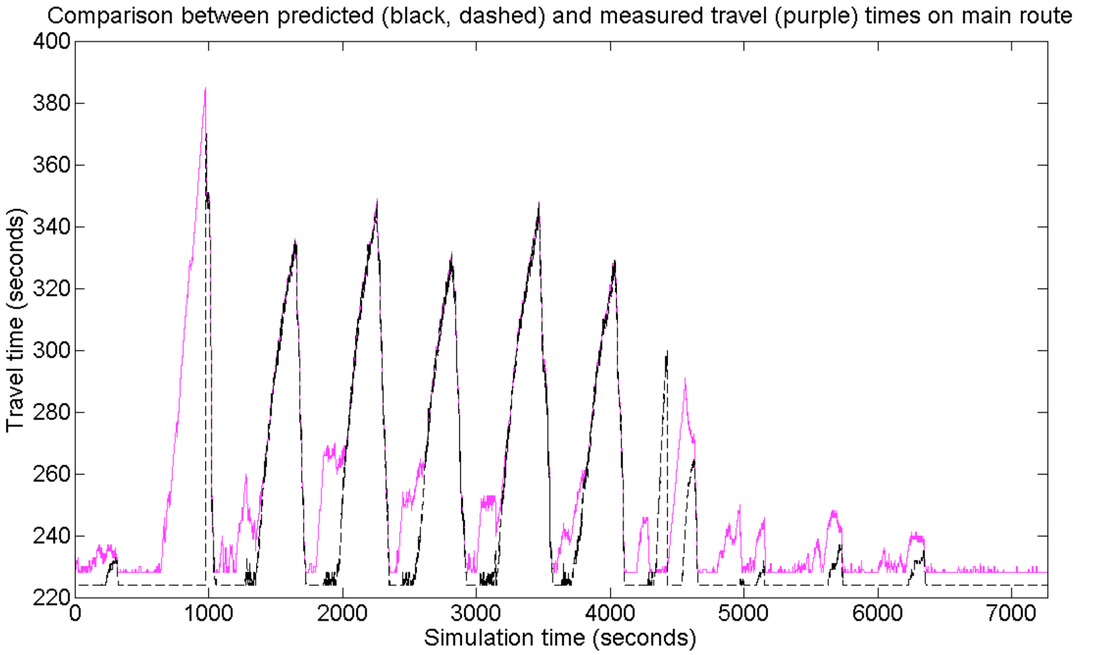

Figure 15: Main route predictions with basic SB-control of pulsating traffic.

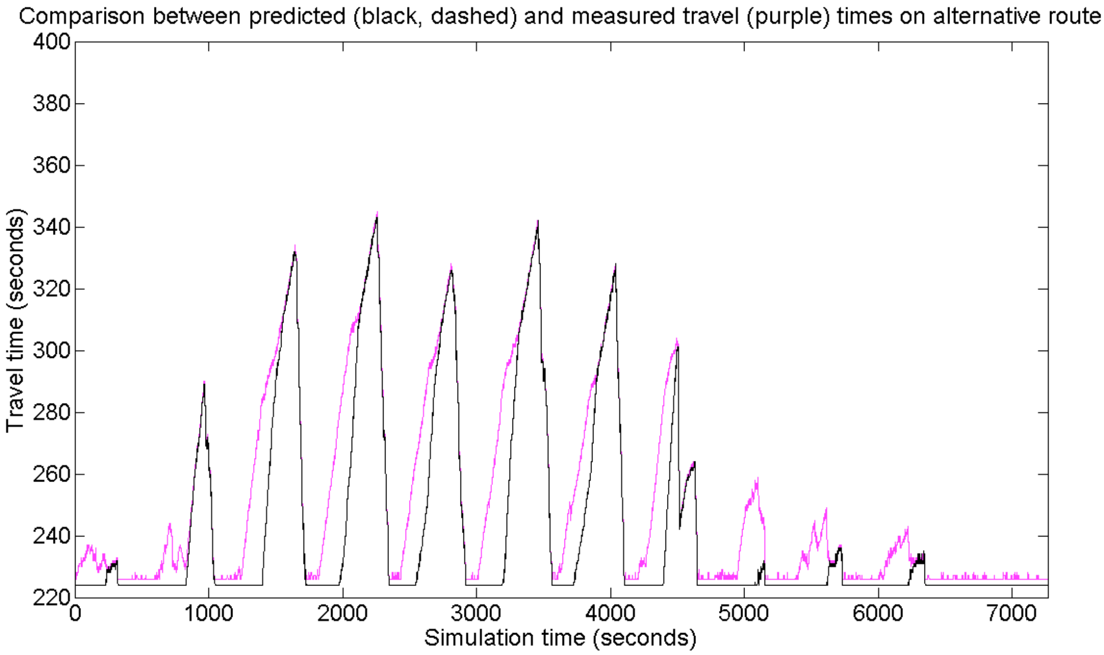

Figure 16: Alternative route predictions with basic SB-control of pulsating traffic.

In figures 16 and 17 above, we can see how the travel times increases every time a pulse of traffic comes on the route. It is obvious that the SB model is too late, predicting too low when a pulse comes. In these moments the traffic situation looks something like in figure 14.

Now, let us see what the predictions look like using the Distribution model.

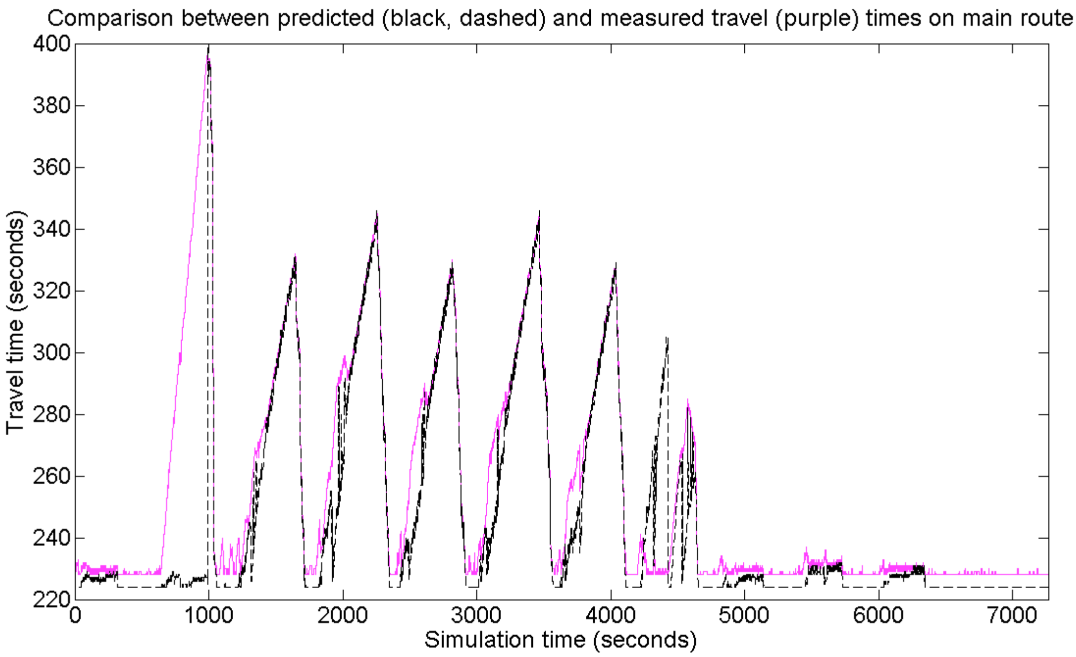

Figure 17: Alternative route predictions using Distribution Model on pulsating traffic.

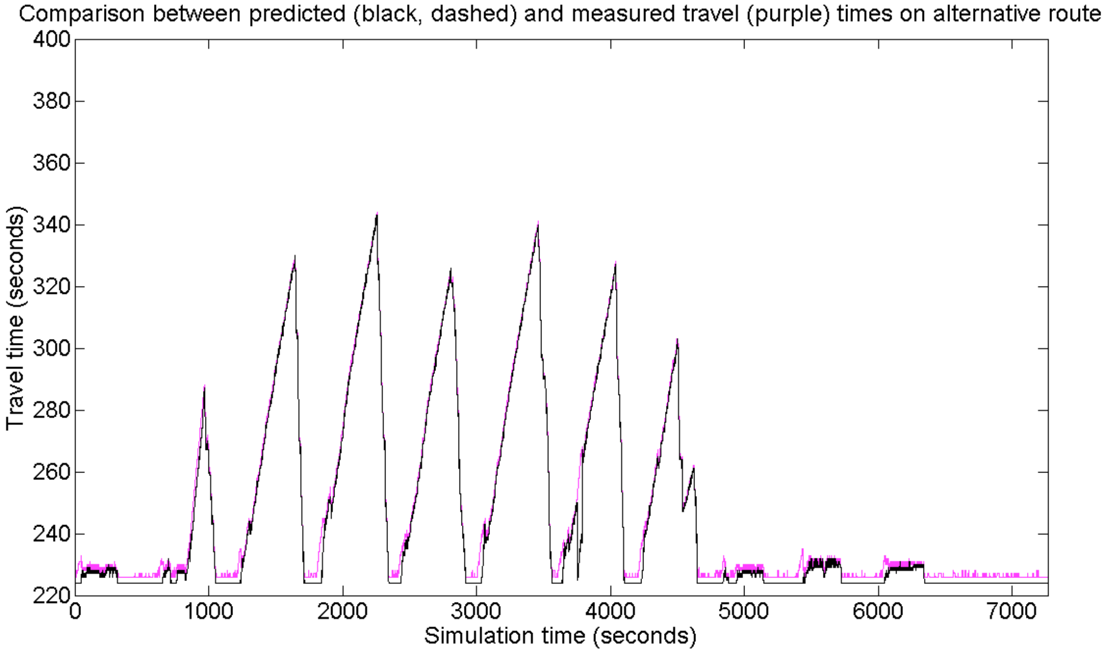

Figure 18: Main route predictions using Distribution Model on pulsating traffic.

The results are good. This model predicts the travel times much better. The prediction errors are reduced on the main route and virtually eliminated on the alternative route. As shown by the figure 18 and figure 19 and in the evaluation table below, adding the distribution-check increased the fit slightly on the main route, and considerably on the alternative route. The conclusion is that with this feature in the prediction model pulsating traffic patterns will be handled better. And more reliable information about the travel times make way for better system performance when using feedback control.

Table 2: Evaluation table for the scenario with pulsating traffic.

	Normal Case	Accident No Control	Single Bottleneck Model	Distribution Model
Main route:				
TT	233	683	253	254
Agents	3998	3998	3354	3352
Fit	-	-	95.6	96.1
Alternative route:				
TT	232	232	259	255
Agents	3998	3998	4642	4644
Fit	-	-	96.3	99.1
Total				
Std. Nash deviation	7.2	425.8	19.4	21.9
Average TT	232	539	257	255

3.4.4 Evaluation: Travel time improvements

More accurate travel time prediction means a more adequate control signal which means decreased travel times for the agents – but only if there is potential for more effective route guidance in the system. So is the system performance improved when the distribution check is added?

According to the table, the distribution check improves the average travel time, but very slightly. The Nash deviation is a little bit worsened. The most probable reason for why the travel time only decrease by two seconds is that the usage of the routes' capacities are already close to the optimum. When a pulse of heavy traffic comes, the travel times increases on both routes, as shown in figures 16 and 17. But even with a standard SB-predicted output, the controller makes the best of the situation by letting the total travel times increase equally on both routes simultaneously. On the other hand, if the pulses would be only one of either route, the distribution check would detect them and compensate for the increased travel time on that route by redirecting a proportion of the traffic into the other route, whose capacity were not used fully. To create such a scenario in the test network, the agents in the pulses must not be object to route control, but must travel according to their own plan. Such agents are for example those who enter or leave the route between the sign and destination link. Then we would have a scenario in which the traffic flows cannot be regulated completely by the route guidance system.

The reason why the Nash values are worsened is probably the square factor of the standard Nash deviation calculations. As we can see in figure 20 below, the Nash Times for the Distribution Model has more peaks than the Single Bottleneck Model. One important thing to keep in mind is that the bang-bang controller does not care about the magnitude of the Nash difference; it works exclusively according to the sign of the difference. This means that improved model fit values does not necessarily imply improved travel times – the control signal, i.e. the guiding signal to the agents might be exactly the same. It should also be mentioned that small differences such as a few seconds are hard to analyze, since there might be micro queues or other noise affecting the difference between the models to a large extent. It is undeniable though, that better fit values would be statistically beneficial for the system in the long run.

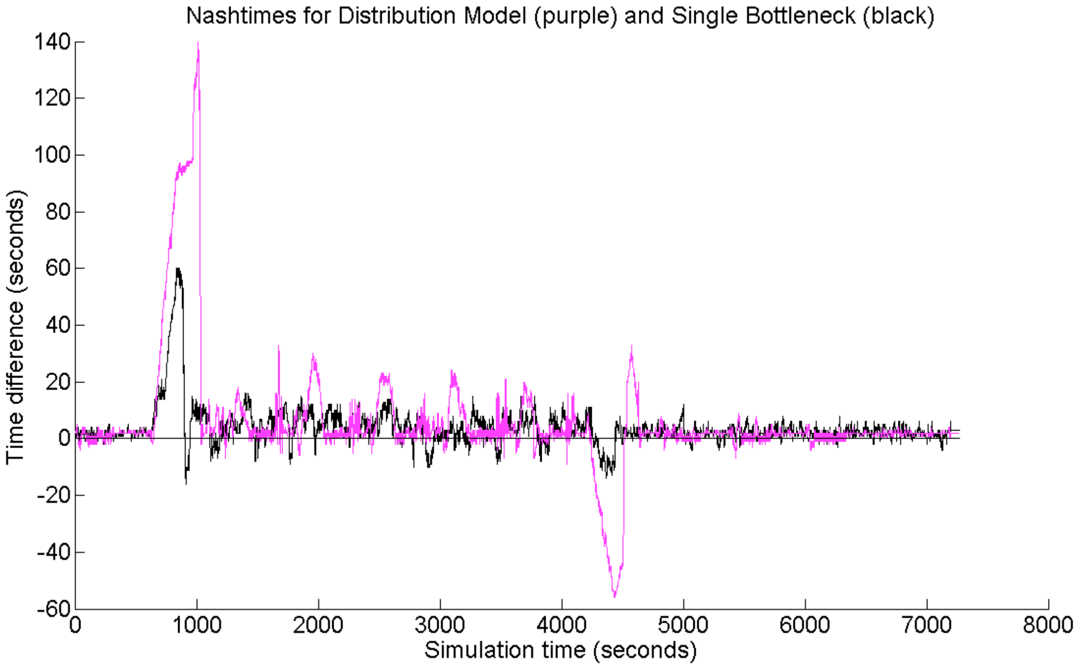

Figure 19: Nash time plot for the distribution model on a scenario with pulsating traffic.

3.5 Disturbance Model

Inflows and outflows of vehicles on a route can alter the traffic situation so that the travel time predicted with the regular SB model increases or decreases. An outflow of agents means that a guided vehicle might experience a travel time shorter than the predicted, whereas an inflow can make the route slower than predicted. If the majority of the agents on the route do not enter via the sign link (the link where the guidance is given), then the regular SB prediction will be based on very inadequate information. In order to compensate for such errors, the in- and outflows must be taken into account when counting the number of cars on the route.

The non-deterministic character of such in and outflows constitute a fundamental problem since they are immeasurable during the prediction time and we therefore cannot know for sure how many vehicles that actually have to pass the bottleneck before the one being guided. The information at hand when giving the guidance is how many cars that is presently on the links, and present and historical inflow and outflow data.

3.5.1 The Disturbance model

Our approach to simulate a continuous in- or outflow is to measure the flow, and multiply that with the time it takes for the guided vehicle to get to the intersection where the in- or outflows take place.

When this is done for all in- and outlinks, we calculate the total net change of agents caused by in- and outflows. The time factor is needed for calculating the number of agents that go on or off the route ahead of the agent now being guided, as a flow just before the bottleneck will result in a greater net change than a flow just after the sign link.

The true travel time to an intersection is obviously not known beforehand since there might be a queue starting at the bottleneck and stretching passed the intersection. However, to predict the queue length correctly, the net flows must be taken into account, which puts us in a complex situation. To keep things simple and avoid complex calculations, we decided to use the free speed travel time up to the intersections when calculating the net flows. The free speed travel time will indeed be the correct simulation time to use when there is no queue stretching back passed the intersection. As long as demand is not extremely high and the guiding works well, this is a reasonable approximation. In any case, the model will always compensate for additional traffic to a certain extent. When calculating the average flow values we use the free speed travel time on the route as time horizon.

Multiplying the additional flows with the travel time to each intersection, a net number of agents will be predicted to go off, or on the route ahead of the guided vehicle and thus the additional flows can be compensated for in the travel time prediction. The longer the queue gets, i.e. the more intersections the queue stretches passed, the worse the approximation will be. This is both because the travel time to the intersections will change and because the intersection dynamics will change when two flows interact, e.g. vehicles have to wait for each other.

3.5.2 Mathematical description

First, some notations are needed:

x_i number of agents on link *i*

f_i flow to link *i* (positive value means inflow)

tt_j free speed travel time for link *j*

$TT_{f.s.}$ free speed travel time for a route; Link 0 to link n

b bottleneck index

n index of last link

c_b bottleneck capacity (vehicles/second)

The total number of agents that have to pass the bottleneck before the guided vehicle, is given by the following formula:

$$X_{tot} = \sum_{i=0}^{b} x_i + \sum_{i=0}^{b} \left(f_i \cdot \sum_{j=0}^{i} tt_j \right)$$

This expresses the agents currently on the link (the first term) and the agents coming onto or going off the link calculated by the flows and the (free speed) travel time to the respective nodes (the second term). The queuing travel time is then:

$$TT_{queue} = \frac{X_{tot}}{c_b} + \sum_{i=b}^{n} tt_i$$

At last, this is compared with the free speed travel time for the route and the biggest value is taken as the prediction.

$$TT = \max\left(TT_{f.s.}; TT_{queue}\right)$$

3.5.3 Evaluation

To evaluate the Disturbance Model, a new population was created that included an extra inflow to the main route before the accident location.[5] Vehicles coming onto the route there are not controlled by the route guidance but have to be compensated for in the model. One could also imagine a test scenario where there is outflow instead of an inflow. In analogy with the results presented in this section, an outflow would cause the regular SB model to make too high estimates of the travel time, and the disturbance model would compensate for it in the same manner as for the inflow. The inflow and the outflow case are theoretically similar and we will here present only the results from the inflow scenario, which in contrast to the outflow results in more congestion. As mentioned earlier, the test network was designed for the model development, while the Berlin network is used to evaluate the model. To evaluate the predictive strength of the model, the basic Single Bottleneck model is first run on the same population and the results are shown in figure 21.

[5] The test population is called PlansInflow. It is like PlansNormal but with an additional inflow on link 31 of 500 agents per hour.

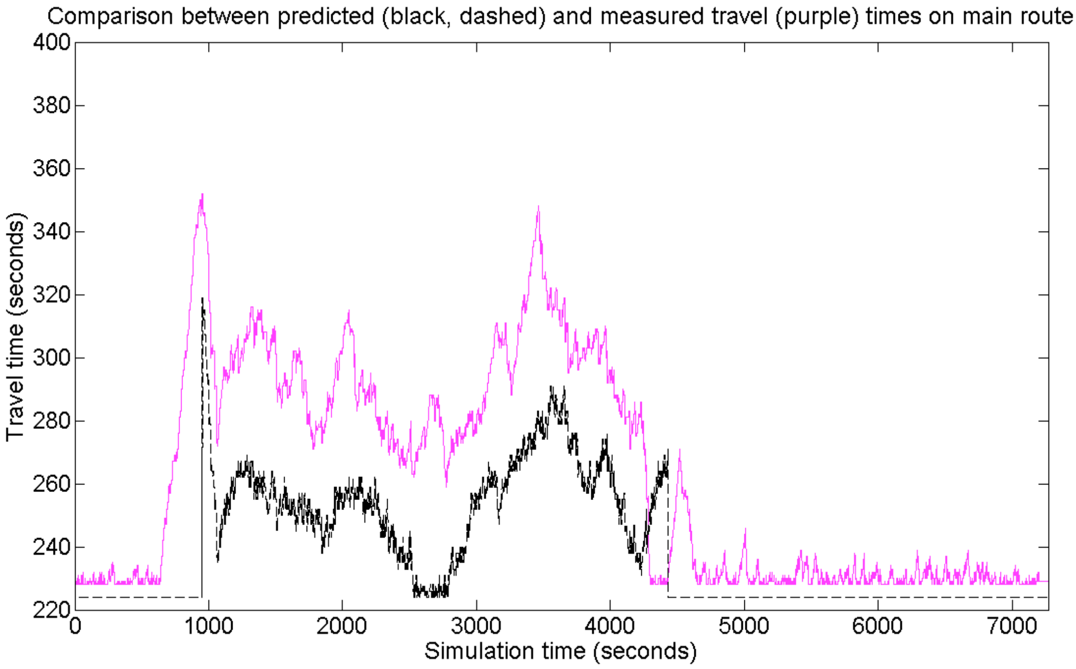

Figure 20: Measured and SB-predicted travel times on the main route when there is an additional inflow.

It is obvious that the standard SB model underestimates the travel times since it does not know about the extra inflow that must pass the accident before the guided car. With the disturbance compensation switched on the predictions improve, as seen in figure 22.

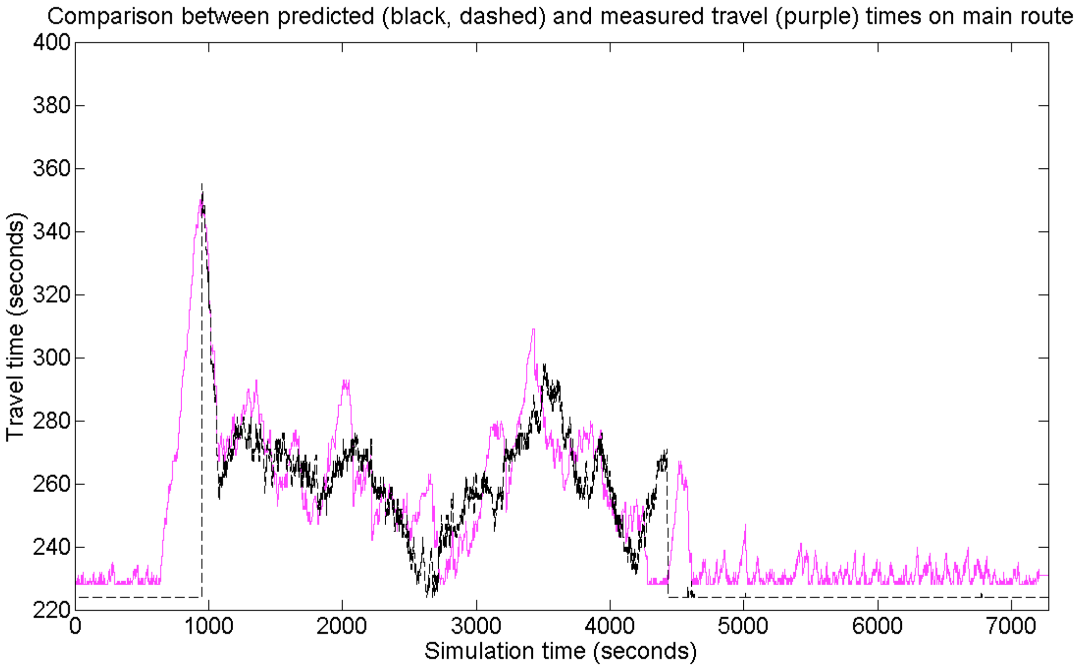

Figure 21: Measured and Disturbance-predicted travel times on the main route when there is an additional inflow.

Table 1. Evaluation table for disturbance model. Population with extra inflow from link 31 on the test network.

	Normal Case	Accident, No Control	Single Bottleneck Model	Disturbance Model
Main route:				
TT	231	981	253	244
Agents	4000	4000	3002	2968
Fit	-	-	91.3	96.1
Alternative route:				
TT	227	228	242	245
Agents	3999	3999	4997	5031
Fit	-	-	98.9	99.1
Total				
Std. Nash deviation	5.7	839.2	33.6	20.7
Average TT	229	605	246	245

The model compensates for the inflow and improves the fit value from 91.3 to 96.1 on the main route (see table 3). The average travel time is not improved significantly. The reason is probably that the network is used in a close to optimal way, meaning that the capacities of the routes are used fully. Nash deviation, however, is reduced by 40 percent, which shows that we are much closer to the control goal than when the standard SB model supplies the input. This is also reflected by the travel time averages, which are almost identical when using the Disturbance model. The improvement in Nash times is clearly displayed in figure 23, where the Nash times for the Distribution Model and the Single Bottleneck Model are plotted simultaneously.

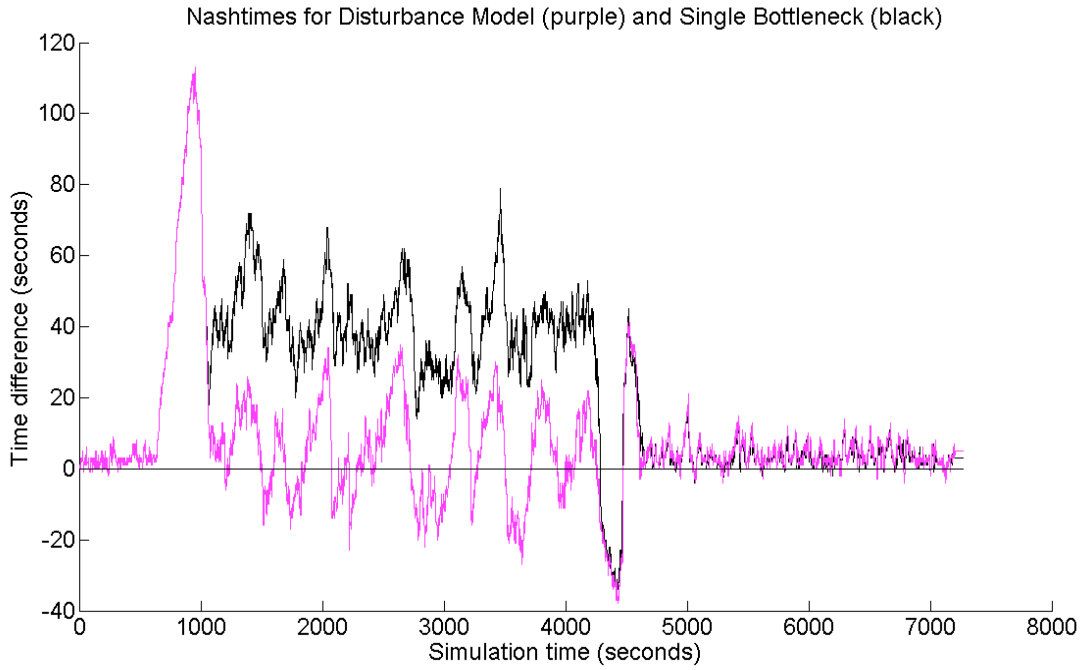

Figure 22: Nash times for the disturbance and the regular single bottleneck model, respectively.

3.6 Incident detection

The prediction approaches described use information about the accident; its location, its amplitude, and the time when it occurs. A traffic manager, who identifies the accident and reports its location and capacity reduction, can supply this information. This manual way of incident identification that is used today (for example in Berlin) is dependent of a fast and effective infrastructure that uses both human and technical resources. We have implemented a functionality in our prediction models that eliminates the need of manual detection of accidents and estimates capacity reductions. In turn it needs to know measured travel times to detect an accident on a link. This additional information could be e.g. given by a possible COOPERS backchannel.

3.6.1 The detection model

To detect an accident, we must continuously search the route for congestions. When there is a queue the travel time on a link is unusually high. The algorithm starts in the end of the route, checking if the last agent leaving a link had a measured travel time longer than the free speed travel time. An unusually slow link means that a bottleneck is found. The measured outflow on the detected bottleneck link is then used for the travel time prediction, instead of the normal capacity of the link. Just like the incident detection, the measuring of flows uses information about how often agents leave a link. The procedure is described by the following pseudo code.

3.6.2 Mathematical description

The notations used in the expressions are described below. The sums' index refers to the sequence of links on the route, where $i = 0$ refers the first link on the route.

y_i observed travel time on link *i*

t_{IQT} ignored queuing time on a link

tt_i free speed travel time for link *i*

x_i number of agents on link *i*

b index of the bottleneck link (the link where the accident has occurred)

n index of the last link on the route

lt_j registered leave time for the *j* last agent

f_b observed flow through bottleneck (vehicles/second)

To identify the bottleneck we check the following inequality for all links on the route, starting in the end of the route and going upstream.

$$y_i > tt_i + t_{IQT}$$

If the measured travel time on a link is longer than the free speed travel time + t_{IQT} seconds, the model has found a capacity reduction. There can be congested links upstream of the detected bottleneck link – such queues are assumed to be due to spill back from the queue at the bottleneck.

It is normal that agents take a little longer time than the free speed travel time to travel a link, also when there is full capacity. When the traffic density is high, but still under congestion level, the agents travel closely and sometime have to wait for each other or lower their speed a little. The constant t_{IQT} makes the detection less sensitive to this kind of noise. In the test network we ignore additional travel time up to 20 seconds.[6] The flow out from the link is then calculated based on the rate, with which the *r* last agents left the link:

---

[6] We started out having a factor $y_i > tt_i \cdot x_{IQT}$ where x_{IQT} = 1.20 would imply that travel times up to 20 percent longer than the free speed time would be ignored. On very short links (e.g. link 14 where the accident occurs) micro-queues of less than a second was considered a sign of congestion, and for this reason the factor was replaced by a constant.

$$f_b = \frac{\sum_{j=1}^{r} lt_j}{r}$$

By default the last 20 measurements are used. A higher number means a higher historical resolution with a more exact value, but at the same time a greater momentum and slower reflection of changes in the flow. After the bottleneck is found, the travel time is predicted as usual:

$$TT_{queue} = \frac{\sum_{i=0}^{b} x_i}{f_b} + \sum_{i=b+1}^{n} tt_i$$

$$TT = \max\left(TT_{fs}; TT_{queue}\right)$$

$$\hat{y} = TT_{main} - TT_{alt}$$

In a real world implementation the supply of travel time data for all links on the routes would mean either a quite extensive infrastructure of sensors or COOPERS-devices sending this data. However, for a sufficiently reliable measurement of links flows, it would be enough with a not too small proportion of the agents having a COOPERS-device installed.

3.6.3 Evaluation

In contrast to when evaluating the prediction models described previously, one could expect neither the prediction ability nor the travel times to improve in this case. After all it is a handicap not having immediate and totally reliable information about the accident. But on the other hand it is not reasonable to assume perfect information about accidents, as the other models do.

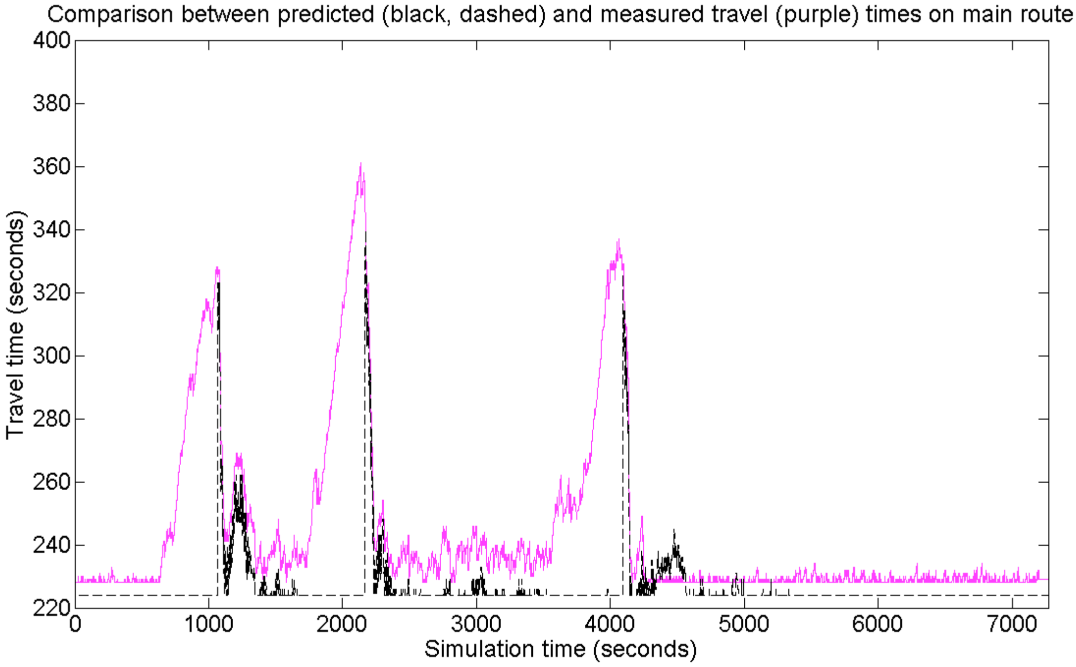

Figure 23: Travel times on main route with automatic incident detection.

There are three obvious places with prediction errors in the travel time plot in figure 24. When an accident occurs it takes some time before it will be detected. The constant t_{IQT} = 20 seconds corresponds to a small part of the total time delay. It can take up to ten minutes before the guidance react to the accident in the test network, which corresponds to the point where the observed travel times start to rise, to when the queue is reflected in the predicted travel times. These delays can be seen on three places in figure 24. When the bottleneck is detected, agents on the sign link will be getting a control signal that reflects the capacity reduction and vehicles will immediately be guided over to the other route. The long travel times in the plot represents agents who entered the main route before the accident occurred, and hence could not be helped by the route control. They will have to continue travelling the route that they started, even though an incident was detected by the traffic management system. This error is also to be seen in the main route travel time plots of the other models, but only once. In figure 24 the pattern appears three times. The reason is that the model “forgets” the accident. When an incident is detected the model uses this link as bottleneck for the travel time predictions for a certain period defined by a user parameter. The accident must be forgotten – or else the model will use the outflow as capacity even after the road is restored to normal, and it would not be able to detect other incidents. In the simulation that generated the plot above the model was set to remember the accident for 30 minutes and then again use the normal capacity. If the bottleneck had to be detected more often, there would be more large peaks than three in the graph. In a real traffic network, the choice of the reset time would not have such a big impact. When there is an incident on the main route – which is normally the fastest route – this route will always have a queue when the route guidance is active to keep the system at Nash equilibrium. This means that there is no risk that the incident is forgotten unintentionally due to absence of travel time observations. Therefore, in the Berlin scenario evaluated later in this report, the model is set to forget the accident every second – there will still be measured travel times of queuing agents.

Another and more effective way of handling the forgetting problem is to have a traffic manager that reset the detection function when the road is cleared and its capacity is restored to normal. In contrast to when incidents occur, the restoration does not happen unexpectedly; hence resetting the model is trivial to do manually.

Apart from that the Incident detection model is a little slower than the regular SB when the accident starts and is forgotten, the predicted graph is also less smooth. This is due to inaccuracies in the flow measurements. If the average measured flow for the last *n* cars is used, every new car will have an influence on the flow measurements of *1/n* parts. Thus, one has to weigh the disadvantages of noisy behaviour with the advantages of measuring only the most recent cars.

Table 3: Evaluation table for the Incident detection model. [7]

	Single Bottleneck Model	SB with Incident Detection
Main route:		
TT	232	243
Agents	3405	3321
Fit	96.8	93.6
Alternative route:		
TT	229	230
Agents	4594	4678
Fit	98.3	98.0
Total		
Std. Nash deviation	8.6	31.2
Average TT	230	236

Table 4 informs that the fit value is 3 percent lower for the main route with the incident detection. This prediction error is a result of the extended time delay discussed above, and it also makes us lead a few more agents into the accident route before the bottleneck is detected, which is reflected by in the increase of average travel time on the main route.

[7] The normal test population is used: 2000 agents/hour on each route. Accident: 50 percent capacity on link 14 from 7:15 to 8:15.

3.7 Multi Bottleneck Model

When running the Berlin scenarios, we realized that the travel times were predicted well when there was only one queue. When more than one queue appeared, due to inflows and natural bottlenecks in the network, our models predicted too low, at least if the first queue is the more severe one. When the last queue on a route is the more severe, then its queuing time will swallow the other queuing time fully or to some extent. For developing a model that could handle this, the situation below was created on the test network. An accident is set on the small link 11, causing a queue there. Another accident is set on link 14, and due to the inflow from link 31, a queue builds there as well.

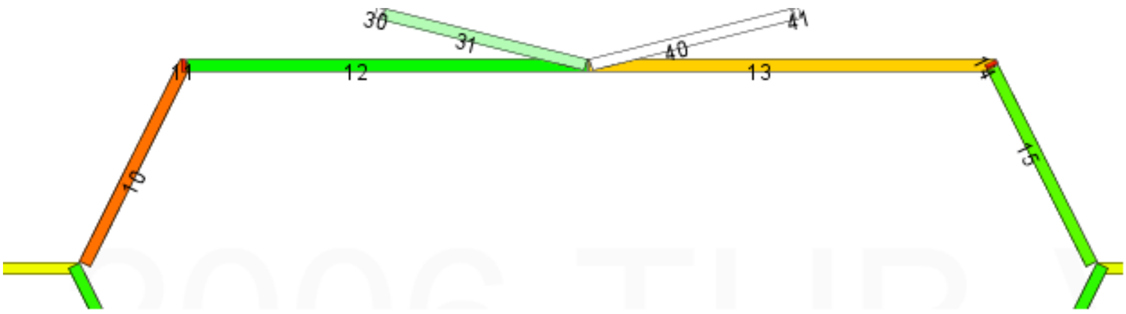

Figure 24: Scenario for developing the Multi Bottleneck Model. Accidents are reducing the capacity on link 14 to 77 percent and on link 11 to 63 percent. Test population: PlansInflow (as PlansNormal but with additional inflow on link 31 of 500 agents per hour).

It can be seen from figure 25 that the queue on link number 11 and number 10 is denser than the one on link number 14 and 13. Green colour represents free flow conditions and then there is a scale from yellow to red representing worsened conditions. Thus should all previous models underestimate the travel times on this route since only one of the bottlenecks is considered (link 14).

3.7.1 The Multi Bottleneck Model

The Multi Bottleneck Model divides the routes into segments, where every segment has its bottleneck. First, the incident detection procedure is done for the whole route and all bottleneck capacities where a queue has built up are stored. Then all flows are stored; flows on and off the route, but also flows on the route, i.e. from one segment to the next. The travel time for each segment is done as in the previous models; the only function not implemented is the distribution check (the problem with non-homogeneous traffic distribution is reduced partly by the route segmentation performed by Multi Bottleneck model). For the first segment it is done as before, but for the following segments, the number of agents ahead of the guiding object when reaching that segment must be simulated by assuming the flows to be constant during the time it takes to get there. Simulating/predicting the number of agents on every segment by the time the guided agent gets there allows the model to treat all segments with the same queue check approach. If there is no bottleneck on the last link of a route, all the links after the last bottleneck will be treated as free speed parts as in the previous models.

3.7.2 Mathematical description

The main calculation is to determine the total number of agents, X_{tot}^{k} going through the bottleneck on segment *k* before the agent currently being guided. This can be expressed mathematically as:

$$X_{tot}^{k} = \sum_{i=s}^{b} x_i + (F_s - F_b) \cdot \sum_{j=1}^{k-1} STT_j + \sum_{i=s}^{b} \left(f_i \cdot \left(\sum_{j=1}^{k-1} STT_j + \sum_{m=s}^{i} tt_m \right) \right)$$

Where the following notation is used:

x_i number of agents on link i at t_0 (start of simulation)

F_s Intra flow onto link s at t_0

F_b Intra flow out from link b at t_0

STT_j travel time for segment j

f_i in or outgoing additional flows to/from link i

tt_m free speed travel time for link m

b bottleneck link index of the segment

s first link of the segment

k index of the calculated segment

f_{bk} observed flow through bottleneck on segment k

The first term expresses the agents on the link at time t_0. The second term represents the agents coming onto the segment from the previous segment subtracted with the agents leaving the segment during the time it takes for the guided agent to get to the beginning of the segment. The last term is the disturbance compensation for agents coming onto the route from additional links. Those flows are assumed to be constant during the time it takes to get to the intersections, counted as the predicted time to the start of the segment plus the free speed time to that certain intersection (node).

After that calculation, the regular queuing travel time, TT_{queue}, is calculated, i.e. the simulated number of agents on the segment is divided by the segment's bottleneck capacity. At last, this is compared with the free speed travel time; $TT_{f.s.}$ for the route and the biggest value is taken as the prediction.

$$TT^{k}_{queue} = \frac{X^{k}_{tot}}{f_{bk}}$$

$$STT_k = \max\left(TT^{k}_{f.s.}; TT^{k}_{queue}\right)$$

These simulations and calculation are done for all *n* route segments, which add up to the predicted route travel time for the route. The predicted output is as usual the predicted time difference between the routes, and this value is sent to the controller that directs the agents.

$$TT_{route} = \sum_{k=0}^{n} STT_k$$

$$\hat{y} = TT_{main} - TT_{alt}$$

3.7.3 Evaluation

To show the predictive abilities of the Multi Bottleneck Model, we chose a scenario where the model predicts the travel times when no control is applied. This is because the model uses incident detection to discover the bottlenecks and that the incidents easily are forgotten as soon as the control is applied on the test network. This is not a problem on the Berlin scenario, as mentioned in the previous chapter. Figure 26 displays the main route predictions of the Single Bottleneck Model on this scenario. Incident detection is active, to make is more comparable with the multi bottleneck approach.

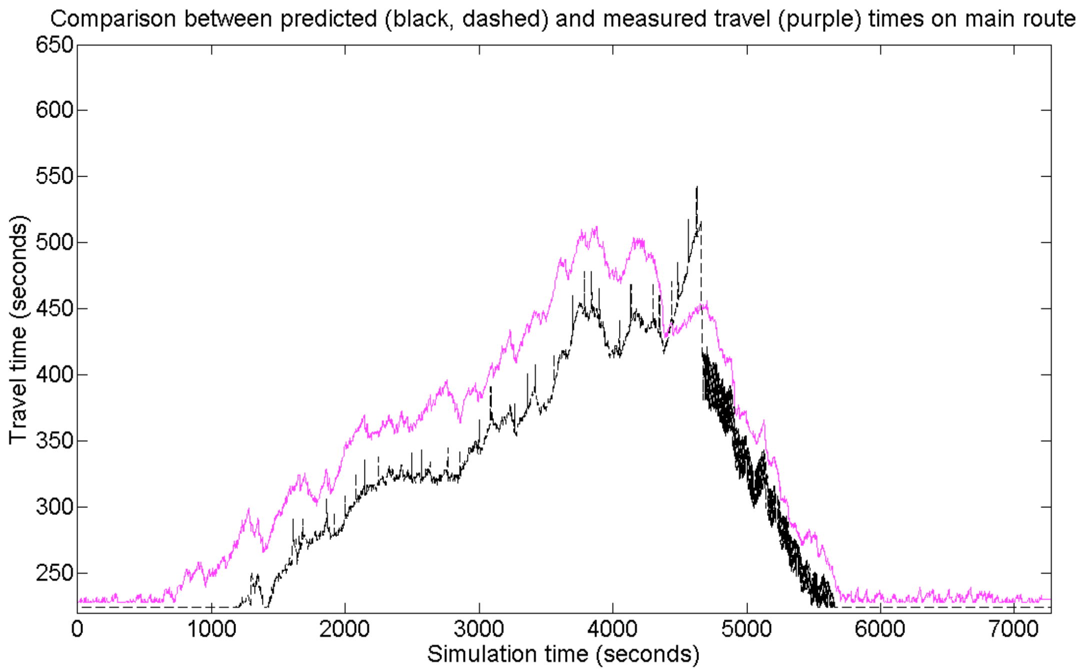

Figure 25: The regular SB with incident detection on a scenario with two queues.

The SB predicts to low in figure 26, which is expected since only one the queuing time from one of the two congestions is be included in the predictions. Since the basic SB model does not handle inflow traffic at all, figure 27 shows the predictions when the disturbance compensation was activated:

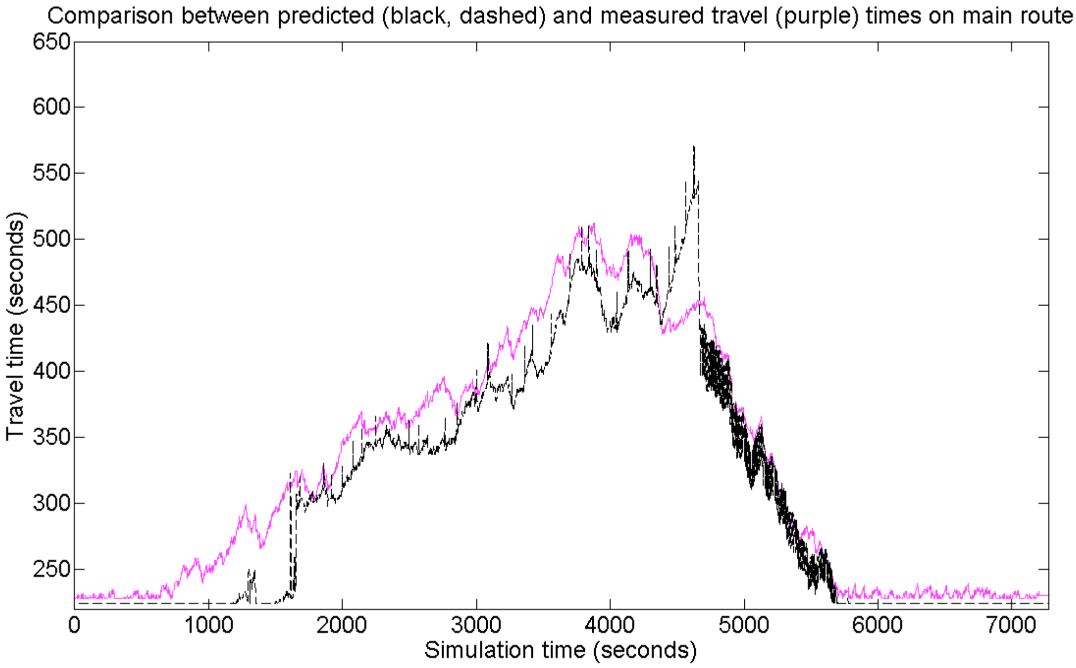

Figure 26: SB, incident detection and Disturbance compensation on a scenario with two queues.

The result is visibly better, but there is still an error, especially in the queue build-up phase, i.e. from 1000 seconds to 4500 seconds. Presumably, this is due to the limitations to one queue. Applying the Multi Bottleneck Model to the scenario, the results get as in figure 28.

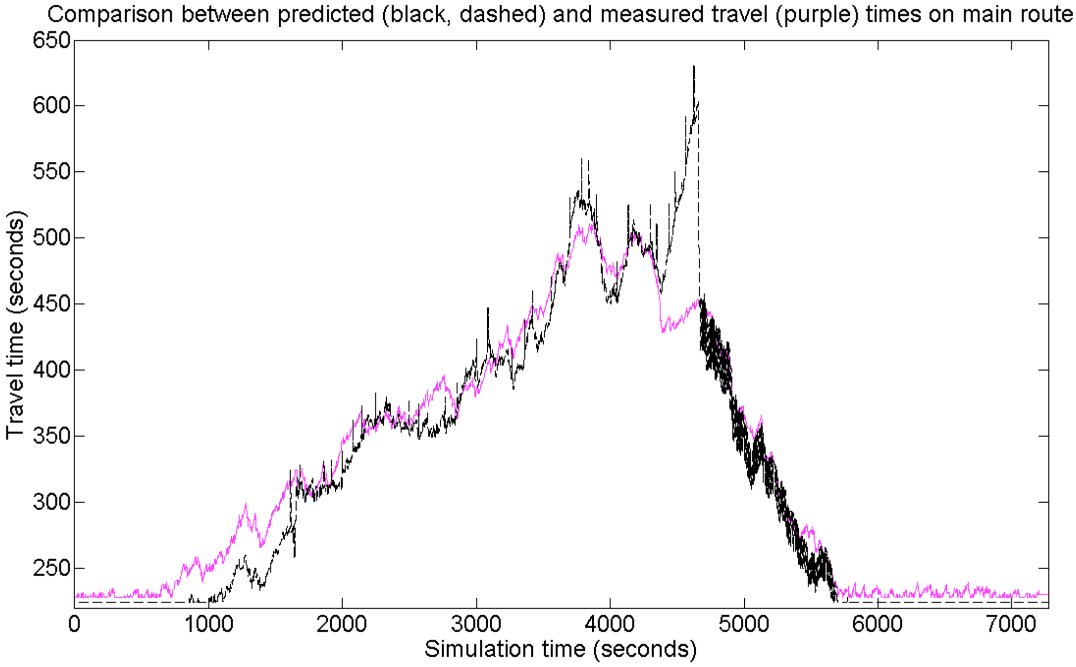

Figure 27: Multi Bottleneck model on a scenario with two queues.

Table 2. Evaluation table for multi-bottleneck prediction on the no control scenario. Prediction improvements for the main route.

	Single Bottleneck Model with Incident detection	Single Bottleneck Model with Disturbance compensation	Multi Bottleneck Model
Fit	91.2	93.6	95.4

The change in fit values between the models is significant. The result indicates that the Multi Bottleneck Model reduces prediction problems due to multiple congestions. Better prediction accuracy implies generally improved travel times, which will be evaluated in the part III of this report.

4 Testing the approach in a real world scenario: The Berlin simulations

Choosing a location in the Berlin network in MATSim and applying the predictive models and the guidance there was the next step in evaluating the models. A Berlin location constitutes a more complex system and the scenarios presented here are more similar to real world traffic than the scenarios in the test network (which were designed for implementing the models).

4.1 Merging the models

At this point the individual versions of the single-bottleneck model have been evaluated in the test network. To make it possible to run the model using several of the extra features at the same time, the java-classes for the individual models we merged into one class. This class has final class variables specifying what functions should be active. Another important motive for merging the classes were to get rid of redundant code that were common for all classes (Much redundancy had already been eliminated earlier when the super class `AbstractControlInputImpl` had been created.) Hence we are permitted to try the Single Bottleneck model in its most complete version. There are now only two model classes, the *Single Bottleneck Model* and the *Multi Bottleneck Model*. The different features of the Single Bottleneck Model can be tested separately though, by setting user parameters that switch the different functionalities on or off.

4.2 Scoring

MATSim has a scoring function that produces a score value in Euros for every agent according to how well the activities of the day were fulfilled. The scoring is calculated as the sum of the utilities of the *activities*, the (dis)utilities of the *travel,* and penalties for being late. The utility of an *activity* depends on the duration of the activity, waiting time, late arrivals and early departures. The (dis)utility of the travel is only dependent on the travel time (Charypar and Nagel, 2005), There is of a certain mathematical complexity behind this, but to simplify, one can say that longer travel times produce a lower score. In the absence of other effects, one additional hour of travel time lowers the score by 12 Euro per agent. Although it *is* possible to perform more sophisticated economic scoring with MATSim (Nagel et al., in press), including schedule delay effects and the like, that capability was not used for the study described here.

4.3 Subpopulation score

The score that MATSim produces is for the entire population. Since only a very small part of the agents in the Berlin simulations are actually affected by the guidance, the changes in the score due to the guiding are relatively small. Therefore, a scoring function that filters out a subpopulation was developed. The filter used in our simulations produces scores for all agents that passed both the sign link and the direction link during the whole day.

To find out how many agents have to be equipped with a COOPERS' device to get a benefit for all agents receiving the guidance we set up simulation runs with different compliance rates. E.g. a compliance rate of 50 % is equivalent with a scenario in which half of the travelers possess a device. The agents equipped with a device are randomly selected in each simulation run.

4.4 Evaluation of Travel Times

In this chapter the different models are discussed by the evaluation measures introduced in chapter 3.2. The Tegel highway connects the north-western suburbs of Berlin to the city centre and also to the ring road around Berlin (the “Berliner Ring”). An accident that reduces the highway capacity to 50 percent is simulated between 16.20 and 17.50, when many vehicles are leaving the centre and travelling north.[8] Figure 29 shows the traffic situation at 17:20 in the normal case and the accident case. In the accident case almost the whole main route is congested due to the accident, but the alternative route has unused capacity.

[8] A 50 percent capacity reduction can be interpreted as one lane is shut down on a two-lane highway, the other having full capacity. Another, perhaps more realistic interpretation is that one out of three lanes is shut down, leaving two passable lanes but not with free speed.

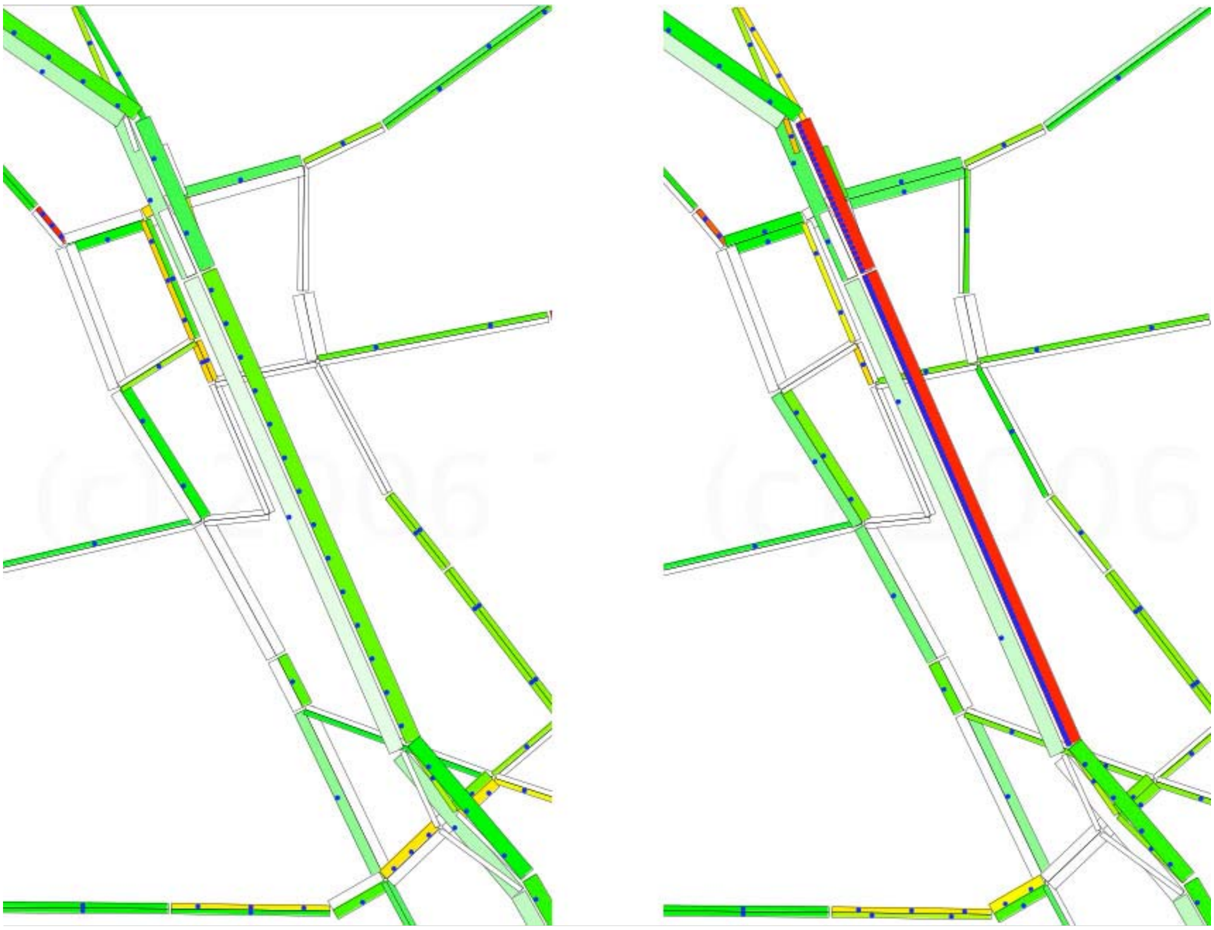

Figure 28. The no accident case (left) and the accident case (right) at 17:20. The accident causes a long queue on the highway, but the on the alternative route the demand is low.

The scenarios were simulated for a whole day. The evaluation values, except the score, are extracted only between 16.10 and 18.10, in order to examine the effect of the route guidance only around the time of the accident. The reason for the longer time after the accident is that there must be time for a queue to dissolve after the accident is gone, so that all agents that were queuing will be measured. Figure 30 shows the rising travel time caused by the accident. The average travel time for agents on both routes is 1004 seconds, compared to 367 seconds without accident. All the simulation results are presented later in this chapter in tables 5A and 5B.

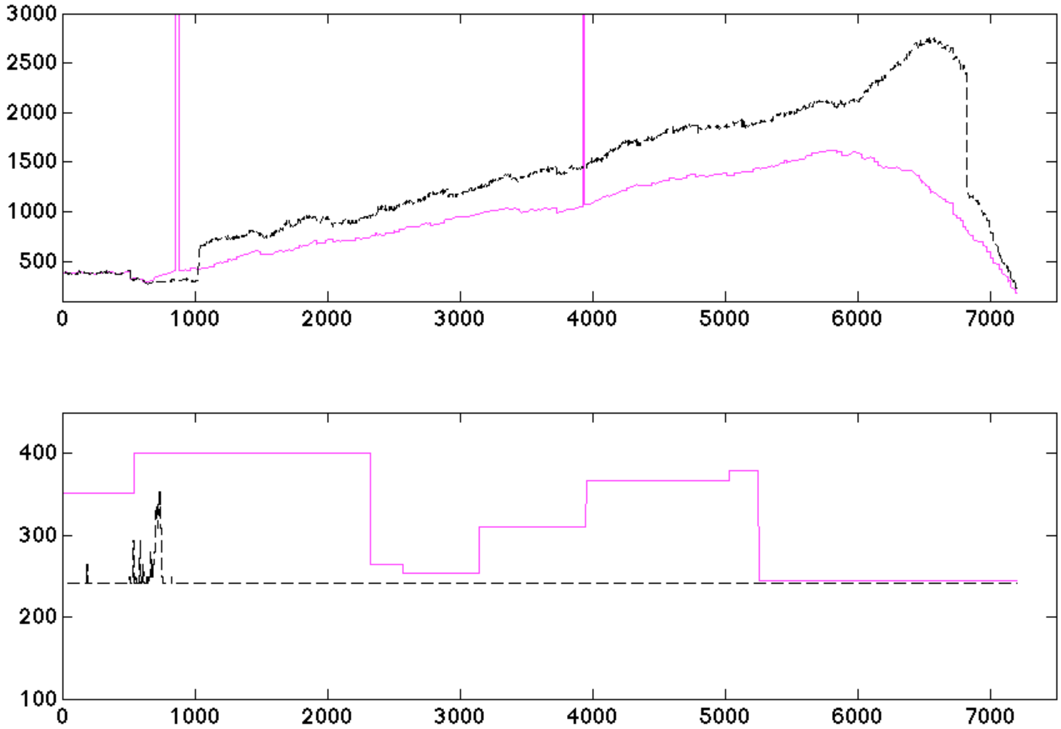

Figure 29. Travel times for the main route (above) and the alternative route (below) in the accident – no control scenario. Purple graphs are observations; black are values predicted by the Single Bottleneck Model. There are two outliers in the upper diagram; these are caused by vehicles that stop on the route for an activity, and thus look like a vehicle with a very long travel time along the route.

4.4.1 Reactive Control

We start out applying route guidance based on observed travel times. The result is displayed in figure 31. Thanks to the feedback control, some traffic is redirected into the alternative route, which lower s the overall average travel time from 1004 to 580 seconds. (The average travel time is based on all agents passing both the sign link and the destination link, i.e. all agents that travel one of the two routes in the regulated system. Also the travel times of agents that do not comply with the advice are taken into account.)

The characteristic system behaviour that were displayed in chapter 2 can be seen in figure 31, although natural variations in the demand makes the pattern less clear in this case. For example, at x = 4000 seconds, the traffic is guided into the main route, based on observed travel times of 400 seconds. In reality, the main route travel time is 800 seconds, which is much slower than the alternative route.

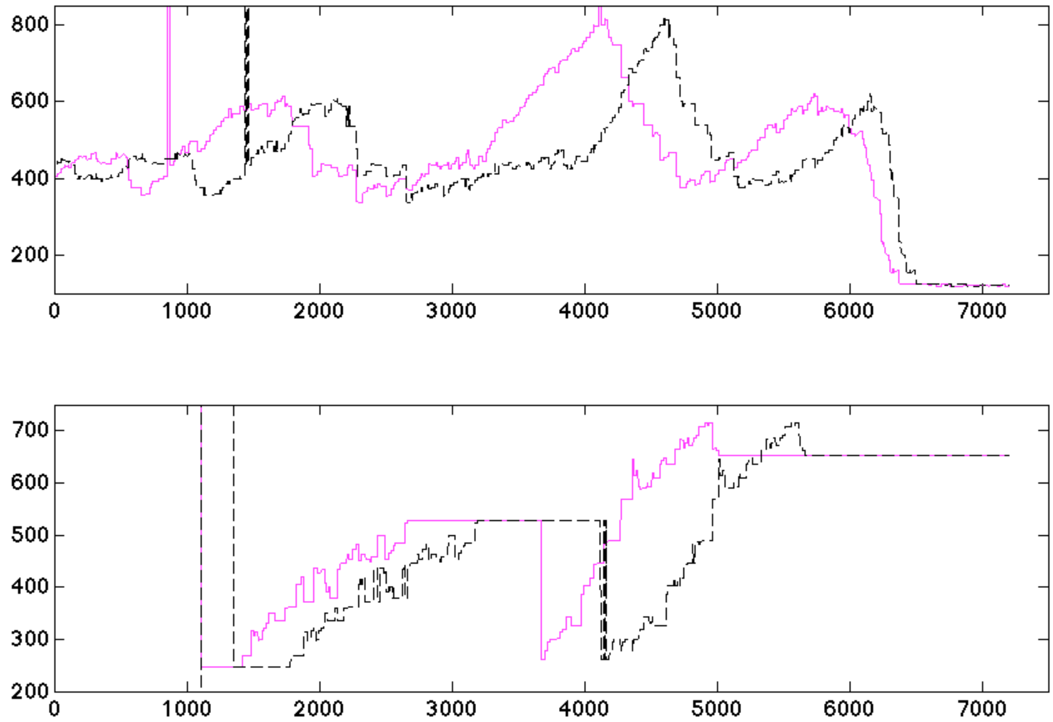

Figure 30. Route guidance based on observed travel times for the main route (above) and the alternative route (below). Purple represents the real value; black represents value used in feedback.

4.4.2 Basic Single Bottleneck Model

The system performance increases significantly when the guidance is based on predictive travel times. The destination link is now reached in averagely 349 seconds – even faster than in the no-accident-scenario without control (367 seconds). The traffic is distributed into the two routes, keeping the predicted travel times around 250 seconds on both routes and accordingly the predicted Nash time around zero. Figure 32 shows that the model generally predicts a little too high on the main route, but too low on the alternative route. A possible cause is the presence of several inflows on the alternative route, and one outflow on the main route. If this is the case one can expect an error reduction from disturbance compensation (evaluated below).

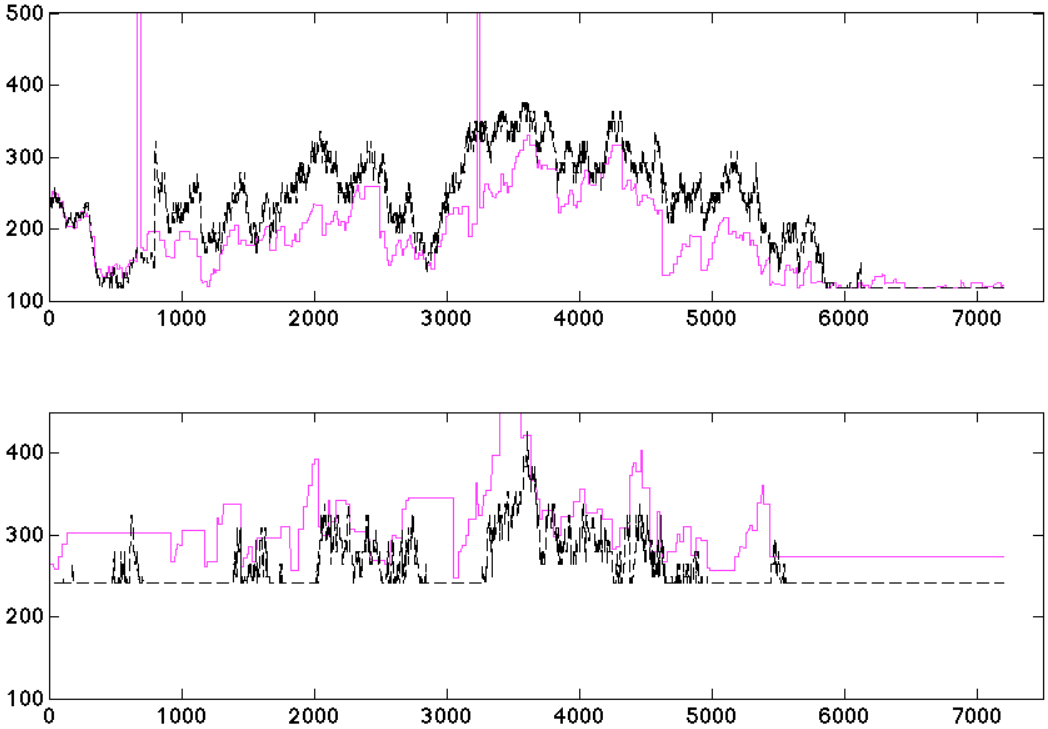

Figure 31. The Basic Single Bottleneck model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback.

Compared to when no control is applied, the number of agents taking the alternative route increased from 7 to 123 (remember that this corresponds to a increase from 70 to 1230 agents if the whole Berlin population would be simulated). The NetVis visualization to the right in figure 33 shows mostly green or yellow links in the controlled system, indicating that the demand is on an acceptable level. The situation is similar throughout the time with reduced capacity. Without the modifications made of some capacities, the slip road leading off the highway would have caused queues.

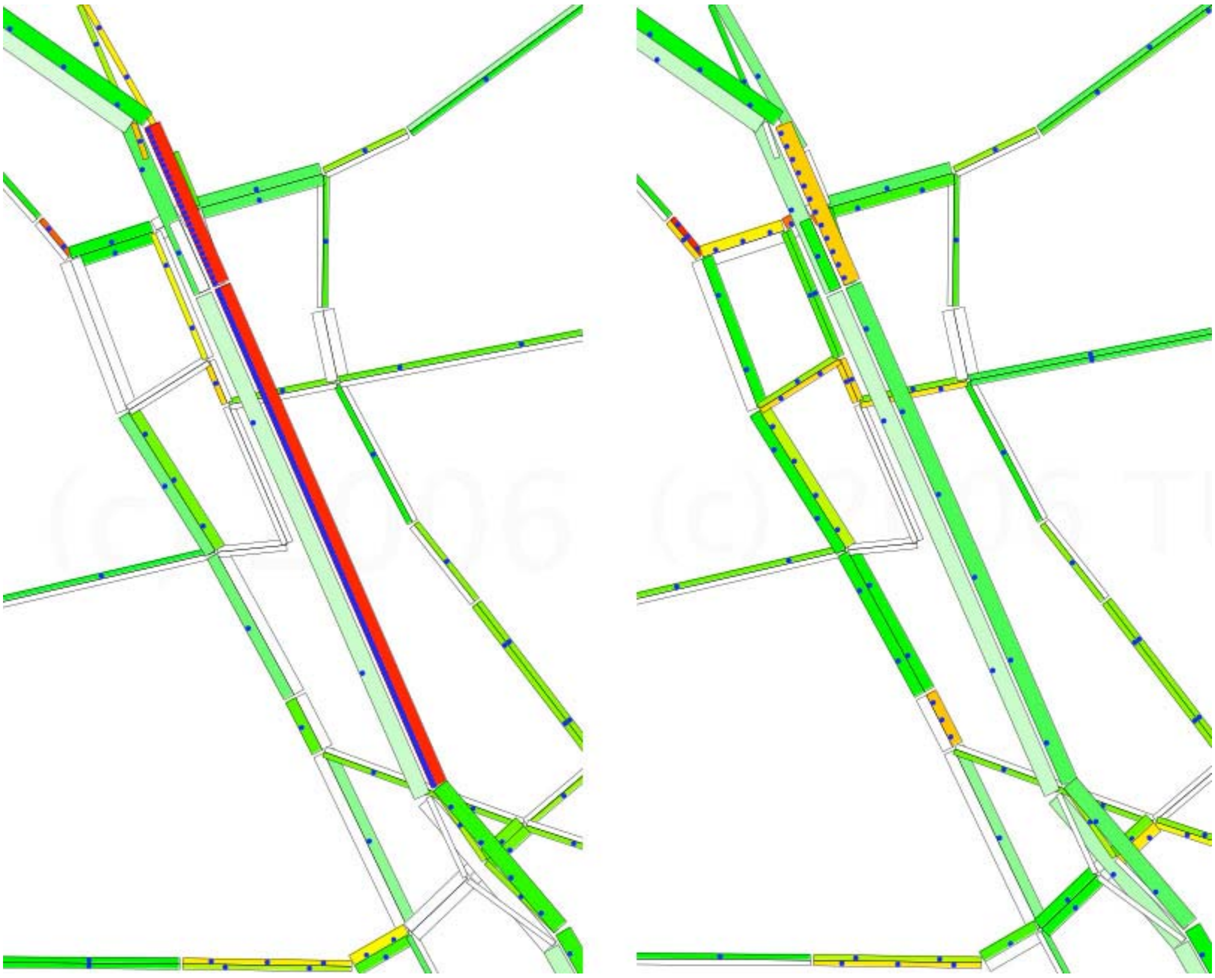

Figure 32. System at 17:20. Left: no control; right: control with Basic SB model. Compared to the open loop system, the queue on the highway is significantly reduced and the demand on the alternative route is considerably higher.

Apart from the benefits described above, the predictive feedback has positive effect also on the travel times before and after the accident. At 16:20, when the accident occurs, there is already heavy traffic on the northbound highway at Tegel. But as the traffic management system is active all day, some of the highway traffic has already been diverted onto the alternative route, which makes the initial conditions for the accident better than in the no-control scenario. Figure 34 visualizes the situation. In order to briefly estimate the benefits from predictive feedback control in ordinary traffic situation, the Basic SB model was evaluated also on a no-accident scenario, and the results are presented in the evaluation table 5A.

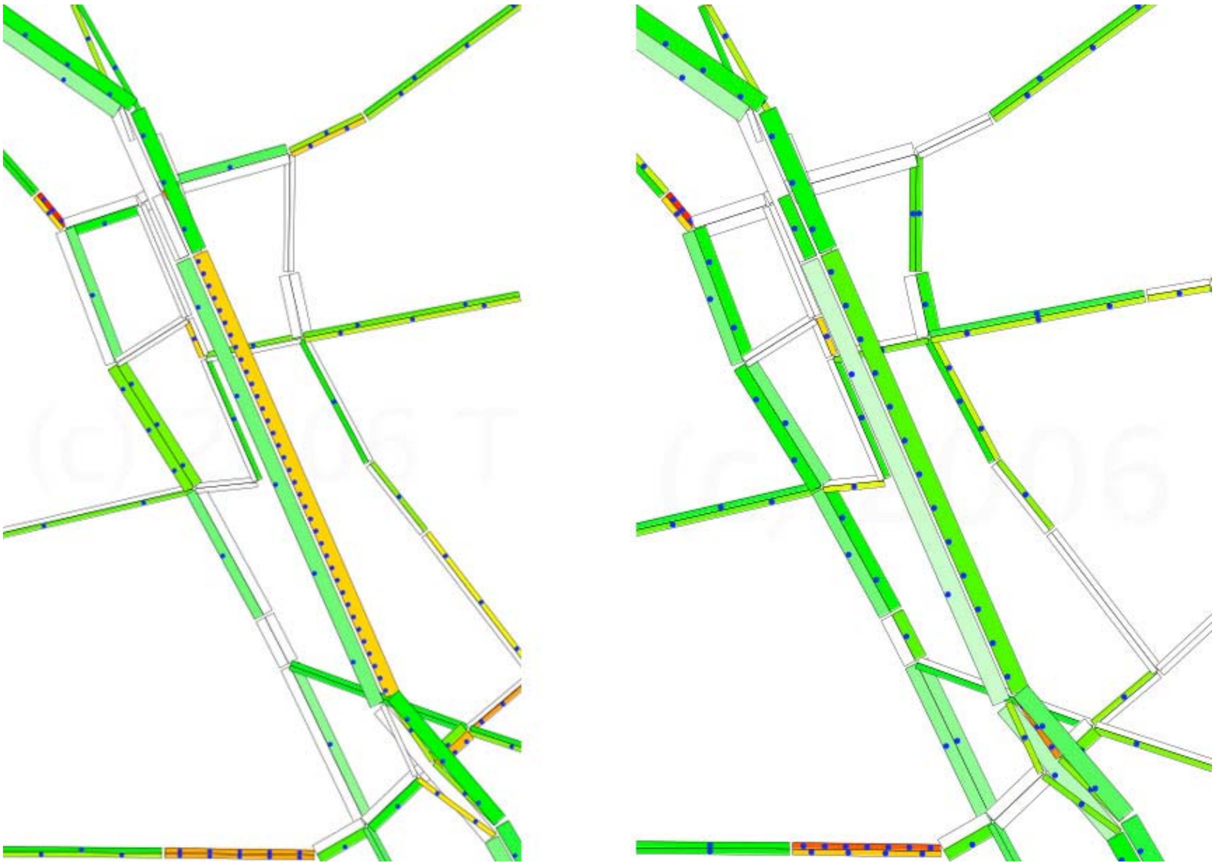

Figure 33. Traffic situation at 16:20 (just before the accident occurs). No control (left) and Basic SB Control (right).

4.4.3 Distribution model

Switching on the distribution functionality does not result in significantly shorter travel times (347 compared to 349 seconds without the distribution feature). Figure 35 presents the travel times. The Distribution model overestimates a little on travel times on the highway, just like the basic SB model. But the prediction errors are reduced on the alternative route, where the model fit increases from 85.0 to 90.5 percent. This contribution to the fit value comes from situations when the distribution model identifies traffic patterns on the alternative route when most of the traffic is located in the beginning of the route. The basic SB model assumption that the traffic will pass through the bottleneck in a steady pace would lead to underestimation of the travel times, but the distribution model recognizes that pattern and predicts a higher value than the free speed travel times. (See chapter 6 for a more detailed explanation of the problem with heterogeneous distribution of the traffic.)

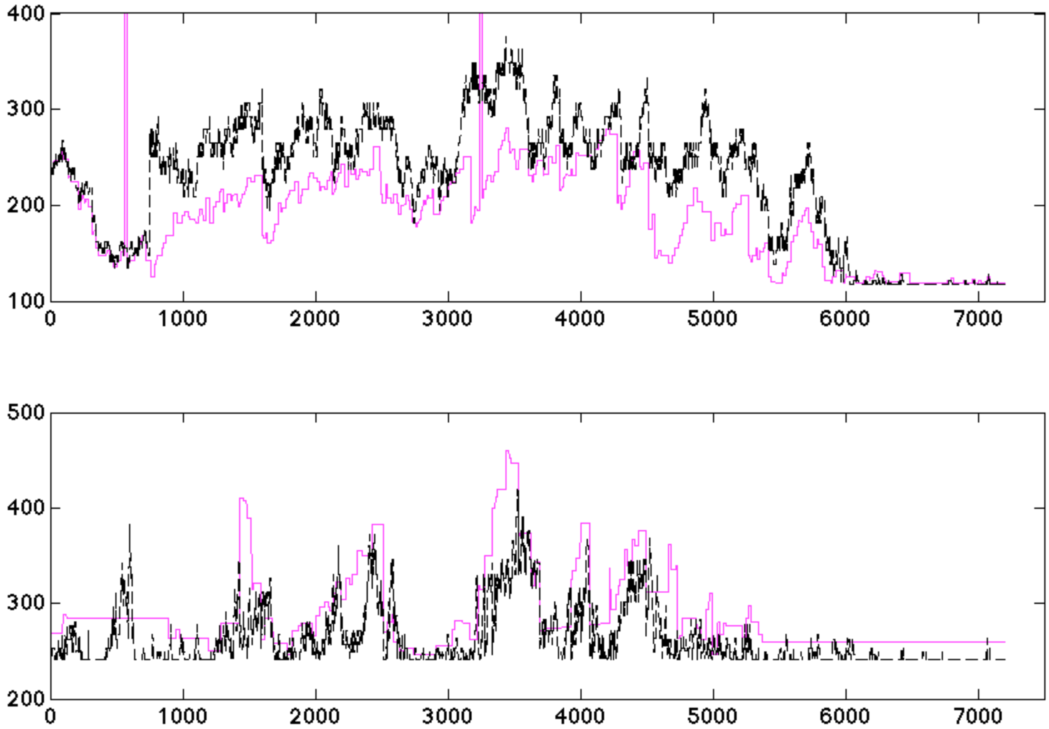

Figure 34. Distribution model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback.

4.4.4 Disturbance model

The evaluation of the basic SB model produced the hypothesis that the overestimation of travel times on the highway could possibly be due to an outflow upstream the bottleneck location. The evaluation results of the SB model with compensation of background traffic, does not confirm this hypothesis; see figure 36. The Disturbance model does no predict significantly better than the Basic SB model on neither route. In systems where there is proportionally many agents that travel only part of the route, predictions would certainly benefit from including such background traffic. The more intersections on a route, the higher is the influence on in- and outflows. The disturbance functionality should be evaluated further on another system.

In order to improve the prediction technique, the cause of the overestimation of the main route travel times in the Tegel scenario should be investigated. Useful information can possibly be derived from the fact that the models predict better when the highway capacity is not reduced. Figure 37 presents the travel times in the normal day, no accident case. The prediction errors are considerably smaller on the highway (model fit = 94.3 percent). There is a marked static error in the predictions on the alternative route, due to the insufficient basic data; only seven agents travel the complete alternative route during the time window evaluated.

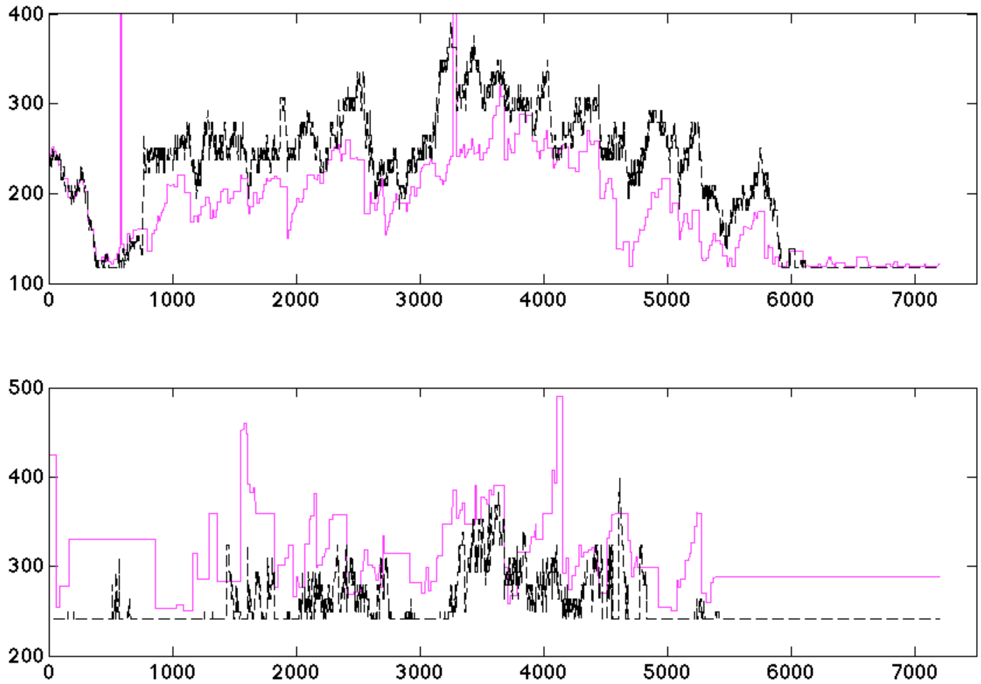

Figure 35. Disturbance model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback.

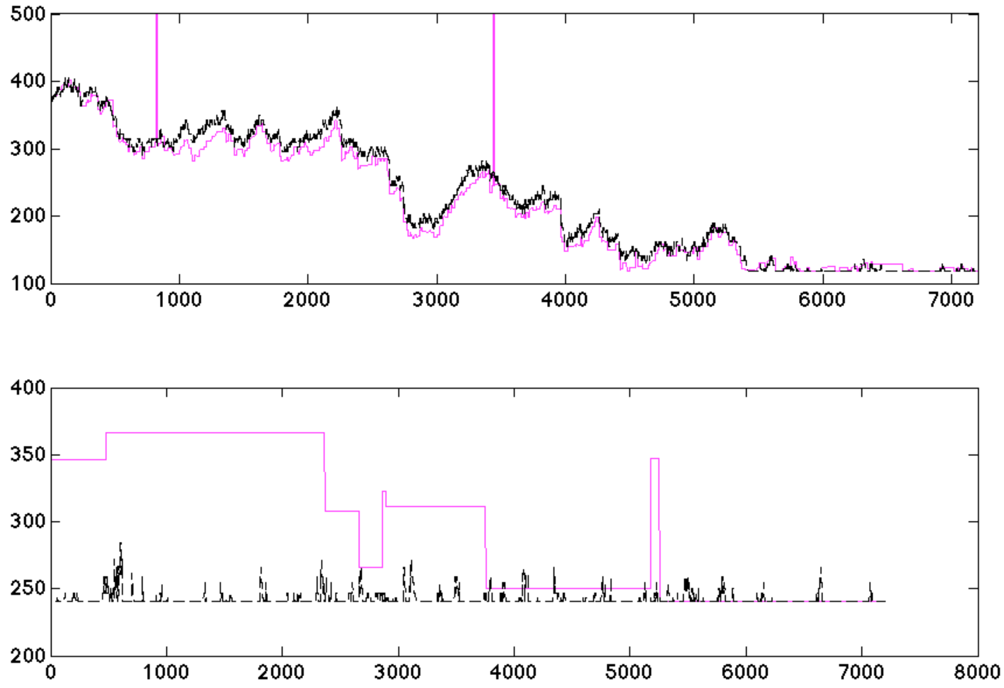

Figure 36. Prediction of the Basic SB model when the highway has normal capacity and is not subject to route guidance. Purple represents the real travel times; black represents predicted values.

4.4.5 Incident Detection

The Single Bottleneck model with automatic incident detection is the configuration that performs best of all models in the accident scenario, measured in terms of model fit on the highway route. This result was unexpected, as this model detect the bottleneck by measuring traffic flows, instead of beforehand information about when the accident occur and its amplitude. The highway travel times are 80 percent correct in average; the predictions are plotted in figure 38.

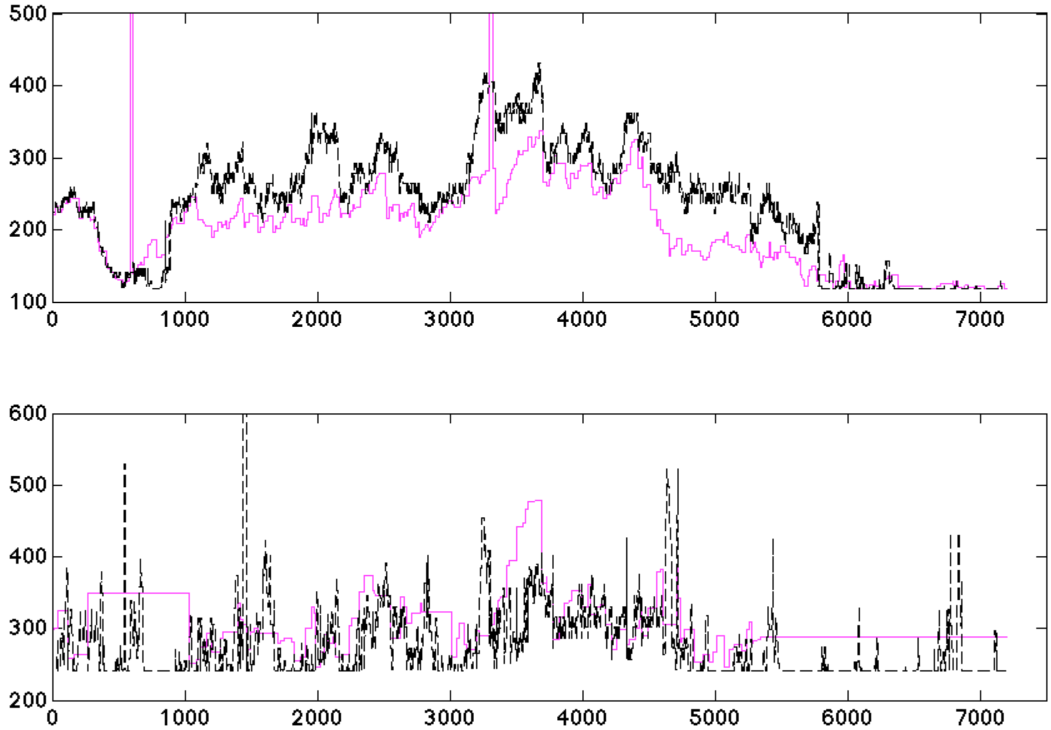

Figure 37. Incident detection. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback.

Shifting focus to the alternative route, the predictions seem to be considerable influenced by noise. The parameter t_{IQT} (ignored queuing time) discussed in chapter 8 is set to 20 seconds, like in the evaluations in the test network. As soon as the travel time on one link is 20 seconds longer than its free speed travel time, the link is considered to have an incident (or some other kind of temporary capacity reduction). That bottleneck will be the used in the travel time prediction. Choosing the parameter value is a trade-off between sensitivity and robustness. A longer ignored queuing time would result in less noisy predictions, whereas a lower value would slower detection of bottlenecks and more free speed travel time predictions. This parameter, just like the other model parameters in the additional SB features, deserves attention in order to optimize the prediction ability.

4.4.6 Single Bottleneck Model With All Features

Figure 39 shows the prediction ability of the Single Bottleneck Model when all when three additional features are active simultaneously. Of all models evaluated in this chapter, this combination is the one that is closet to reaching the control goal. Figure 40 shows the Nash times during the evaluation period. Prediction errors are reflected by the Nash time, but the main reason for the negative Nash time average is the travel time difference between the routes. The observed Nash time used as evaluation measure here corresponds to the Nash equilibrium only partially. The Nash time approaches zero when both routes are equally fast, and the predictions are correct. But the control goal is also achieved when one route is faster, and all agents take that route. This is the case between x = 6000 seconds and x = 7000 seconds in figure 39 and 40. To refine the Nash time as an evaluation measure, one could let *y = 0* represent the both cases of Nash equilibrium when calculation the total measure.

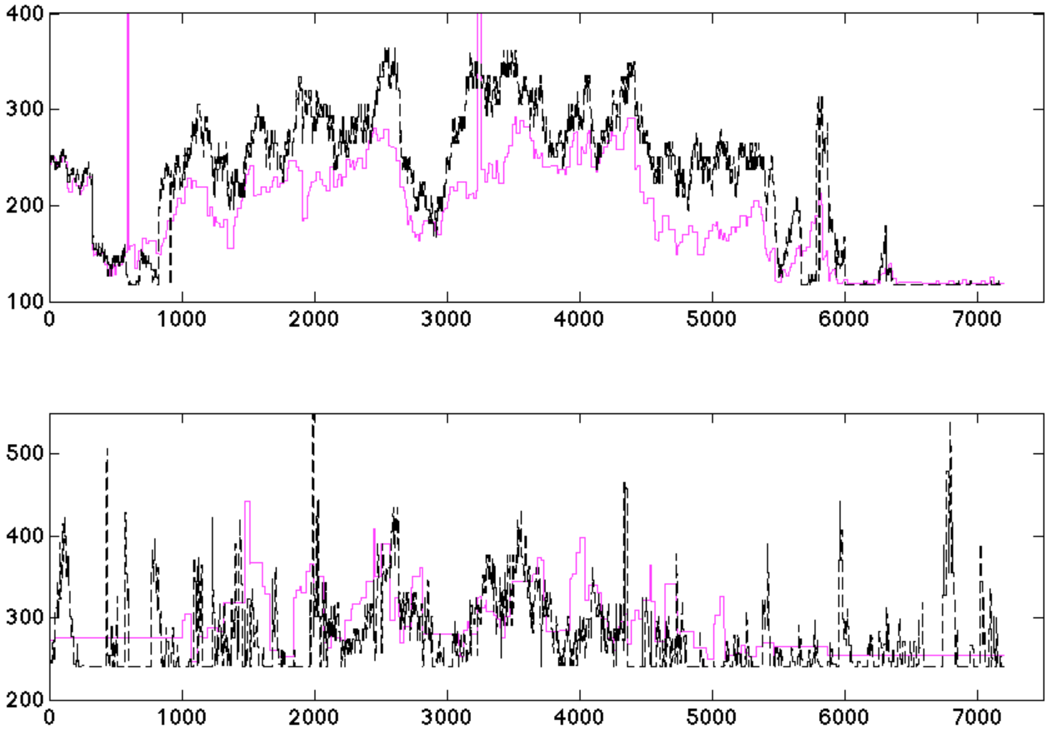

Figure 38. SB model with distribution, disturbance and incident detection functionality active. Purple represents the real travel times; black represents predicted value applied in feedback.

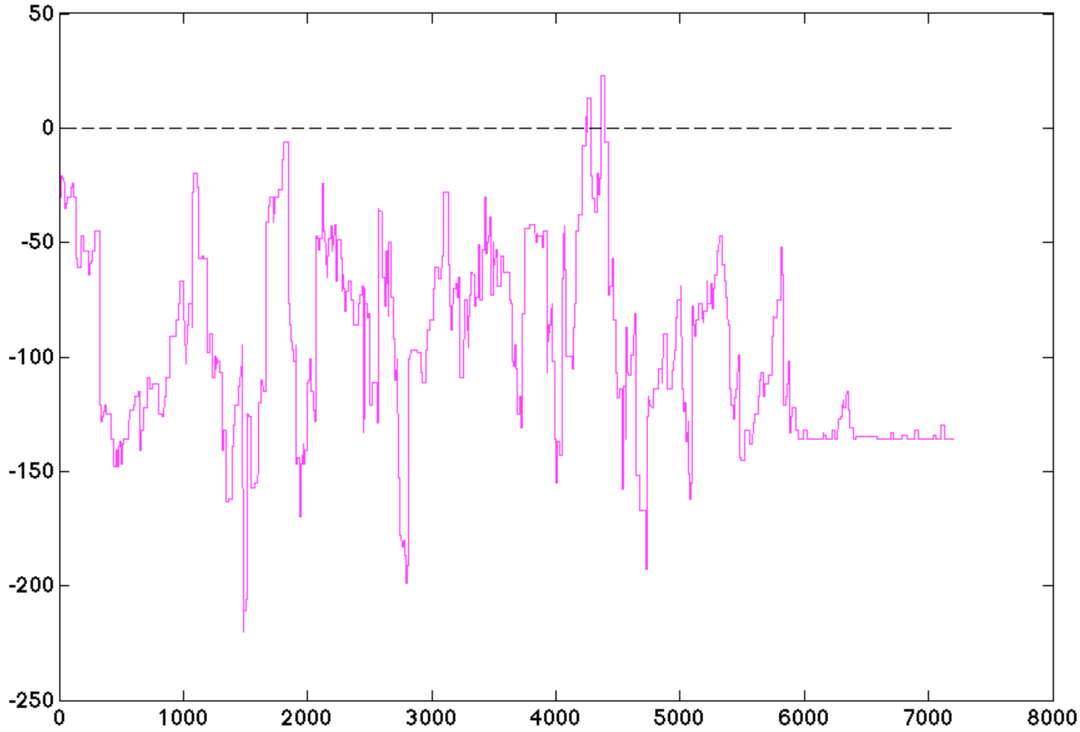

Figure 39. Observed Nash times for the accident scenario with the SB model with all features. The Nash times are centered around –100, which corresponds to the prediction error on the main route, subtracted with the prediction error on the alternative route.

4.4.7 Multi Bottleneck Model

The Multi Bottleneck Model was designed to handle the situation when there is more than one bottleneck influencing the travel times of a route. In the initial studies of the Tegel location, the alternative route was an example of such conditions. When the final simulations were carried out, however, no multi-queue pattern arisen on neither route. Therefore, the model cannot be evaluated here in a informative way. For this purpose, another and more suitable scenario should be simulated. Given the current conditions, the performance of MB model is best compared with the results of the SB model when all the features are active. Figure 40 and table 5B, (see the column “SB all features”) shows that the SB gives better predictions, as well as better system performance. Hence the MB Multi Bottleneck Model should be object for further evaluation and/or improvement.

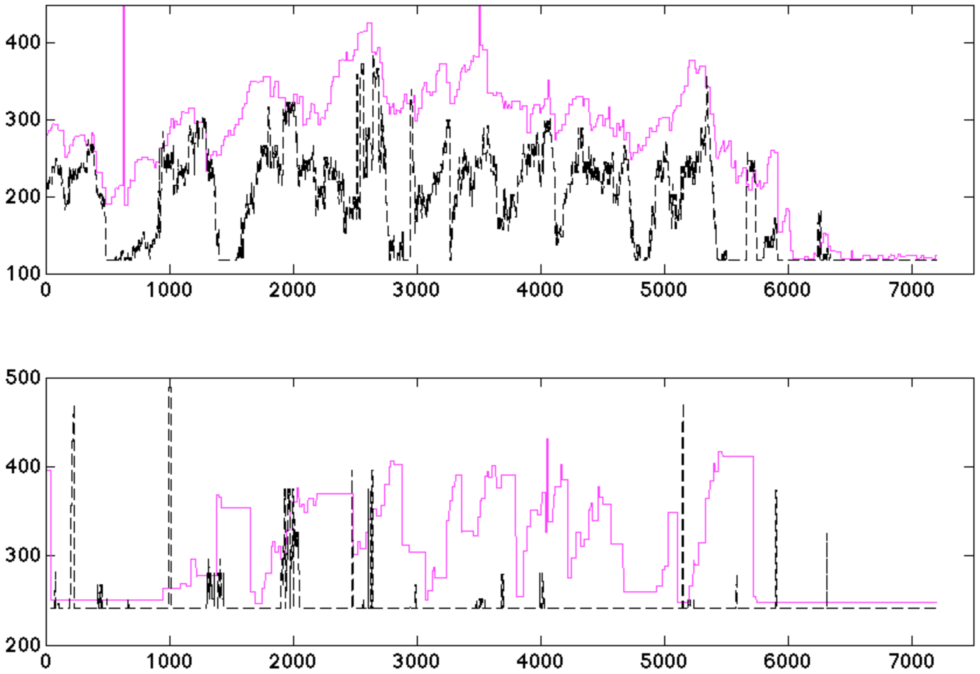

Figure 40. Multi Bottleneck Model. Main route (above) and the alternative route (below). Purple represents the real travel times; black represents predicted value applied in feedback.

Table 5A. Evaluation table for the Berlin network.
"Agents" are all agents that travel the whole main route and the alternative route, respectively.
"Travel time", "fit" and "Std. Nash deviation" is based on the group of agents defined above.
"Total Score" is based on all agents in the population. "Subpopulation Score" is based on agents that pass both the sign link and the destination link.

	Normal case	**No accident, Basic SB**	**Accident, No Control**	**Reactive (No model)**	**Basic SB**
Main route					
Travel time	368	292	1014	604	358
Agents	515	499	511	427	390
Fit	(94.33)	93.61	(61.12)	(no pred.)	77.85
Alternative route					
Travel time	311	292	317	482	320
Agents	7	133	7	101	123
Fit	(79.26)	80.70	(78.09)	(no pred.)	84.96
Total					
Travel Time	367	292	1004	580	349
Std. Nash deviation	116	160	764	256	127
Score					
Total Score	136,01820	135,89427	136,03201	135,90236	135,81067

Table 5B. Evaluation table for the Berlin network.

	Distribution	Disturbance	Distribution+ Disturbance	Basic SB Incident detection	SB all features	Multi Bottleneck
Main route						
Travel time	357	354	352	369	359	432
Agents	393	391	389	395	396	402
Fit	77.79	77.95	70.21	80.22	79.02	72.53
Alternative route						
Travel time	313	325	308	313	311	333
Agents	119	126	126	123	122	116
Fit	90.46	82.67	87.73	84.89	87.16	81.90
Total						
Travel time	347	347	341	356	348	410
Std. Nash deviation	110	135	118	121	105	78
Score						
Total Score	135,79666	135,80525	135,83151	135,87289	135,85715	135,85100

4.5 Economic evaluation

In contrast to the last paragraph we evaluate in this section the score introduced in chapter 4.2 when using different models. The subpopulation score reflects the economic benefit (or disbenefit) of the route guidance for the agents traveling either of the two routes. Table 6 displays the score analysis of the different guidance strategies for the Berlin Tegel location. The first row contains aggregated scores for the subpopulation of interest. The second row displays the difference in the aggregated score from the accident case when no control is applied. Average score values are found in the third row. The fourth row shows the difference in the average score from the accident case when no control is applied. For all models except the distribution model the guided agents benefit from the guidance. The disturbance model gives the highest gains; agents benefit 2.45 euros on average. The resulting aggregated benefits for a 100% population range from 22'813 euros with reactive control to 44'955 euros with the disturbance model. Note that these economic benefits are *per incident.* A successfully implemented system would be able to accumulate these benefits from incident to incident.

Table 6. Aggregated and average score values for the subpopulation.

	Accident, no control	Reactive control	Basic SB	Distri-bution	Distur-bance	Detection	SB with all features	MB

Score sum	2'420'115	2'442'928	2'447'561	2'418'502	2'465'069	2'456'258	2'456'119	2'455'236
Difference		22'813	27'446	-1'613	44'955	36'144	36'004	35'121
Score avg.	132.03	133.27	133.53	131.94	134.48	134.00	133.99	133.95
Difference		1.24	1.50	-0.09	2.45	1.97	1.96	1.92

One should also note that maximizing the economic benefit is easier than the control goal, which is getting the time difference between the alternatives to zero. This is because the economic benefit is maximized when both alternatives operate at capacity as much as possible, no matter how the travel times are distributed between the alternatives. If this appears counter-intuitive, one should consider that a notion of "fairness" is not included in standard economic appraisal (although it is usually ingrained into the behavioral models).

Drivers who receive the guidance will benefit. Unfortunately, the benefit to the traffic system as a whole is not that obvious. The average scores of the whole population are displayed in the last rows in table 5A and 5B. Applying guidance *de*creases the total score slightly. Hypothetically, this is due to the negative influence of the control on travelers using the alternative route already under normal traffic conditions, i.e. when there is no accident. Furthermore, the accident takes place in the evening peak which means that agents that receive the guidance are on their way home from work. In the scoring setup, there is no penalty for arriving late at home, and the benefit for longer home activities is low. It is likely that utility benefits would increase if the accident would occur during the morning rush hour, when delays would have a considerably larger negative impact on the score.

Another reason for the marginal influence of the system as a whole is that guidance focuses only on the travel times on the two routes. The reason for negative network effects could be that the guidance has a local control goal, i.e. it cares only about the subsystem that is subject to control. Therefore, it seems worth to understand and improve the system as whole taking not only travel time minimization into account.

4.6 Equipment ratio

The results presented so far are all based on the assumption that 80 percent of the agents are equipped with a COOPERS' device, and comply with the advice. It can be argued for that this is a relatively high equipment ratio, imposing the question of how many agents must be equipped with a COOPERS' device in order to get a significant benefit from the route guidance? Table 7 shows results of simulation runs using the Single Bottleneck model with all features and different compliance rates. It is clear that even with a equipment rate of only 20 percent, the subpopulation is better off than if no control were applied.

Table 7. Scores of agents travelling on the highway during the day for different equipment rates. Even when only 20 percent of the agents possess a device, guidance results in economic benefits.

	No control	**20 %**	**30 %**	**40 %**	**50 %**
Score sum	242011.49	242475.36	246567.64	244707.11	244662.06
Score avg.	132.03	132.28	134.52	133.50	133.48

5 Discussion

This report concentrates on two elements:

- Improvements of simple traffic controllers by using better, but still simplified, models for the prediction of traffic jams
- Evaluation of traffic control using multi-agent simulation

The main result concerning "model based control" is that good predictive models reduce the necessary sophistication of the controller itself. While our previous report indicated that simple controllers should have a proportional component, with the need to tune the proportionality constant, using a good predictive model means that a simple bang-bang controller brings a similar performance without the need to tune the controller. The only necessary sensors are car-counting devices.

The approach can be used everywhere where the following conditions are fulfilled: (a) there is a clear alternative to a main route; (b) both the main route and the alternative route have relatively few additional entries and exits, and all entries and exits are covered by car counting devices. Once these pre-requisites are fulfilled, it should be relatively straightforward to implement the method.

The other result is that agent-based simulation can be successfully used to design and evaluate telematics measures. Advantages of agent-based simulation include:

- It is possible to include how traffic management could be improved if the COOPERS device knew the final destination, e.g. making detours dependent on the final destination of the trip. This was tested in our investigations by giving recommendations only to a subset of people that had a certain intermediate destination.
- It would be possible to simulate the effect of individual guidance, e.g. guiding different vehicles along different detours even if they have the same destination.
- It would be possible to simulate the effects of a COOPERS back channel.
- An economic evaluation is automatically included in the approach. This is because the behaviour of the individual agents is based on economic theory. This includes the effects of schedule delay and risk, aspects that are difficult to capture with more traditional methods.
- The simulations run fast enough to approach problems of wide area control (regions with several millions of inhabitants).

This report describes the first steps in this direction.

The multi-agent simulation can, in principle, be transferred from one location to another. A study of, say, the Austrian freeway network, would be possible. The challenge is that setting up a valid base case is a rather time-consuming exercise.

6 Conclusions

There is no doubt that the prediction of travel times improves the route guidance, measured in terms of lower travel times. The simulations in the test network very clearly confirm that the Single Bottleneck approach for calculating travel times is well suitable in MATSim. Compared to reactive control, the Nash times are reduced by 95 percent in the test network and 57 percent in the Berlin simulation. The travel times are reduced by 26 and 41 percent, respectively.

According to the test network, all of the investigated models have positive effect on the situation that they were designed to handle. The Distribution model, the Disturbance model and the Multi Bottleneck model all increase the model fit significantly in the test network. For the real world test case (the “Berlin scenario”), the Single Bottleneck model clearly outperforms purely reactive control. The other models did not deliver any further improvement beyond that, but that may have been a property of the selected test situation rather than a property of the models.

One of our aims that were achieved was to eliminate the need of parameter tuning. With good predictions and a minimal message hold-time, there is no need to apply more advanced controllers than the bang-bang controller, which has no control parameters. Neither do the prediction models need to be estimated – all the necessary information is in the network characteristics. Once the routes are specified, neither the prediction model nor the feedback controller needs any more estimation. In contrast to any controller that produces a continuous control signal, the bang-bang control signal is already compatible with the COOPERS device, that is, for every specific vehicle there is a specific guidance.

Finally, the report demonstrates that it is in principle possible to attach an economic benefit to the telematics device. That economic benefit is calculated from individual agent scores, which in turn reflect effects such as value of (travel) time, opportunity cost, and schedule delay cost. An advantage of a behaviourally oriented traffic microsimulation (such as MATSim) is that such measures can be extracted in a conceptually forward way. It would also be possible to include aspects such as environmental cost into the benefit calculation.

7 References

Cascetta, E., Cantarella, C. (1991), A day-to-day and within-day dynamic stochastic assignment model, Transportation Research A, 25A(5), 277–291.

Cetin, N. (2005), Large scale parallel graph-based simulations, Ph.D. thesis, Swiss Federal Institute of Technology ETH.

Charypar, D., Nagel, K. (2005), Generating complete all-day activity plans with genetic algorithms, Transportation, 32(4), 369–397.

Diakaki, C., Papageorgiou, M., McLean, T. (1997), Simulation studies of integrated corridor control in Glasgow, Transportation Research Part C, 5, 211–224.

Gawron, C. (1998), An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model, International Journal of Modern Physics C, 9(3), 393–407.

Illenberger, J., Flötteröd, G., Nagel, K. (2007), Enhancing MATSim with capabilities of within-day re-planning, in: IEEE Intelligent Transportation Systems Conference, Seattle, WA.

Messmer, A., Papageorgiou, M., Mackenzie, N. (1998), Automatic control of variable message signs in the interurban Scottish highway network, Transportation Research 6C, pp. 173–187.

Nagel, K., Grether, D., Beuck, U., Rieser, M., Chen, Y., Axhausen, K. (in press), Multi-agent transport simulations and economic evaluation, in: LeBaron, B., Winker, P. (Editors) Agent based models for economic policy advice, Journal of Economics and Statistics, also VSP working paper number 07-14, see www.vsp.tu-berlin.de/publications.

Raney, B., Nagel, K. (2006), An improved framework for large-scale multi-agent simulations of travel behavior, in: Rietveld, P., Jourquin, B., Westin, K. (Editors) Towards better performing European Transportation Systems, London: Routledge, similar version TRB preprint number 05-1846.

Rommel, C. (2007), Automatic feedback control applied to microscopically simulated traffic – the potential of route guidance in the Berlin traffic network, VSP Working Paper 07-16, TU Berlin, Transport Systems Planning and Transport Telematics.

Rossetti, R., Liu, R. (2005), A Dynamic Network Simulation Model Based on Multi-Agent Systems, in: Klügl, F., Bazzan, A., Ossowski, S. (Editors) Applications of Agent Technology in Traffic and Transportation, Whitestein Series in Software Agent Technologies and Autonomic Computing, pp. 88–93, Birkhäuser.

Wang, Y., Papageorgiou, M. (2005), Real-time traffic state estimation based on extended Kalman filter: a general approach, Transportation Research Part B, 39, 141–167.