Driver Behavior in Traffic

December 2012

NTIS Accession No. PB2012-105330

Publication No. FHWA-HRT-12-036

U.S. Department of Transportation
Federal Highway Administration

Notice

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the use of the information contained in this document. This report does not constitute a standard, specification, or regulation.

The U.S. Government does not endorse products or manufacturers. Trademarks or manufacturers' names appear in this report only because they are considered essential to the objective of the document.

Quality Assurance Statement

The Federal Highway Administration (FHWA) provides high-quality information to serve Government, industry, and the public in a manner that promotes public understanding. Standards and policies are used to ensure and maximize the quality, objectivity, utility, and integrity of its information. FHWA periodically reviews quality issues and adjusts its programs and processes to ensure continuous quality improvement.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-HRT-12-036	PB2012-105330	

4. Title and Subtitle	5. Report Date
Driver Bahavior in Traffic	December 2012
	6. Performing Organization Code:

7. Author(s)	8. Performing Organization Report No.
Montasir Abbas, Alejandra Medina, Linsen Chong, Bryan Higgs, Catherine McGhee, Mike Fontain, Lukas Kautzsch, Axel Leonhardt, and Kiel Ova.	

9. Performing Organization Name and Address	10. Work Unit No.
Virginia Polytechnic Institute and State University Office of Sponsored Program 1880 Pratt Drive, Suite #2006 Blacksburg, VA 24060	11. Contract or Grant No. DTFH61-09-H-00007

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Office of Safety R&D Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296	Final Report September 2009 – February 2012
	14. Sponsoring Agency Code

15. Supplementary Notes

Contracting Officer's Technical Representative: David Yang, HRDS-30

16. Abstract

Existing traffic analysis and management tools do not model the ability of drivers to recognize their environment and respond to it with behaviors that vary according to the encountered driving situation. The small body of literature on characterizing drivers' behavior is typically limited to specific locations (i.e., by collecting data on specific intersections or freeway sections) and is very narrow in scope. This report documented the research performed to model driver behavior in traffic under naturalistic driving data. The research resulted in the development of hybrid car-following model. In addition, a neuro-fuzzy reinforcement learning, an agent-based artificial intelligence machine-learning technique, was used to model driving behavior. The naturalistic driving database was used to train and validate driver agents. The proposed methodology simulated events from different drivers and proved behavior heterogeneities. Robust agent activation techniques were also developed using discriminant analysis. The developed agents were implemented in VISSIM simulation platform and were evaluated by comparing the behavior of vehicles with and without agent activation. The results showed very close resemblance of the behavior of agents and driver data. Prototype agents prototype (spreadsheets and codes) were developed. Future research recommendations include training agents using more data to cover a wider region in the Wiedemann regime space, and sensitivity analysis of agent training parameters.

17. Key Words	18. Distribution Statement
Naturalistic Driving, Agent-Based Modeling and Simulation, Reinforcement Learning, Artificial Intelligence, Neuro-Fuzzy Actor-Critic	No restrictions. This document is available through the National Technical Information Service; Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	227	

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
^oC	Celsius	1.8C+32	Fahrenheit	^oF
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

$$SDV2 = \left(\frac{\Delta x - L_{n-1} - AX}{CX2}\right)^2$$

$$b_{max} = BMAXmult * (v_{max} - v * FaktorV)$$

$$FaktorV = \frac{v_{max}}{v_{des} + FAKTORVmult * (v_{max} - v_{des})}$$

$$b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1}$$

$$b_{null} = BNULLmult * (RND4_n + NRND)$$

$$b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1} + b_{min} * \frac{ABX - (\Delta x - L_{n-1})}{BX}$$

$$b_{min} = -BMINadd - BMINmult * RND3_n + BMINmult * v_n$$

LIST OF TABLES

Chapter 1. INTRODUCTION

BACKGROUND

Existing traffic analysis and management tools do not model the ability of drivers to recognize their environment and respond to it with behaviors that vary according to the encountered driving situation. The small body of literature on characterizing drivers' behavior is typically limited to specific locations (i.e., by collecting data on specific intersections or freeway sections) and is very narrow in scope. The majority of traffic modeling and parameter calibration research has assumed somewhat similar driving conditions and behavioral sets for the entire population of drivers. Differences in drivers' actions are represented merely by drawing samples from statistical distributions assigned to each driver type. The number and properties of driver types used in this approach becomes a calibration issue, which might or might not improve the modeling effort. This approach does not capture or predict the actions of individual drivers responding differentially to situational factors.

A comprehensive representation of driver behavior varying in response to encountered driving situations is not available in the literature and, therefore, reproduction of such behaviors using statistical parameters is not adequate. At any instant, a driver's action will be influenced by the driver's perception of the surrounding environment or changes in the driving situation, the vehicle's position along its driving path, and the vehicle's position relative to other vehicles.

This report documents the research findings of the Exploratory and Advanced Research "Driver Behavior in Traffic." The objective of this research was to characterize driver behavior under naturalistic driving conditions. More specifically, the research aims to determine critical parameters related to freeway driving, including target speed selection; speed change accelerations; lane changing behavior; car-following distance; response times; and emergency stopping behavior for representative freeway state conditions. Of special interest to this research was the modeling of both normal and safety-critical driving. The techniques used in this research, namely agent-based simulation techniques, allow modeling of individual driver behavior beyond existing models' capabilities.

OVERALL SCOPE OF THE EFFORT

The scope of the work is to develop and implement intelligent agents that can encapsulate individual drivers' decisions in response to varying traffic situations. The developed agents were designed to learn individual drivers' actions for any given traffic state retrieved from the Virginia Tech Transportation Institute's (VTTI) database of naturalistic driving data. These characteristic driving rules of the agents were coded in a VISSIM simulation environment to test and study their effects on agents' behavior in different conditions and scenarios (as illustrated in Figure 1). The goal of this effort is to provide the industry with methods for developing more accurate and more sensitive traffic simulation models. This could also enable future research to develop new generations of traffic simulation models that accurately model driver behavior during incidents and other complex traffic situations.

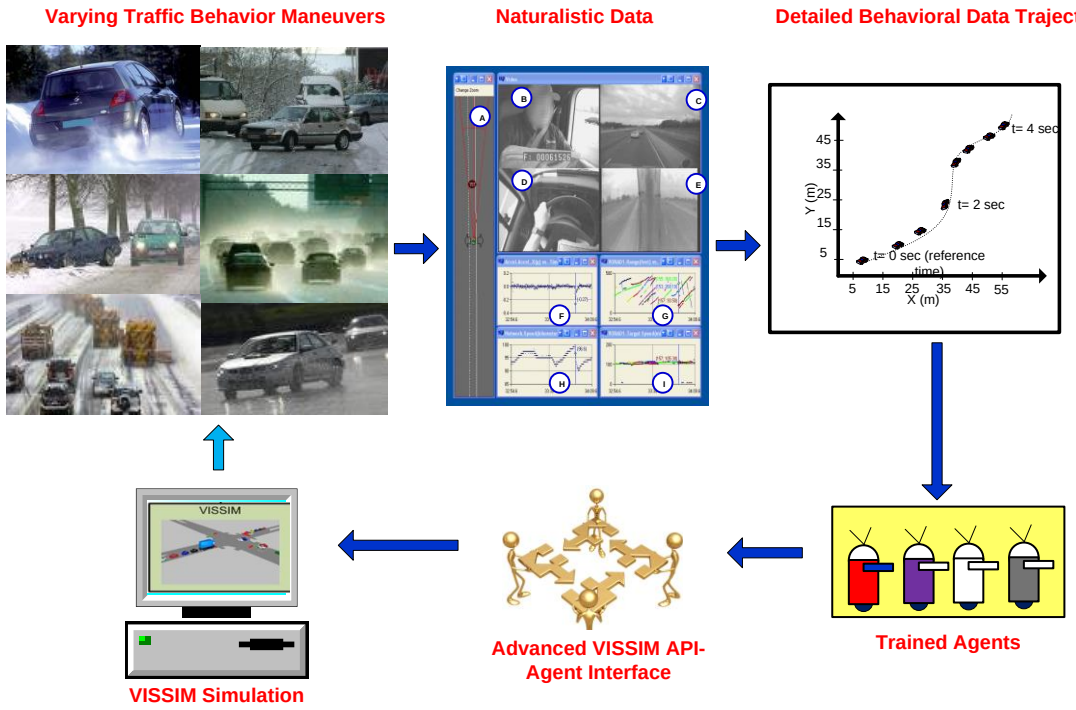

Figure 1. Illustration. System representation of driver's response to encountered traffic situation.

REPORT ORGANIZATION

This report is organized as follows: Chapter 2 provides a synthesis of the literature on key domestic and international past and continuing studies on car-following and lane-changing microscopic models, naturalistic data collection efforts, driver behavior and characteristics, artificial intelligence learning techniques, linear and non-linear function approximation techniques, and agent-based simulation. Chapter 3 describes the VTTI naturalistic driving data processing and provides a summary of the car and truck drivers that were selected for further analysis. Chapter 4 provides a detailed description of the Wiedemann car-following model and underlines the procedure used to calibrate the model. The chapter also describes how an improved Wiedmann model was developed based on the information found in the naturalistic driving data. Chapter 5 describes the development of intelligent agents that emulate drivers' actions for any given traffic state. Artificial intelligent techniques, such as reinforcement learning (RL), Artificial neural networks, and fuzzy logic implementations are also described. Chapter 6 discusses robust statistical techniques to activate the developed agents for simulating safety-critical events. The chapter describes the use of discriminant analysis to find the difference between normal driving behavior and safety critical event behavior. Chapter 7 describes the agent implementation in the VISSIM platform to validate the research methodology and identify potential implementation issues for future research. Chapter 8 describes the evaluation of the developed prototypes and compares the results of agent-based simulation with Wiedemann (normal) simulation. Finally, chapter 9 provides a summary of the findings of the research and recommendation for future work.

Chapter 2. LITERATURE REVIEW

INTRODUCTION

Existing traffic analysis and management tools do not model drivers' ability to recognize their environment and respond to it differently depending on the situation. There is very limited literature on characterizing driver behavior. What research exists is typically limited to specific locations (i.e., by collecting data on specific intersections or freeway sections) and is very limited in scope. The majority of traffic modeling and parameter calibration research has assumed somewhat similar driving conditions and behavioral sets for the entire driver population. Differences in drivers' actions are represented merely by drawing samples from statistical distributions assigned to each driver type. The number and properties of driver types used influences calibration, and this influence might or might not improve the modeling effort. This approach does not capture or predict the effects of situational factors on individual drivers' actions. A comprehensive theoretical representation of drivers' varying behavior in response to encountered driving situations is very limited and, therefore, reproduction of such behaviors (or actions) using statistical parameters is not adequate.

This chapter synthesizes a literature review conducted by the research team on key domestic and international studies on car-following and lane-changing microscopic models, naturalistic data collection efforts, driver behavior and characteristics, and artificial intelligence learning techniques especially reinforcement learning.

CAR-FOLLOWING MODELS

In the last 50 years, a considerable amount of research has focused on modeling longitudinal driver behavior, producing a large number of car-following models (Treiber et al. 2006[1]), including Gazis-Herman-Rothery (GHR) models, safety distance models, linear models, and psychophysical or action point models. Most microscopic models assume that human drivers react to the stimuli from neighboring vehicles with the dominant influence originating from the directly leading vehicle. This is known as the "follow-the-leader" or "car-following" approximation. Other models introduced "safe time headway" and a desired velocity. Calibrating these car-following models required different levels of effort, and the results depend on data availability, calibration method, and model structure. The GHR model, for instance, requires the calibration of two parameters, limiting the application of the model. The Gipps model (Gipps 1981[2]) (a safe-distance model) uses two different transfer functions for reproducing free-flow and car-following conditions. Psychophysical models, on the other hand, assume that the driver will perform an action when a threshold (a function of speed difference and distance) is reached. Estimating distances and speed differences among drivers makes it difficult to calibrate the individual threshold associated with this model. Linear regression and heuristic algorithms (e.g., genetic algorithms) are two widely used methods for model calibration. Despite different mechanisms and software interfaces, when multiple simulation software applications were compared, the resulting behaviors of the models showed similarities (Olstam et al. 2004[3]). In addition, it seems that error cannot be eliminated even if more parameters are introduced.

The importance of not only developing calibration procedures but also interpreting obtained calibration results is discussed in Ossen (2008[4]). The three main findings were "(1) measurement errors can yield a considerable bias in the estimation results; (2) parameters

minimizing the objective function do not necessarily capture following dynamics best; (3) measurement errors substantially reduce the sensitivity of the objective function and consequently reduce the reliability of the results."

A significant part of the deviations between measured and simulated trajectories can be attributed to the interdriver variability and the intradriver variability (human drivers do not drive constantly over time, and their behavioral driving parameters change). The latter accounts for a large part of the deviations between simulations and empirical observations (Kesting and Treiber 2008[5]).

Development of Car-Following Models

The first car following model describing car following in terms of vehicle trajectory rather than just velocities and spacing was proposed by Herrey and Herrey in 1945 (Herrey and Herrey 1945; Newell 2002). Without extensive experimental basis, they proposed that a driver must maintain a "safe driving distance" (including the stopping distance), which results in a quadratic relationship between spacing and speed. The safety distance models or collision avoidance model represents model in which the driver of the following vehicle always keep a safe distance to the vehicle in front.

Pipes Model

In 1953, Pipes(Pipes 1953) developed one of the first car-following model using the California Motor Vehicle Code rules: "allow at least the length of a vehicle between the subject and leading vehicle for every ten miles per hour speed at which the subject vehicle is traveling" and assuming a 20 feet for the length of the vehicle. The model is divided into two equations: The first equation applies when the lead vehicle has a constant acceleration while trying to reach a cruising speed. The second equation applies after the lead vehicle reaches and maintains the cruising speed. Both equations estimate the acceleration of the following vehicle at time *t* based on the cruising speed and acceleration time of the lead vehicle. This model is presented in a fashion that is applicable to a line of traffic with each vehicle accelerating according to the action of the vehicle in front of it.

$$a_{k+1}(t) = \frac{v_m}{T_0} G_k\left(\frac{t}{T}\right) if\ 0 < t < T_0$$

$$a_{k+1}(t) = \frac{v_m}{T_0}\left[G_k\left(\frac{t}{T}\right) - G_k\left(\frac{t - t_0}{T}\right)\right] if\ t > T_0$$

Where:

t = time,

T= a time constant ≈ 1,

a_{k+1}= acceleration of the kth+1 (following) vehicle,

v_m= constant velocity of lead vehicle (cruising speed),

T_0= time for lead vehicle to reach cruising speed at a constant acceleration from rest at t=0, and

$G_k(t) = 1 - e^{-t}\left[1 + \frac{t}{1!} + \frac{t^2}{2!} + \cdots + \frac{t^{(k-1)}}{(k-1)!}\right]$.

$G_k(t)$ is the ratio of an incomplete to a complete gamma transform function.

Most of the early work in car following assumed that the driver is able to consider the space headway and the relative speed between his car and the leader vehicle. To model this stimulus response model, several approaches have been developed, including linear, proportional approaches and rule base approaches based on Fuzzy logic. In general these car-following models compute the reaction of a following vehicle to the actions or stimulus of the leading vehicle as Response (t) = Sensitivity X Stimulus (t-T). The response of the following vehicle is translated into an acceleration or deceleration in proportion to the stimulus after a time lag T (reaction time of the driver of the following vehicle).

The GM models

In 1958, a series of models were developed at the GM Research laboratory by Chandler et al (Chandler et al. 1958), Herman and Potts (Herman and Potts 1959) and Gazis et al (Gazis et al. 1959; Gazis et al. 1961). The most important of these one is the Generalized linear model, usually know as the GM model. In fact most of the other models, for example Pipes (Pipes 1953)) and Forbes(Forbes 1963) are also special cases of the GM Model.

Chandler, Herman and Montroll proposed a linear model where it is assumed that the acceleration of the following car depends exclusively on the speed difference between the cars

$a_n(t) = \alpha \Delta v_n^{front}(t - \tau_n)$

Where:

$a_n(t)$=acceleration applied by the driver n at time t,

α=Constant,

$\Delta v_n^{front}(t - \tau_n) = v_n^{front}(t - \tau_n) - v_n(t - \tau_n)$,

$v_n^{front}(t - \tau_n)$=subject speed at time $(t - \tau_n)$,

t=time of observation, and

τ_n=reaction time for driver n.

Gazis Herman and Potts developed the GM 2 parameter model (GM2P) where the linear model constant sensitivity term was further developed to consider the macroscopic speed density relationship and the space headway. The incorporation of the space headway implies that the driver is more sensitive to the action of the leading vehicle in the case of smaller headways .

$a_n(t) = \frac{\alpha}{\Delta x_n(t-\tau_n)} \Delta v_n^{front}(t - \tau_n)$

Where:

$\Delta x_n(t - \tau_n)$ is the space headway at time $(t - \tau_n)$,

α=Constant, estimated for each data set using correlation analysis.

The model of Edie or GM 3 parameter model (GM3P) addressed the shortcoming of the linear model of free-flow speed at zero density and assumes that the reaction of the following car depends also on the speed of the following car and not only on the speed difference and the relative distance between the two cars. The sensitivity is proportional to the speed and inversely proportional to the square of the relative distance

$a_n(t) = \alpha \frac{v_n(t)}{\Delta x_n(t-\tau_n)} 2\Delta v_n^{front}(t - \tau_n)$

The Gazis,Herman and Rothery model usually refer as the GHR model or as the *general car following mode* was introduced in 1961 and is represented by the following equation

$a_f(t+T_r) = cv_f^m(t+T_r)\frac{\Delta v(t)}{\Delta x^l(t)}$

Where:

T_r =time between the observation of a certain stimulus and the reaction on that stimulus,

$a_f(t+T_r)$ =acceleration of the following vehicle at time $t+T_r$,

$v_f(t+T_r)$ = speed of following vehicle at time $t+T_r$,

$\Delta v(t)$= relative speed between the following car and the car immediately in front ($v_{leader} - v_{follower}$),

$\Delta x(t)$= relative distance between following car and car immediately in front ($x_{leader} - x_{follower}$),

m, l, c= parameters describing the car following behavior.

The GHR model is therefore a general form of the early models. When l=m=0 , the GHR model represents that of Chandler, Herman and Montroll; when m =0 and l=1, the model represents Gazis, German and Potts' model; and when m=1 and l=1 the model represent's Edie's model.

The Wiedemann Model

In 1974, Wiedemann (Wiedemann 1974) introduced a car-following model that is based on the psycho-physical behavior. Wiedemann model addresses two unrealistic assumptions of the GM models: First, driver follows the leading vehicle whatever the spacing is. Second, driver has perfect reaction and response time. The model has been improved since then by Leutzbach (Leutzbach 1988) by introducing perceptual thresholds as minimum values of the stimulus. The concept of thresholds in Wiedemann model captures the driver's alertness in small space headway and the lack of explicit car-following behavior with large headways. In addition, it allows the model to explain the oscillation phenomena observed in car-following behaviors. The following diagram shows Wiedemann car-following thresholds with respect to vehicle speed difference and space headway.

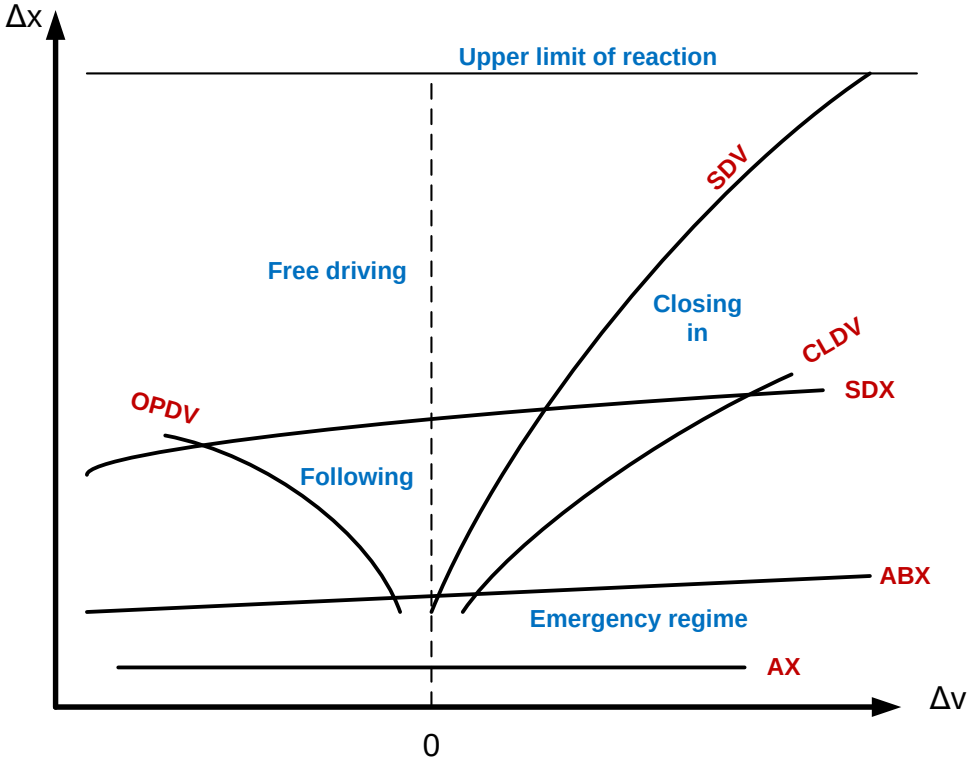

Figure 2 Wiedemann car-following model's thresholds (AX, ABX, SDX, CLDV, SDV, OPDV) (Wiedemann and Reiter 1992) (Olstam and Tapani 2004)

The following is a definition of Wiedemann car-following model's thresholds:

1- The threshold AX represents the desired distance between stationery vehicles. This distance consists of the leading vehicle length and the desired front-to-rear distance.

$AX = L_{n-1} + AXadd + RND1_n \times AXmult$

Where:

L_{n-1} : The leading vehicle length,

AXadd and *AXmult* are calibration parameters,

$RND1_n$ is a normal distributed random number for vehicle (n).

2- The threshold ABX represents the desired minimum following distance at low speed differences.

$ABX = AX + BX$

$BX = (BXadd + BXmult \times RND1_n) \times \sqrt{v}$

$v = \begin{cases} v_{n-1} \, for \;\; v_n > v_{n-1} \\ v_n \quad for \;\; v_n \leq v_{n-1} \end{cases}$

Where:

BXadd and *BXmult* are calibration parameters,

v: Vehicle speed,

3- The threshold for maximum following distance, SDX, fall between 1.5 and 2.5 times ABX

$SDX = AX + EX \times BX$

$EX = (EXadd + EXmult \times (NRND - RND2_n))$

Where:

EXadd and *EXmult* are calibration parameters,
NRND is a normal distributed random number, and
$RND2_n$ is normal distributed driver dependent parameter.

4- The threshold approaching point, SDV, represents the point where the driver notices that he/she is approaching a slower vehicle.

$$SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX} \right)^2$$

$$CX = CXconst \times (CXadd + CXmult \times (NRND1_n + RND2_n))$$

Where:
CXconst, *CXadd* and *CXmult*i are calibration parameters.

5- Decreasing speed differences, CLDV, threshold provides a mechanism for a different behavior when the driver approaches the leading vehicle more closely than the approaching point. In VISSIM, CLDV is ignored and simply assumed to be equal to SDV.

6- Increasing speed difference, OPDV, threshold describes the point where the driver observes that he/she is traveling at a lower speed than the leader.

$$OPDV = CLDV \times (-OPDVadd - OPDVmult \times NRND)$$

Where:
OPDVadd and *OPDVmult* are calibration parameters,
NRDV is a normally distributed random number.

The Fritzsche Model

Fritzsche (Fritzsche 1994) developed a similar psycho-physical model that has been used in Paramics. The model has multiple regimes that apply to different traffic situation (i.e., normal car-following, free flow, closing in, etc.). Fritzsche's model accounts for human perception in the definitions of the model regimes as driver perceived only speed differences with certain magnitude. Fritzsche defined the thresholds for perception as: perception threshold of positive speed difference (PTP) and perception threshold of negative speed difference (PTN). The model also incorporates four thresholds for the follower's space headway.

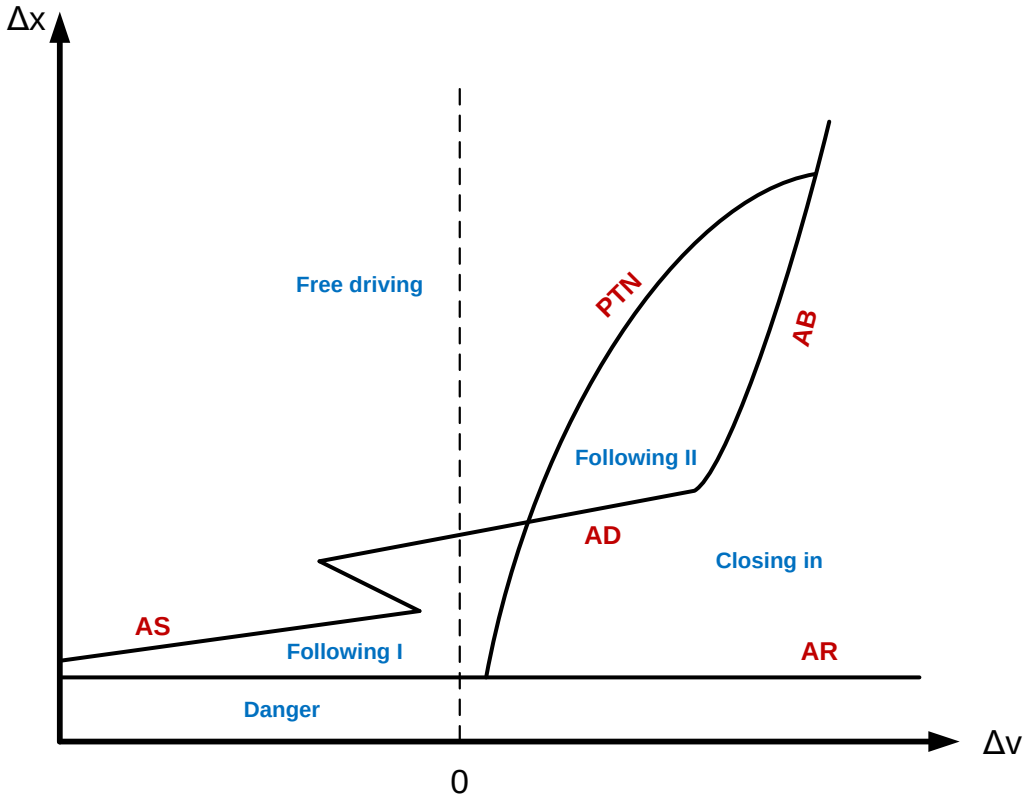

Figure 3. Different threshold and regimes in the Fritzsche car-following model(Olstam and Tapani 2004)

1- Desired distance threshold, AD, represents the gap which following driver wants to maintain
$AD = S_{n-1} + T_D \times v_n$
Where:
T_D : Parameter represents the desire time gap,
S_{n-1}: Effective length of the leading vehicle, and
v_n : following vehicle speed.

2- The risk distance threshold, AR, represents the distant where driver apply heavy deceleration to avoid collisions.
$AR = S_{n-1} + T_r \times v_{n-1}$
Where:
T_r : Parameter represents the risky time gap.

3- The safe distance threshold, AS, represents the smallest headway where positive acceleration is accepted if the distance between the vehicles is increasing.
$AS = S_{n-1} + T_s \times v_n$
Where:
T_s : Parameter represents the safe distance gap.

4- The risk braking distance threshold, AB, represent the distant where maximum deceleration is limited. Therefore collision is possible to occur if the initial speed difference is large.

$$AB = AR + \frac{\Delta v^2}{\Delta b_m}$$

Where

$\Delta b_m = |b_{min}| + a_{n-1}$

b_{min} , a_{n-1} : Parameter controlling maximum deceleration.

The CA Model

The stochastic automaton model , usually know as CA Model (Nagel and Schreckenberg 1992), is a model to simulate freeway traffic. Monte-Carlo simulations of this model show a transition from laminar traffic flow to start-stop-waves with increasing vehicle density.

The CA model can be described as following seven steps.(Nagel and Schreckenberg 1992; Lan et al. 2009)

Step 1- Determination of the randomization probability

$$p\big(v_n(t), t_h, t_s, S_{n+1}(t)\big) = \begin{cases} p_b: if\ S_{n+1} = 1\ and\ t_h < t_s \\ p_0: if\ v_n = 0\ and\ t_{st} \geq t_{k,c} \\ p_d: in\ all\ other\ cases \end{cases}$$

Where

$t_h = \frac{d_n}{v_n(t)}, t_s = \min(v_n(t), h_k),$

t_{st} denotes time vehicle stops.

Step 2: Acceleration. Determine the speed of vehicles in the next time step. $S_n(t)$ is also taken into consideration. $S_n(t)$ is determined in Step 5

If $(S_{n+1}(t) = 0)$ or $(t_h \geq t_s)$ then $v_n(t+1) = \min\big(v_n(t) + a_k, v_{k,max}\big)$

Else $v_n(t+1) = v_n(t)$

Where $v_{k,max}$ and a_k are the maximum speed and acceleration capacity of vehicles of type k

Step 3 Deceleration. Set speed restriction when the vehicle in front is too close, thus locates within the effective distance $\left(d_n^{eff}\right)$

$$v_n(t+1) = \min\left(d_n^{eff}, v_n(t+1)\right)$$

Step 4. Randomization

If $(rand() < p)$ then $v_n(t+1) = \max\ (v_n(t+1) - 1{,}0)$

Step 5. Determination of vehicle status identifier $S_n(t)$ in next time-step

$$S_n(t+1) = \begin{cases} 0 & if\ v_n(t+1) > v_n(t) \\ S_n(t) & if\ v_n(t+1) = v_n(t) \\ 1 & if\ v_n(t+1) < v_n(t) \end{cases}$$

Step 6. Determination of time t_{sk} stuck inside the jam

$$t_{st} = \begin{cases} t_{st} = t_{st} + 1\ if\ v_n(t+1) = 0 \\ t_{st} = 0 \quad if\ v_n(t+1) > 0 \end{cases}$$

Step 7 Update position

$$x_n(t+1) = x_n(t) + v_n(t+1)$$

Where:

n: nth vehicle,

$n+1$: Vehicle in front,

k: Type of vehicle,

P_b: Accounting for impact of decelerating vehicle in near front,

P_0: Reflect the delay-to-start behaviors of vehicles stuck in traffic jam,

P_d: Other situations,

P_c: Probability of lane change,

$t_h = d_n/v_n(t)$: Time headway of nth vehicle to front,

h_k: Preset time threshold of vehicle of type k in reflecting the effect of synchronized distance,

$t_s = \min(v_n(t), h_k)$: Final time threshold for initiating the consideration of the front brake light effect, taking the vehicular speed into consideration,

t_{st}: Accumulated time of vehicle stuck in traffic jam,

$t_{k,c}$: Time threshold of vehicle of type k for initiating the stop-to-start behavior,

$\tau^{(safe)}$: Safe time gap for collision prevention,

Δt: Duration of an individual time step,

d: Space headway,

d_n^{eff}: Effective distance of nth vehicle,

g: Distance gap,

x: Position of nth vehicle,

$\dot{x}$: Speed (1st derivative of vehicular position),

$\ddot{x}$: Acceleration /deceleration (2nd derivative of vehicular position), and

Δx: Basic unit of roadway length

The SK Model

The safety condition is imposed by the term $s_{gap}(t)$.(Hamdar and Mahmassani 2008b). Setting the maximum velocity equal to the space gap between two successive vehicles irrespectively of the required deceleration needs to be changed.

To address the fact that the CA model fails to reproduce metastable states of very high throughput and hysteresis effects existing in traffic flow, the model was generalize to provide a one-parametric family of models with a part of reproduces metastable states and hysteresis usually know as SK Model (Krauss et al. 1997), and represented by

$$v \le v_{safe} = b\left(\alpha_{safe} + \beta_{safe}\right)$$

$$\alpha_{safe} = \left[\sqrt{2\frac{d_p+g}{b} + \frac{1}{4}} - \frac{1}{2}\right]$$

$$d_p = b(\alpha_p\beta_p + \frac{\alpha_p(\alpha_p-1)}{2})$$

b =maximum deceleration

d_p=braking distance of leader car

α_p, β_p integer and fractional part of v_p/b

$\alpha_{safe}, \beta_{safe}$ integer and fractional part of $\left[\sqrt{2\frac{d_p+g}{b}+\frac{1}{4}}-\frac{1}{2}\right]$

The SK Model is a development of CA model(Krauss et al. 1997). Instead of step 7, SK uses the updating rules below:

$v_1 = min[v(t)+b, v_{max}, v_{safe}]$

$v_0 = v_1 - \epsilon\{v_1 - [v(t)-b]\}$

$v(t+1) = v_{ran,v_0,v_1}$

$x(t+1) = x(t) + v(t+1)$

Where:

ϵ: parameter,

v_1: optimal velocity for next update,

v_{max}: maximum velocity of cars,

v_{safe}: maximum safe velocity ,and

v_{ran,v_0,v_1}: a random number between v_0 and

v_1 =maximum decal

The IDM Model

The Intelligent Driver Model (IDM) (Treiber et al. 2000) is defined by the acceleration function:

$$a_n(t+\tau) = a\left[1-\left(\frac{v_n(t)}{v^*}\right)^{\delta}-\left(\frac{\Delta x^*_{min}(v_n(t),\Delta v_{n,n-1(t)})}{\Delta x_{n-1,n}(t)}\right)^2\right]$$

$$\Delta x^*_{min}\left(v_n(t),\Delta v_{n,n-1(t)}\right) = d + T_{safe} * v_n(t) + \frac{v_n(t)\Delta v_{n,n-1(t)}}{2\sqrt{a^{max}{}_* b^{\max abs}}}$$

Where:

$\Delta v_{n,n-1}(t)$=approaching rate of the following vehicle,

v^*=desired speed,

T_{safe}=safe time headway,

a^{max}=maximum desired acceleration of following vehicle,

$b^{\max abs}$=absolute maximum desired deceleration of following vehicle,

δ acceleration component,

$\Delta x_{n-1,n}(t)$=distance headway, and

d=vehicle length.

This expression combines the acceleration strategy $\dot{a}_{free} = a\left[1-\left(\frac{v_n(t)}{v^*}\right)^{\delta}\right]$ towards a desired speed v^* on a free road with the parameter a for the maximum acceleration with a braking strategy $\dot{a}_{break} = \left(\frac{\Delta x^*_{min}(v_n(t),\Delta v_{n,n-1(t)})}{\Delta x_{n-1,n}(t)}\right)^2$ which is dominant if the current gap $\Delta x_{n-1,n}(t)$ to the preceding vehicle becomes smaller than the desired minimum gap.

The Gipps Model

Gipps (Gipps 1981) developed the first general acceleration model that applies to both car-following and free-flow conditions. The model determines the maximum acceleration based

on: driver's desired speed and safe minimum safe headway. Gipps car-following model is implemented in AIMSUN. In Gipps car-following model, the acceleration is calculated for five different situations:

- Vehicle moves in its desired speed
- Vehicle in free move with speed less than the desired speed
- Vehicle is accelerating from stopping position
- Vehicle in a car-following state with constraint space headway
- Vehicle in a car-following state with active-non collision constraint

The following threshold constrained vehicle acceleration in Gipps' model:

$$v_n^a(t+T) = v_c(t) + 2.5\, a_n^{max}\, T\, \left(1 - \frac{v_n(t)}{v_n^{desired}}\right) . \sqrt{0.025 + \frac{v_n(t)}{v_n^{desired}}}$$

$$v_n^b(t+T) = d_n^{max}.T\, . \sqrt{(d_n^{max}.T)^2 - d_n^{max}\, . \left[2\{x_{n-1}(t) - s_{n-1}(t)\} - v_n(t).T - \frac{v_n(t)^2}{\hat{d}_{n-1}}\right]}$$

Where:

a_n^{max} : Maximum desired acceleration, vehicle n (m/s^2)

d_n^{max} : Maximum desired deceleration, vehicle n (m/s^2)

$\hat{d}_{n-1}$: Estimation of maximum deceleration desired by vehicle n-1, (m/s^2)

s_{n-1} : Effective length of vehicle

T: Reaction time

$\hat{d}_{n-1}$: Leader desired deceleration

x_{n-1} : vehicles spacing

The VDIFF Model

In the Velocity difference model (VDIFF) (Jiang, Wu et al. 2001(Jiang et al. 2001)) the acceleration function consists of a term proportional to a gap-dependent "optimal velocity" $v_{opt}(s)$ and a term that takes velocity differences Δv as a linear stimulus into account. The parameter τ is the relaxation time which describes the adaptation to a new velocity due to changes in s and v. The sensitivity parameter λ considers the crucial influence of Δv. The properties of the VDIFF are defined by the function for the optimal velocity $v_{opt}(s)$.

The parameter v_0 defines the desired velocity under free traffic conditions. The "interaction length" l_{int} determines the transition regime for the s-shaped function going from $v_{opt}(s = 0) = 0$ to $v_{opt} \to v_0$ when the distance to the leading vehicles becomes large. The "form factor" β defines the shape of the equilibrium flow-density relation. VDIFF exhibits collisions for some regimes of the parameter space.

$$a(s, v, \Delta v) = \frac{v_{opt}(s) - v}{\tau} - \lambda \Delta v$$

$$v_{opt}(s) = \frac{v_0}{2}\left[tan\left(\frac{s}{l_{int}} - \beta\right) - \tan(-\beta)\right]$$

Where:
s=distance headway,
Δv=relative velocity,
τ=relaxation time,
λ=sensitivity parameter,
v_0 =desired velocity,
l_{int}=interaction length, and
β=form factor.

A summary list of significant car-following models is shown n **Error! Reference source not found.** below:

Table 1 Compilation of Important Car Following Models

Model	equation	variables
Pipes	$A_{k+1}(t) = \frac{v_m}{T_0} G_k\left(\frac{t}{T}\right) if\ 0 < t < T_0$ $A_{k+1}(t) = \frac{v_m}{T_0}\left[G_k\left(\frac{t}{T}\right) - G_k\left(\frac{t-t_0}{T}\right)\right] if\ t > T_0$	t = Time T=Time constant ≈ 1 A_{k+1}=Acceleration of following vehicle v_m= Constant velocity of lead vehicle (cruising speed) T_0=Time for lead vehicle to reach cruising speed at a constant acceleration from rest at t=0 $G_k(t) =$ ratio of the incomplete gamma function to the complete gamma function.
GM	$a_n(t) = \alpha \Delta v_n^{front}(t-\tau_n)$	$a_n(t)$=acceleration applied by the driver n at time t α=Constant $\Delta v_n^{front}(t-\tau_n) = v_n^{front}(t-\tau_n) - v_n(t-\tau_n)$ $v_n^{front}(t-\tau_n)$=subject speed at time $(t-\tau_n)$ t=time of observation τ_n=reaction time for driver n τ_n=reaction time for driver n
GM2P	$a_n(t) = \frac{\alpha}{\Delta x_n(t-\tau_n)} \Delta v_n^{front}(t-\tau_n)$	$\Delta x_n(t-\tau_n)$ is the space headway at time $(t-\tau_n)$ α=Constant, estimated for each data set using correlation analysis

GM3P	$a_n(t) = \alpha \frac{v_n(t)}{\Delta x_n(t-\tau_n)} \Delta v_n^{front}(t-\tau_n)$	Same as GM2P
GHR	$a_f(t+T_r) = c v_f^m(t+T_r) \frac{\Delta v(t)}{\Delta x^l(t)}$	T_r =time between the observation of a certain stimulus and the reaction on that stimulus $a_f(t+T_r)$ =acceleration of the following vehicle at time $t+T_r$ $v_f(t+T_r)$ = speed of following vehicle at time $t+T_r$ $\Delta v(t)$= relative speed between the following car and the car immediately in front $(v_{leader} - v_{follower})$ $\Delta x(t)$= relative distance between following car and car immediately in front $(x_{leader} - x_{follower})$ m, l, c= parameters describing car following behavior
Wiedemann model	$AX = L_{n-1} + AXadd + RND1_n \times AXmult$ $ABX = AX + BX$ $BX = (BXadd + BXmult \times RND1_n) \times \sqrt{v}$ $v = \begin{cases} v_{n-1} \; for \; v_n > v_{n-1} \\ v_n \quad for \; v_n \leq v_{n-1} \end{cases}$ $SDX = AX + EX \times BX$ $EX = (EXadd + EXmult \times (NRND - RND2_n))$ $SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX}\right)^2$ $CX = CXconst \times (CXadd + CXmult \times (NRND1_n + RND2_n))$ $OPDV = CLDV \times (-OPDVadd - OPDVmult \times NRND)$ $AD = S_{n-1} + T_D \times v_n$ $AR = S_{n-1} + T_r \times v_{n-1}$ $AS = S_{n-1} + T_s \times v_n$ $\Delta b_m = \lvert b_{min} \rvert + a_{n-1}$ $AB = AR + \frac{\Delta v^2}{\Delta b_m}$	AX: the desired distance between stationery vehicles ABX: the desired minimum following distance at low speed differences SDX: the threshold for maximum following distance SDV: the point which the driver notice that he approach a slower vehicle. OPDV : Increasing speed difference, L_{n-1} : The leading vehicle length *AXadd* and *AXmult* are calibration parameters $RND1_n$ is a normal distributed random number for vehicle (n) v: Vehicle speed *CXconst*, *CXadd* and *CXmult*i are calibration parameters $OPDV = CLDV \times (-OPDVadd - OPDVmult \times NRND)$ *NRDV* is a normally distributed random number *OPDVadd* and *OPDVmult* are

		calibration parameters
Fritzsche	$AD = S_{n-1} + T_D \times v_n$ $AR = S_{n-1} + T_r \times v_{n-1}$ $AS = S_{n-1} + T_s \times v_n$ $\Delta b_m = \lvert b_{min} \rvert + a_{n-1}$ $AB = AR + \frac{\Delta v^2}{\Delta b_m}$	AD: desired distance threshold AR: risk distance threshold AS: The safe distance threshold AB: The risk braking distance threshold T_D : Parameter represents the desire time gap S_{n-1}: Effective length of vehicle v_n : following vehicle speed T_r : Parameter represents the risky time gap T_s : Parameter represents the safe distance gap b_{min} , a_{n-1} : Parameter controlling maximum deceleration
CA	Step 1: Determination of the randomization probability $p(v_n(t), t_h, t_s, S_{n+1}(t))$ $= \begin{cases} p_b: if\ S_{n+1} = 1\ and\ t_h < t_s \\ p_0: if\ v_n = 0\ and\ t_{st} \geq t_{k,c} \\ p_d: in\ all\ other\ cases \end{cases}$ Where $t_h = \frac{d_n}{v_n(t)}, t_s = \min(v_n(t), h_k)$, t_{st} denotes time vehicle stops. Step 2: Acceleration. Determine the speed of vehicles in the next time step. $S_n(t)$ is also taken into consideration. $S_n(t)$ is determined in Step 5 If $(S_{n+1}(t) = 0)$ or $(t_h \geq t_s)$ then $v_n(t+1) = \min(v_n(t) + a_k, v_{k,max})$ Else $v_n(t+1) = v_n(t)$ Where $v_{k,max}$ and a_k are the maximum speed and acceleration capacity of vehicles of type k Step 3 Deceleration. Set speed restriction when the vehicle in front is too close, thus locates within the effective distance (d_n^{eff}) $v_n(t+1) = \min\left(d_n^{eff}, v_n(t+1)\right)$ Step 4. Randomization If $(rand() < p)$ then $v_n(t+1) = \max(v_n(t+1) - 1, 0)$	n nth vehicle $n+1$ Vehicle in front k Type of vehicle P_b Accounting for impact of decelerating vehicle in near front P_0 Reflect the delay-to-start behaviors of vehicles stuck in traffic jam P_d Other situations P_cProbability of lane change $t_h = d_n/v_n(t)$ Time headway of nth vehicle to front h_kPreset time threshold of vehicle of type k in reflecting the effect of synchronized distance $t_s = \min(v_n(t), h_k)$ Final time threshold for initiating the consideration of the front brake light effect, taking the vehicular speed into consideration. t_{st} Accumulated time of vehicle stuck in traffic jam $t_{k,c}$ Time threshold of vehicle of type k for initiating the stop-to-start behavior $\tau^{(safe)}$ Safe time gap for collision prevention Δt Duration of an individual time step d Space headway

	Step 5. Determination of vehicle status identifier $S_n(t)$ in next time-step $S_n(t+1) = \begin{cases} 0 & if\ v_n(t+1) > v_n(t) \\ S_n(t) & if\ v_n(t+1) = v_n(t) \\ 1 & if\ v_n(t+1) < v_n(t) \end{cases}$ Step 6. Determination of time t_{sk} stuck inside the jam $t_{st} = \begin{cases} t_{st} = t_{st} + 1 & if\ v_n(t+1) = 0 \\ t_{st} = 0 & if\ v_n(t+1) > 0 \end{cases}$ Step 7 Update position $x_n(t+1) = x_n(t) + v_n(t+1)$	d_n^{eff} Effective distance of nth vehicle g Distance gap x Position of nth vehicle $\dot{x}$ Speed (1st derivative of vehicular position) $\ddot{x}$ Acceleration /deceleration (2nd derivative of vehicular position) Δx Basic unit of roadway length
SK Mode	The SK Model is a development of CA model. Instead of step 7, SK uses updating rules below $v_1 = min[v(t) + b, v_{max}, v_{safe}]$ $v_0 = v_1 - \epsilon\{v_1 - [v(t) - b]\}$ $v(t+1) = v_{ran,v_0,v_1}$ $x(t+1) = x(t) + v(t+1)$	ϵ parameter v_1 optimal velocity for next update v_{max} maximum velocity of cars v_{safe} maximum safe velocity v_{ran,v_0,v_1} a random number between v_0 and v_1
Intelligent Driver Model (IDM)	$a_n(t+\tau) = a\left[1 - \left(\frac{v_n(t)}{v^*}\right)^{\delta} - \left(\frac{\Delta x^*_{min}(v_n(t),\Delta v_{n,n-1(t)})}{\Delta x_{n-1,n}(t)}\right)^2\right]$ $\Delta x^*_{min}(v_n(t), \Delta v_{n,n-1(t)}) = d + T_{safe} * v_n(t) + \frac{v_n(t)\Delta v_{n,n-1(t)}}{2\sqrt{a^{max} * b^{\max abs}}}$	$\Delta v_{n,n-1}(t)$=approaching rate of the following vehicle v^*=desired speed T_{safe}=safe time headway a^{max}=maximum desired acceleration of following vehicle $b^{\max abs}$=absolute maximum desired deceleration of following vehicle δ acceleration component $\Delta x_{n-1,n}(t)$=distance headway d=vehicle length
Gipps	$v_a^n(t+T) = v_c(t) + 2.5\, a_n^{max}\, T\left(1 - \frac{v_n(t)}{v_n^{desired}}\right).\sqrt{0.025 + \frac{v_n(t)}{v_n^{desired}}}$ $v_n^b(t+T) = d_n^{max}.T.\sqrt{(d_n^{max}.T)^2 - d_n^{max}.\left[2\{x_{n-1}(t) -$	a_n^{max} : Maximum desired acceleration, vehicle n (m/s^2) d_n^{max} : Maximum desired deceleration, vehicle n (m/s^2) $\hat{d}_{n-1}$: Estimation of maximum deceleration desired by vehicle n-1, (m/s^2) s_{n-1} : Effective length of vehicle T: Reaction time $\hat{d}_{n-1}$: Leader desire deceleration x_{n-1} : vehicles spacing

Velocity difference model (VDIFF)	$a(s, v, \Delta v) = \frac{v_{opt}(s) - v}{\tau} - \lambda \Delta v$ $v_{opt}(s) = \frac{v_0}{2}\left[tan\left(\frac{s}{l_{int}} - \beta\right) - \tan(-\beta)\right]$	s=distance headway Δv=relative velocity τ=relaxation time λ=sensitivity parameter v_0 =desired velocity l_{int}=interaction length β=form factor

CAR-FOLLOWING MODELS AND SIMULATION SOFTWARE

Panwai and Dia (2005[9]) compared a number of car-following models between simulation software including AIMSUN, PARAMICS and VISSIM. Using data collected by an instrumented vehicle that records differences in speed and headway (leading speed, relative distance, relative speed, follower acceleration), the data was simulated using MITSIM, the Wied/Pel model, the Wied/Vis model, the Nagel/Schreckenberg model, the Optimal velocity model, and the T^3 model. The follower vehicle was programmed for modeling, and speed, time, and the distance headways of both leader and follower were captured and compared to field measurements. The leading vehicle was simulated using the GETRAM module in AIMSUN an external file in VISSIM and an API interface in PARAMICS. Performance measures and error indicators were used to assess the fitness. Results showed similarities for psychophysical models in VISSIM and PARAMICS. However, the root mean squared error and qualitative drift and goal-seeking analyses showed a substantially different car-following behavior for PARAMICS.

Siuhi (2010), Osaki (1992)[11], Subranmanain (1996[12]), and Ahmed (1999[13]) modified the GM model by separating the acceleration and deceleration responses. Table 2 compares the GM calibration parameters for acceleration and deceleration models.

Table 2. Comparison of parameter values in GM-based model for acceleration and deceleration models [10].

Parameter	**Acceleration/ Deceleration**	**Ozaki (1993)**	**Subra-main (1996)**	**Ahmed (1999)**	**Toledo (2003)**	**Siuhi (2010)**
Driver Sensitivity	acceleration	1.1	9.21	0.0225	0.0355	1.851
Driver Sensitivity	deceleration	1.1	15.24	0.0418	0.860	3.247
Speed	acceleration	-0.2	-1.667	0.722	0.291	-0.961
Speed	deceleration	0.9	1.086	N/A	N/A	1.298
Relative Speed	acceleration	1	1	0.600	0.520	0.667
Relative Speed	deceleration	1	1	0.682	0.143	1.243
Separation	acceleration	-0.2	0,884	-0.242	-0.16	0.667
Separation	deceleration	-0.9	-1.659	-0.151	-0.565	-1.544
Driver Response time lag (sec)	acceleration	N/A	1.97	N/A	N/A	0.80
Driver Response time lag (sec)	deceleration	N/A	2.29	N/A	N/A	0.70
Stimulus response threshold	acceleration	N/A	N/A	N/A	N/A	1.30

(mph)						
Stimulus response threshold (mph)	deceleration	N/A	N/A	N/A	N/A	-1.00

Siuhi and Kasvo (2010)[10] discussed findings in terms of GM calibration parameters. The study concluded the need for separate deceleration and acceleration models because the responses are impacted differently by the vehicle's speed, speed difference, and separation between the vehicles.

MODELING DRIVER BEHAVIOR

Several models aim to capture driver behavior. However, little is known about the differences in car-following behavior between driver-vehicle combinations (Ossen 2004[6]). For example, calibrating the Wiedemann Model, which captures driver indifference to small changes in the stimuli and allows different execution modes including emergency braking, requires estimating 18 parameters found in 17 different equations (Weidemann 1992[7]). Microsimulation software packages use a variety of car-following models, including Gipps' (AISUM2, SISTM, and DRACULA), Wiedemann's (VISSIM), Pipe's (CORSIM), and Fritzsche's (PARAMICS). Automated calibration parameters such as genetic algorithms have been used to calibrate the distribution of car-following sensitivity parameters (Schultz and Rilett 2004[8]).

In addition, direct correlation with real driving variables is rare, and parameterization of objective behavior is still in its infancy. Ossen et al. (2008)[4] studied the car-following behavior of individual drivers using vehicle trajectory data extracted from high-resolution digital images collected at a high frequency from a helicopter. The analysis was performed by estimating the parameters of different specifications of the GHR car-following rule for individual drivers. The results showed that measurement error has a larger influence than the component of noise. Their study pointed out that optimal parameters differ, and also that the suitability of a car-following model appears to be based on the individual driver data.

Brackstone et al. (2004) [18] used data from a series of instrumented vehicles driven by two groups. The independent variables for this study were age, approximate mileage driven per year, passivity/aggressiveness (P/A, 1–50), driver externality and internality ratings (DE and DI), and Sensation Seeking Scale V (SSSV). Among the results, the study showed that following behavior may be split in two phases (between 30 and 50 miles), and there seems to be an inverse relation between following distance and sensation seeking.

MODEL CALIBRATION WITH LONGITUDINAL TRAJECTORY DATA

Using two models of similar complexity (number of parameters), the Intelligent Driver Model (IDM) and the Velocity Difference Model (VDIFF), Kesting and Treiber (2008) [5] researched car-following behaviors of individual drivers using publicly available trajectory data for a straight, one-lane road in Stuttgart, Germany. They used a genetic algorithm to minimize the deviations between the observed driving dynamics and the simulated trajectory. One of the major findings of the study was that a significant part of the deviations between measured and simulated trajectories can be attributed to the interdriver variability and the intradriver variability (human drivers do not drive constantly over time, and their behavioral driving parameters change). The latter accounts for a large part of the deviations between simulations and empirical observations.

The results showed that the calibrated parameter values of the VDIFF strongly depend on the optimization criterion, while the IDM is more robust. Also, a deterministic car-following model allows for only an average description of human driving behavior. The authors suggest using a set of time-dependent model parameters reflecting the driver adaptation process.

Ossen and Hoogendoorn (2004)[6] analyzed trajectory data and found considerable differences between car-following behaviors of individual drivers. These differences can be expressed as different optimal parameter values for reaction time and sensitivity. Special software obtained the data from digital images captured at high frequency from a helicopter in the Netherlands. Three models from the GHR model family (Chandler, Gazis, and Edie) were analyzed, with the sensitivity parameters c and reaction times parameter T derived from accelerations, relative speeds, and distances. When individual drivers were analyzed, one model outperformed the others. However, after results for all drivers were combined, no model outperformed the others. Results show that in 80 percent of cases, a relationship between the relative speed, distance, speed of following car, and acceleration of the following car could be established.

Menneni et al. (2008)[20] presented a calibration methodology based on integrated use of microscopic and macroscopic data. Microscopic data from the Next Generation Simulation (NGSIM) data collection effort was utilized in range definition of calibration parameters and qualitative calibration of the VISSIM Wiedemann car-following model. NGSIM data was used to produce relative distance versus relative velocity graphs that contain vehicle following distances and speed oscillation.

Brockfeld et al. (2004) [21] tested the validity of different models through data collected from a Differential Global Positioning System-equipped vehicle on a test track in Japan. Data of the leading vehicle was fed into the model to compute the headway time series of the following vehicle. Deviations between measured and simulated headways were then used in model calibration and validation. Brockfeld et al. tested and independently calibrated 10 models (CA0.1, SK_STAR, OVM, IDM, IDMM, Newell, GIPPSLIKE, Aerde, PRITZSCHE, and MitSim). The calibration results showed that no best model exists and that the differences between individual drivers are larger than the differences between different models. The results also showed that although FRITZSCHE and MitSim have a large number of parameters, they do not provide better results in general.

Hoogendoorn and Hoogendoorn (2010)[22] proposed a generic calibration framework for joint estimation of car-following models. The method employed relies on the generic form of most models and weights each model based on its complexity. This new approach can cross-compare models of varying complexity and even use multiple trajectories when individual trajectory data is scarce. Prior information can also be used to realistically estimate parameter values.

DRAWBACKS OF PARAMETER CALIBRATION METHODS

Punzo and Tripodi (2007)[15] addressed the problem of the calibration of the Gipps model. Traffic stream models were developed, and stationary traffic data were used to calibrate the model. By using different combinations of microscopic parameters, the study proved that the Gipps model is unable to reproduce unstable traffic phenomena. Instead, the study generated a new model to represent multiclass traffic scenarios.

Rakha et al. (2007)[16] presented a methodology to calibrate the Gipps model for the steady-state condition. Their work assumes that all drivers have similar behaviors and characteristics. Researchers converted the car-following model into its associated macroscopic traffic stream model and calibrated key macroscopic parameters (free-flow speed, speed at capacity, capacity, and jam density) using loop detector data. These calibration methods, however, can only estimate the driver's actions on average and cannot predict the driver's actions in response to varying driving situations.

ALTERNATE MODELING OF THE DRIVING PROCESS

Hamdar and Mahmassani (2008)[23] challenge the traditional discrete-event approach of most car-following models. They use hazard-based duration models, which represent the driving process as a continuous story divided into multiple episodes. Driver behavior was modeled as "car following" and "free flow episodes." As a result, changing lanes depends on the traffic conditions and the duration of time that a driver is following a given leader; the probability of changing lanes increases during the first minute and decreases after that. The authors recommended further studies because episodes for the same driver are expected to have some interdependence and drivers will surely have some influence on one another.

Hamdar et al. (2009)[25] assessed a recently formulated hazard-based duration car-following model on its performance during congested periods. To calibrate, they applied a genetic algorithm to car-following models with complex structures, including a free-flow region and a congested region. The model was calibrated against microscopic trajectory data taken from the NGSIM data. The utility of this model is derived from prospect theory of decision-making under risk. The model also considers a penalty for the risk of accidents, an expectation value, and variance and correlation time of acceleration.

DRIVING BEHAVIOR HETEROGENEITY

Chiabaut et al. (2010)[26] studied driving behavior heterogeneity and proposed a methodology to estimate the parameters of Newell's car-following model (NCF) at a microscopic scale using I-80 NGSIM data. Driving behavior variability among drivers is appraised through the minimal spacing and congested wave velocity distributions. This method improves the existing calibration methods due to the measurement.

Ranjitkar and Nakatsuji (2010)[29], using vehicle trajectory data collected from car-following experiments conducted on a Japanese test track with RTK GPS receivers, investigated the response time of drivers. The data was further analyzed to estimate time-variant response time when considering interpersonal and intrapersonal differences in car-following behavior, such as drivers' perception time threshold in different driving conditions. Significant intrapersonal variations were observed for different speed patterns, though no definite trends were observed in such variations, emphasizing that the influence of driving conditions and intrapersonal variations in the response time cannot be ignored. It was also found that the drivers' response time under accelerating conditions is significantly different from response time under decelerating conditions.

Hamdar and Mahmassani [32] continued their studies on different driver behaviors in a simulation, instead of using the usual homogeneous traffic stream, to identify individual variations versus collective traffic patterns. The model used a utility function that assessed the change in risk when changing speeds and also the reward for increasing or reducing speed. The model was calibrated using individual trajectory data from NGSIM. Simulation was run for both homogeneous and heterogeneous traffic. The results were scattered flow-density data points that created a triangular shape. This is representative of instability and the hysteresis triangle, which are congestion dynamics.

INTRODUCING INCIDENT PARAMETERS WITH TRAJECTORY DATA

Different attempts have been made to simulate safety critical events with existing car following models. Most of the car- following models represents a safety critical event free environment. (Hamdar et al., 2008)[30]. The main factors for an accident free-environment depend on the model, varying from a sensitivity term λ (GHR model), the assumption that following vehicle will decelerate and come to a rest before hitting the leader (Gipps), the desired gap setting (IDM, IIDM), or free space to the vehicle ahead constraints (CA). Table 3 shows the safety constraints imposed by some of the models.

Table 3. Example Safety Constraints imposed in Car following models[30]

Model	Constraint
Gipps	$x'_{n-1} - s_{n-1} > x'_n$. x'_{n-1} Location where he leader vehicle will come to a complete stop x'_n Location where the follower vehicle will come to rest before hitting the leader vehicle s_{n-1} Space between the two vehicles
GHR	T □□reaction time, $\dot{x}_n(t)$□□position of a vehicle n at time t $\ddot{x}_{n+1}(t+T)$ □□position of a vehicle $n+1$ following vehicle n at time t, and λ □□sensitivity term.
CA	$v_{des} = min\left[v(t) + a_{max}, v_{max}, s_{gap}(t)\right]$ $v(t+1) = max\left[0, v_{des} - \sigma n_{ran,0,1}\right]$ $x(t+1) = x(t) + v(t+1)$ $s_{gap}(t)$ □□free space to the vehicle ahead, a_{max}□□maximum acceleration, $n_{ran,0,1}$ □□random number in the interval (0, 1), and σ□□□maximum deceleration due to the noise.
SK Model	$s_{gap}(t)$. Setting the maximum velocity equal to the space gap between two successive vehicles irrespectively of the required deceleration

IDM and IDMM	$s^*(v,\Delta v) = s_0^{(\alpha)} + s_1^{(\alpha)}\sqrt{\frac{v}{v_0^{(\alpha)}}} + T^{(\alpha)}v + \frac{v\Delta v}{2\sqrt{a^{(\alpha)}b^{(\alpha)}}}$
Wiedemann	Emergency braking mode Safety buffer term BXadd.

Xin et al., (2008)[31] proposed a model that aims to emulate "less-than perfect" everyday driving, capturing both safe and unsafe driver behavior. The datasets used include vehicle trajectories for 6 crashes and 4 near crashes in Minnesota as well as vehicle trajectories that were collected in a test track in Japan using GPS. The longitudinal driving tasks were described as a negative feedback control process between the external world and the driver-vehicle-unit (DVU). The authors assumed that the equations for the perceptual threshold of Visual Expansion Rate provide "a logical and coherent mechanism for variable reaction time". The authors determine the vehicle acceleration or deceleration action based on checking whether safety conditions are true or false. If the conditions are false the driver is in a "subconscious" driving state, without motivation for accelerating or decelerating. The situational factors do not include multi-lane effects and do not incorporate personal factors and environmental factors on driver's perception-decision process. While the idea is intriguing, it still suffers from the limitations of statistical techniques since it still uses the Gipps model.

Hamdar and Mahmassani, 2008[30] attempted to capture congestion dynamics and model accident-prone behaviors by calibrating and modifying (relaxing) seven car-following models: GHR, Gipps model, Cellular Automation (CA), SK model, intelligent driver model (IDM), intelligent driver model with memory (IDMM), and Wiedemann model. The purpose of the investigation was to relax safety term that was designed to avoid the creation of a crash limited following distance from above mentioned models and restrict the models to realistic decelerations using NGSIM data.

Chatterjee and Davis (2010)[33] used VTTI naturalistic data to better define the critical elements that lead to a rear-end crash. The results demonstrate the feasibility of using vehicle trajectory data to understand the interaction between individual vehicles. Assuming that the driver's behavior can be modeled as a piecewise constant series of acceleration that are input in the dynamic trajectory model the authors try to overcome the simple brake to stop model. Trajectories were duplicated using the initial values for the states variables and the time history of the accelerations. Major parameters were identified where the piecewise constant acceleration, the points where the driver changes between different accelerations, the reaction time of the driver, and the critical headway. To estimate the parameters they used a Bayesian analysis (Markov Chain Monte Carlo), and the numerically equations were solved using WINBUGS.

REINFORCEMENT LEARNING METHOD

Reinforcement learning is a sub-area of machine learning in computer science concerned with how an agent is supposed to take actions in an environment so as to maximize the notion of long-term reward. The objective of reinforcement learning algorithms is to find to a policy that maps out states to the actions the agent ought to take in those states. State in our case can be time from trigger, speed, car following distance, yaw angle, and environment. Reinforcement learning reinforces agent actions when they perform approximately close to naturalistic actions and penalize actions which are far away from them. The only information available for learning is the

system feedback, which describes in terms of reward and punishment on the task the agent has to realize. At each time step, the agent receives a reinforcement signal according to the last action it has performed in the previous state. The problem involves optimizing not only the direct reinforcement, but also the total amount of reinforcements the agent can receive in the future. Finally, reinforcement learning should extract driving rules from naturalistic dataset and establish a similar driver specific state action mapping rules.

Reinforcement learning method has been applied in two parts of transportation problems mostly, in finding optimal traffic control policy for intersection controllers and modeling driver behavior especially travel time and route choice I network simulation or optimization ptoblems. There is also a successive study in modeling helicopter controller through learning from an expert pilot to simulate the trajectory and movement of helicopter control in Abbeel, 2006.

REINFORCEMENT LEARNING USED IN TRAFFIC ADAPTIVE CONTROL SYSTEM

Optimization of traffic signals for efficient movement of traffic on urban streets is a challenging part of an urban traffic-control system. Adaptive system control has been used to reduce delay and congestion through an array of traffic control and management strategies (such as SCATS). With the ability to exert real-time, adaptive control over a transportation process is useful for a variety of intelligent transportation systems services, including control of a system of traffic signals, control of the dispatching of paratransit vehicles, and displays a dynamic route guidance system (Abdulhai, 2003)[49].

The objective of reinforcement learning algorithms applied in adaptive control is through mapping states to actions, agent controllers are able to get optimal policies according to traffic environment changes. Optimal policies agents are seeking are policies which can get smallest queuing length, number of stops per vehicle.

Abdulhai, (2003)[51] designed a test bed and train controllers by using Q-learning in a single, isolated intersection consisted of a simulated two-phase signal controlling the two two-lane roads. The isolated traffic signal test-bed consisted of a simulated two-phase signal controlling the intersection of two-lane roads. Vehicle arrivals were generated using Poisson processes with average arrival rates on each of four approaches. Q-learning agents control each of the four approaches with two phases control. State information available to the agent include queue lengths and elapsed phase time. From simulation results, the performance of networked signal systems and integration with dynamic route guidance has improved. One restriction of the method is that their adaptive control method used an intersection without any turning vehicle, which is not compatible with field data.

Jacob and Abdulhai, (2006)[52] continued traffic system control on a corridor control bases on their previous study (Abdulhai, (2003)[51]). They proposed an automated adaptive traffic corridor control using reinforcement learning to develop a self-learning adaptive integrated freeway-arterial corridor control for both recurring and nonrecurring congestion. Q-learning approach is used to provide an optimal control for a freeway corridor to determine the appropriate actions chosen from staying on the current metering rate, increase red time, decrease red time and decrease red phase. Simulation software PARAMICS was used to train and evaluation agent in an offline model within a simulated environment.

Bingham (2001) [53]applied an actor-critic reinforcement learning algorithms in neuro-fuzzy traffic signal control system with the purpose of minimize vehicular delay caused by the signal control policy. The controller receives measurements of incoming traffic and chooses green signal length accordingly. A neural network adjusts the fuzzy controller by fine-turning the form and location of membership functions. Two phases was considered for any intersection. Reinforcement learning in neural network gives credit for successful system behavior which will be chosen more often in the. In simulation experiments, the simulations were run at several different traffic volumes and traffic detector location. The result shows that different membership functions are found optimal at different traffic situations. The advantage of fuzzy control systems over traditional ones is their ability to use expert knowledge in the form of fuzzy rules and small number of parameters needed. However, intersections usually have four approaches and eight movements, which makes this reinforcement learning approach less persuasive.

Choy et al. (2003) [54]formed a multiagent architecture for real-time coordinated signal control in an urban traffic network. Three hierarchical layers of controller agents exist: intersection, zone and regional controller. Fuzzy logic, neural network and regional controllers are implemented to agents. Each individual controller agent recommends an appropriate signal policy at the end of signal phase. An online reinforcement learning module is used to update the knowledge base and rules of agents based on online data with the objective of minimizing average delay to zone. 25 signalized intersections in a microscopic simulator were implemented in a network with test results shows that the muliagent system improved average delay and total vehicle stop time compared with fix-time traffic signal control. The main contribution of this paper is the formulation of multi-agent based architecture and online reinforcement learning module. It has shown that agent can coordinate its local goal with zone and regional objectives autonomously.

Adam et al. (2009)[55] proposed a Q-learning method applied in reducing the number of vehicles travelling in dilemma zone. Algorithms used arbitrary policies to determine terminating green or not determined by vehicles in dilemma zone and whether dilemma zone has not cleared after a period of time. A control agent develops an optimal policy by learning from number of vehicles in dilemma zone. The optimal policy considerate number of vehicles trapped in dilemma zone after taking action green extension or not. Unlike arbitrary and stationary policies derived by using existingl approaches, the control policy adapts to changes in volume. Reinforcement learning based policy reduced number of vehicles caught in dilemma zone by up to 32 percent in experimental simulation framework VISSIM.

REINFORCEMENT LEARNING IN TRANSPORTATION NETWORK PROBLEMS

Bogers et al. (2007) [56]focused on route choices considering two types of learning: reinforcement based learning and explicit belief-based learning (including memory decay). A model that captures learning types from parameters calibrated from a large data set from experimental research is developed. The model uses a Markov formulation for updating daily, based on a person's belief about travel time. Reinforcement is modeled by including the latest 10 route choices in the model. Results indicate that 20 percent of perceived travel time is from the most recent experience. Furthermore, the reinforcement part of the model can make up a significant part of the route utility and therefore should be a standard component in route choice models.

Arentze and Timmermans, (2003) [57]developed a framework for modeling dynamic choice based on a theory of reinforcement learning and adaption since individuals develop and continuously adapt choice rules while interacting with the environment. A reward function, incremental action value functions and action selection methods, are presented in this activity based analysis and memory and search play a key role in reinforcement learning. The reinforcement learning concept assumes that actions which produce positive rewards if they are reinforced and have a higher probability of being repeated in future choice situations under similar conditions and actions with negative outcomes tend to be avoided. The system assumes multi-stage decision making in potentially very large condition spaces and can deal with stochastic, non-stationary, and discontinuous reward functions. A hypothetical case is considered that combines route, destination, and mode choice for an activity under time-varying conditions of the activity schedule and road congestion probabilities. The learning and adaption is robust and most successful if the system chooses a realistic reward reference level and assigns only a modest weight to new rewards in updating action values. The present framework takes exploration and adaption in choice behavior into account but the framework does not take into consideration environment effects.

Han and Timmermans, (2006)[58] proposes a general model of interactive learning behavior that evolves toward equilibrium in strategic situations under the assumption that travelers may take particular choice options by taking into account their expectations about behavior of other travelers. The properties of the model are examined by using numerical computer simulations. The results of the simulations support the face validity of the formulated model.

Wahba and Shalaby (2006) [60]propose an operational prototype of an innovative framework for the transit assignment problem by a learning-based approach. The proposed framework uses a basis of representing a passengers learning and decision-making activities. In this prototype a hypothetical transit network will consist of 22 routes and 194 stops have been developed within a micro simulation platform (Paramics). They then generated 3000 passengers synthesized to model the transit assignment process in the peak morning periods. Reinforcement learning is used to model passengers' adaption and account for the differences in passengers' preference in the dynamics of the transit network. Using reinforcement learning to represent passengers' adaptation and accounting for differences in passengers' preferences and the dynamics of the transit network, the prototype has demonstrated that the proposed approach can simultaneously predict how passengers will choose their routes and estimate the total passenger travel cost in a congested network as well as loads on different transit routes.

REINFORCEMENT LEARNING APPLIED IN TRAFFIC KINEMATIC PROBLEM

Abbeel et al[61], presents the first successful autonomous completion of four aerobatic maneuvers: forward flip, sideways roll at low speed, tail-in funnel, and nose-in funnel using a real RC helicopter. First, a pilot controls the helicopter to find a helicopter dynamics model and a reward (cost) function. Then a controller optimized for the result model and reward functions are formulated using the reinforcement learning (RL) algorithm. Differential dynamic programming (DDP) is used in this optimization process.

During the flip, the helicopter rotates 360 degrees forward around a lateral axis. In a right axial roll, the helicopter rotates 360 degrees around its longitudinal axis. In a tail-in funnel, the helicopter repeatedly flies in a circle sideways with tail pointing to center of the circle. The Nose-in funnel is similar to tail-in funnel with the exception of the nose pointing to the center of the circle.

The helicopter dynamics are modeled by the apprenticeship learning algorithm method. First, the data is collected from a human pilot flying the desired maneuvers in a helicopter to learn the optimal model. Next, a controller is found to simulate a flight based on the current model test control on the helicopter. The helicopter state includes position (x, y, z), orientation, velocity $(\dot{x}, \dot{y}, \dot{z})$, and angular velocity $(\omega_x, \omega_y, \omega_z)$. The helicopter is controlled by four separate actions: cyclic pitch (u_1, u_2), tail rotor (u_3) controls, and collective pitch angle (u_4). This paper formulates the equations for acceleration and then integrates the accelerations together to obtain helicopter states. (Superscripts b indicate the body coordinate for helicopter, coefficients A, B, C, D and E are estimated from helicopter flight data, ω on the right hand side are zero mean Gaussian random variables).

$$\ddot{\boldsymbol{x}}^{\boldsymbol{b}} = \boldsymbol{A_x}\dot{\boldsymbol{x}}^{\boldsymbol{b}} + \boldsymbol{g_x^b} + \boldsymbol{w_x}$$
$$\ddot{\boldsymbol{y}}^{\boldsymbol{b}} = \boldsymbol{A_y}\dot{\boldsymbol{y}}^{\boldsymbol{b}} + \boldsymbol{g_y^b} + \boldsymbol{D_0} + \boldsymbol{w_y}$$
$$\ddot{\boldsymbol{z}}^{\boldsymbol{b}} = \boldsymbol{A_z}\dot{\boldsymbol{z}}^{\boldsymbol{b}} + \boldsymbol{g_z^b} + \boldsymbol{C_4 u_4} + \boldsymbol{E_0}\left\|(\dot{\boldsymbol{x}}^{\boldsymbol{b}}, \dot{\boldsymbol{y}}^{\boldsymbol{b}}, \dot{\boldsymbol{z}}^{\boldsymbol{b}}\right\|_{\boldsymbol{2}} + \boldsymbol{D_4} + \boldsymbol{w_z}$$
$$\dot{\boldsymbol{\omega}}_{\boldsymbol{x}}^{\boldsymbol{b}} = \boldsymbol{B_x}\boldsymbol{\omega}_{\boldsymbol{x}}^{\boldsymbol{b}} + \boldsymbol{C_1 u_1} + \boldsymbol{D_1} + \boldsymbol{w_{\omega_x}}$$
$$\dot{\boldsymbol{\omega}}_{\boldsymbol{y}}^{\boldsymbol{b}} = \boldsymbol{B_y}\boldsymbol{\omega}_{\boldsymbol{y}}^{\boldsymbol{b}} + \boldsymbol{C_2 u_2} + \boldsymbol{C_{24} u_4} + \boldsymbol{D_2} + \boldsymbol{w_{\omega_y}}$$
$$\dot{\boldsymbol{\omega}}_{\boldsymbol{z}}^{\boldsymbol{b}} = \boldsymbol{B_z}\boldsymbol{\omega}_{\boldsymbol{z}}^{\boldsymbol{b}} + \boldsymbol{C_3 u_3} + \boldsymbol{C_{34} u_4} + \boldsymbol{D_3} + \boldsymbol{w_{\omega_z}}$$

State variables: Helicopter position x, y, z, velocity $\dot{x}, \dot{y}, \dot{z}$, angular velocity $\omega_x, \omega_y, \omega_z$
Actions: u_1, u_2 cyclic pitch, u_3 tail rotor controls, u_4 collective pitch angle
Coefficients A, B, C, D and E are estimated from helicopter flight data
ω Zero mean Gaussian random variables
(Superscripts b indicate the body coordinates for helicopter).

The reinforcement learning method has a sextuple of components$(S, A, T, H, s(0), R)$. The set of states, S, the set of actions or inputs, A, the dynamics model, T which is a set of probability distributions $\{P_{su}^t(s'|s, u\}$. The probability of being in state s' at time $t + 1$ given the state and action at time t are s and u. $s(0)$ is the initial state and R is the reward function. A policy $\pi = (\mu_0, \mu_1, \dots, \mu_H)$ is the relation from the actions to the states. The expected sum of the rewards when acting the following policy π is given by the equation$E[\sum_{t=0}^{H} R\big(s(t), u(t)\big)|\pi]$. The optimal policy π^* is the policy which maximizes the expected sum of the rewards and is given as:

$$\boldsymbol{\pi^* = argmax_\pi\, E}\left[\sum_{\boldsymbol{t=0}}^{\boldsymbol{H}} \boldsymbol{R\big(s(t), u(t)\big)|\pi}\right]$$

π^*, Optimal policy
$s(t), u(t)$, state and action in time t
$R(s(t), u(t))$, Reward function according to policy π when taking action $u(t)$ at state $s(t)$

The RL method solves this linear quadratic regulator (LQR) control problem as: $s(t+1) = A(t)s(t) + B(t)u(t) + w(t)$. The reward for being in the state $s(t)$ and taking the action/input $u(t)$ is given by the function: $-s(t)^T Q(t)s(t) - u(t)^T R(t)u(t)$, where $Q(t), R(t)$ are the positive semi-definite matrices which parameterize the reward function.

In the differential dynamic programming (DDP) design for solving LQR, the error state is defined as : $e = (\dot{x^b} - (x^b)^*, \dot{y^b} - (y^b)^*, \dot{z^b} - (z^b)^*, x - x^*, y - y^*, z - z^*, \dot{\omega_x^b} - (\omega_x^b)^*, \dot{\omega_y^b} - (\omega_y^b)^*, \dot{\omega_z^b} - (\omega_z^b)^*, \Delta_q)$

Helicopter position x, y, z, velocity $\dot{x}, \dot{y}, \dot{z}$, angular velocity $\omega_x, \omega_y, \omega_z$

DDP is used in the first phase of control design to find the open-loop input sequence that would be optimal in a noise-free setting. During the second phase, the DDP is applied to design the actual flight controller with a deviation from the nominal open-loop input sequence with defined inputs.. The reward function contains 24 features, including the squared error state variables, the squared inputs, and the squared change in the inputs between the consecutive time steps and the squared integral of the error state variables. The reinforcement learning algorithm finds the controller using a method of apprenticeship learning through the inverse reinforcement learning algorithm. The inverse reinforcement learning system provides reward weights that are closer to an expertise of a human pilot by increasing or decreasing the weight of the features deviant from the expert.

In the flip experiment, the cost matrices Q and R in the reward function are chosen by hand. The initial controller is oscillated in reality. The controller performs well with an increase in penalty for the changes in the inputs through consecutive time steps. The controller uses the same cost matrices for flips as it does when performing a roll. The controller outperforms the expert human pilots when it comes to both nose-in funnels and tail-in funnels. The DDP-based controller flies helicopter successfully but only after being penalized for the controllers rapid change in actions over consecutive time steps. In conclusion, this paper uses the apprenticeship learning algorithms to determine the reward function and to learn the model. This is a two-phase controller design, first for a feasible trajectory and second for design of the actual controller.

Although this idea looks very similar to our study, the experiment is not applicable in our problem. In their approach, the agent controller is under online training by reinforcement learning algorithm. However, in our problem, we do not have the luxury to design an agent controller to online control a vehicle in the field and collect the data. Instead, naturalistic trajectory is the only information we can get access to. In fact, offline training is the only option.

Jouffe [62] proposed a realistic approach to deal with various continuous traffic state-action mapping problems which is similar to our study. In their paper, two reinforcement learning methods Fuzzy Actor-Critic Learning (FACL) and Fuzzy Q-Learning (FQL) used to tune online the conclusion part of Fuzzy Inference Systems. The only information available for learning is the system feedback, which describes in terms of reward and punishment the task the fuzzy agent has to realize. At each time step, the agent receives a reinforcement signal according to the last action it has performed in the previous state. The problem involves optimizing not only the direct reinforcement, but also the total amount of reinforcements the agent can receive in the future. To illustrate the use of these two learning methods, the authors first applied them to a problem that

involves finding a fuzzy controller to drive a boat from one bank to another, across a river with a strong nonlinear current. Then, the well-known Cart-Pole Balancing and Mountain-Car problems are used to be able to compare the proposed methods to other reinforcement learning methods and focus on important characteristic aspects of FACL and FQL. The genericity of the methods allows agent to learn every kind of reinforcement learning problem (continuous states, discrete/continuous actions, various type of reinforcement functions). The experimental studies also show the superiority of these methods with respect to the existing methods in the literature.

Naturalistic Data – 100-Car Database

Another notable naturalistic data collection study (mentioned briefly above), the 100-Car Naturalistic Study, (Dingus et al. 2006) conducted through VTTI was a light-vehicle naturalistic study in which 100 light vehicles were instrumented. The 100-Car study was unique in that it was the first study conducted where vehicles were instrumented with the primary goal of collecting large-scale, naturalistic driving data. Also, 78 of the 100 instrumented vehicles were those of the participants and the participants were given no special instructions. Approximately 2 million vehicle miles were included in the data set which is rich with extreme cases of driving behavior and performance. This is equivalent to 43,000 hours of driving data.

Naturalistic Data – 34-Truck Database

The Drowsy Driver Warning System Field Operational Test (DDWS FOT) (Hanowski et al. 2008)conducted by VTTI was a naturalistic data collection study, in which data was collected for 18 months from 103 commercial motor vehicle drivers (CMV) from three different fleet companies. This database yielded approximately 2.2 million driving miles, once reduced resulting in 1,217 valid safety-critical events. The data were processed through software that flags potential events of interest based on trigger threshold values (Olsen et al., 2008).

Naturalistic Data – 8-Truck Database

Similarly, VTTI performed another large-scale naturalistic data collection study investigating crash risk by identifying safety-critical events. The Naturalistic Truck Driving Study's (NTDS) (Olson et al. 2009) and Blanco et al. (in press, not available) objective was to use the naturalistic driving data to investigate issues in the trucking industry such as driver sleep/rest cycles and crash countermeasures. Continuous driving performance data was collected for 100 commercial motor vehicle drivers from four different fleet companies during 4 months of their normal driving routine for four weeks per driver. Approximately 14,600 hours of driving data were collected and 735,000 driving miles. During this period of time over 2,800 safety-critical events were identified: 13 crashes, 58 near crashes, and 1,595 crash-relevant conflicts, and 1,213 unintentional lane deviations.

Driver Behavior Research using Naturalistic Data

Fitch et al. (Fitch et al. 2009) conducted an examination of driver behavior leading to lane-change crashes and near-crashes using the database created from the 100-Car Study. Five objectives were investigated in this analysis: classifying the types of lane change events (e.g., severity and direction of lane-change events), and analyzing striking vehicle drivers' behaviors,

as well as the struck vehicle drivers' behavior and the surrounding vehicles behaviors. 135 lane-change events were identified from the previous study and used in this current analysis. The lane-changes were classified using 3 event scenarios. It was found that 85 percent of drivers used their turn signals during planned left-lane changes, while 24 percent used their turn signals when making unplanned left-lane changes to avoid a forward crash threat. Amidst the statistical findings of this analysis it was found that driver's are limited in that they can visually attend to only one location at a time so safely monitoring one's surroundings may be difficult in heavy traffic conditions and the use of a system that assists drivers in perceiving adjacent vehicles and recognizing crash threats while concurrently monitoring the forward roadway may mitigate this human factors dilemma.

Another study examining lane-change behavior conducted by Lee, et al. (Lee et al. 2004)explored the nature and severity of lane changes in a naturalistic driving environment. Sixteen commuters who normally drove more than 25 miles were recruited to drive one of two instrumented research vehicles for ten days each. Half of the participant's commutes were by interstate and half commuted on a U.S. highway. Lane change identification was specified by review of the video finding the initiation (first lateral move by vehicle) and end points (vehicle settled in the destination lane) for each lane change. The lane change was then categorized by maneuver type, direction, severity, urgency, and success/magnitude. Eleven categories of maneuver type were identified, including slow lead vehicle, return, enter, and exit/prep exit. During the first data pass-through all lane changes were identified, graded, and classified totaling 8,667 lane changes categorized into one of the 11 maneuver types. The analysis of the full data set resulted in many interesting findings regarding frequency, duration, urgency and severity of lane changes in regard to maneuver type, direction, and other classification variables. A sub-set of the full data set consisting of 500 lane changes was then analyzed using sensor data collected from the instrumented vehicles. Additional variables were investigated in-depth for this additional sample and included:

- Steering,
- Lateral acceleration,
- Velocity,
- Braking,
- Turn signal use,
- Eye glance location probability,
- Eye glance link value probability,
- Mean single glance time, and
- Distance to forward and rearward POV's
- TTC to forward and rearward POV's

The sampled lane changes were more severe and were cases where a lane change collision avoidance system (CAS) was likely to help the greatest. The data from this study was used to provide recommendations for designers of lane change CAS as far as display location and activation criteria are concerned. The research proved to provide valuable insight into the behaviors and parameters associated with lane changes, as well as the archived data having the potential to answer other questions related to driver behavior.

McLaughlin et al. (McLaughlin et al. 2009a) explored the use of real crash data collected in a naturalistic driving study to investigate the potential of collision avoidance systems in avoiding rear-end crashes. For this research effort, three previously proposed collision avoidance algorithms were modeled in software and real crash and near-crash data were input into the algorithm models. Real-time data from 13 rear-end crashes and 70 rear-end near-crashes

were selected from the 100-Car Driving Study. Thirteen crashes from the original dataset were used and sixty of the rear-end near-crashes were selected randomly from approximately 400 rear-end near-crashes. An additional 10 near-crashes were included, representing cases where a rear-end crash was avoided through the driver departing the lane to avoid colliding with an LV.

Once these events were selected, the data was prepared for further analysis and put into models of CAS. The three CAS algorithms selected and modeled for evaluation where:

1. Equations developed by Knipling et al. (Knipling et al. 2004) for LV-stationary (LVS) and LV-moving (LVM) scenarios.
2. CAMP Linear – The linear regression approach described in early CAMP work (Kiefer et al. 1999) that predicts a required deceleration after response based on test-track braking by drivers in different scenarios.
3. NHTSA – An algorithm developed by Brunson et al. (S.J. Brunson et al. 2002)that incorporates multiple warning levels and sensitivity settings.

When data from the events were put into the algorithm models, a time-series output was generated that could be overlaid on the real event data. Once the warnings were generated by the algorithms, an evaluation was done to estimate the percentage of the driving population who would be able to respond to the warning in time to avoid a collision based on kinematic estimates of the braking needed to avoid that collision. The methodology provided useful guidance both estimating benefits achieved by the algorithms and in estimating the frequency of alerts in normal driving situations. The algorithms tested appear to generate alerts at higher rates than would be acceptable. Results based on the algorithm with the lowest alert frequency indicated that approximately 20 to 25 percent of drivers would avoid collisions similar to those tested if a 0.5 deceleration is executed in response to the alert.

Fitch et al. (Fitch et al. 2008) used naturalistic data collected by VTTI to study the safety benefits that may be obtained by deploying a forward collision warning system for heavy vehicles using kinematic motion equations and Monte Carlo simulation. The approach taken involved simulating driver forward collision avoidance behavior, with and without FCW alarms, in response to rear-end (RE) conflicts recorded in a previous naturalistic driving study. The naturalistic driving dataset used for this study was the DDWS FOT because the driving data were continuously collected offering a repository that could be mined to investigate other driver phenomena. RE conflicts were identified using methods that were based on Volvo's Intelligent Vehicle Initiative Field Operation Test , which observed heavy vehicles operating with an FCW system for one year. Since components of the same FCW system were used in both the Volvo and DDWS FOT's this study was initiated with the purpose of calculating potential safety benefits using the DDWS data by following the methodology the Battelle study performed on the Volvo FOT data. **Figure 4** provides an overview of the steps performed in this study.

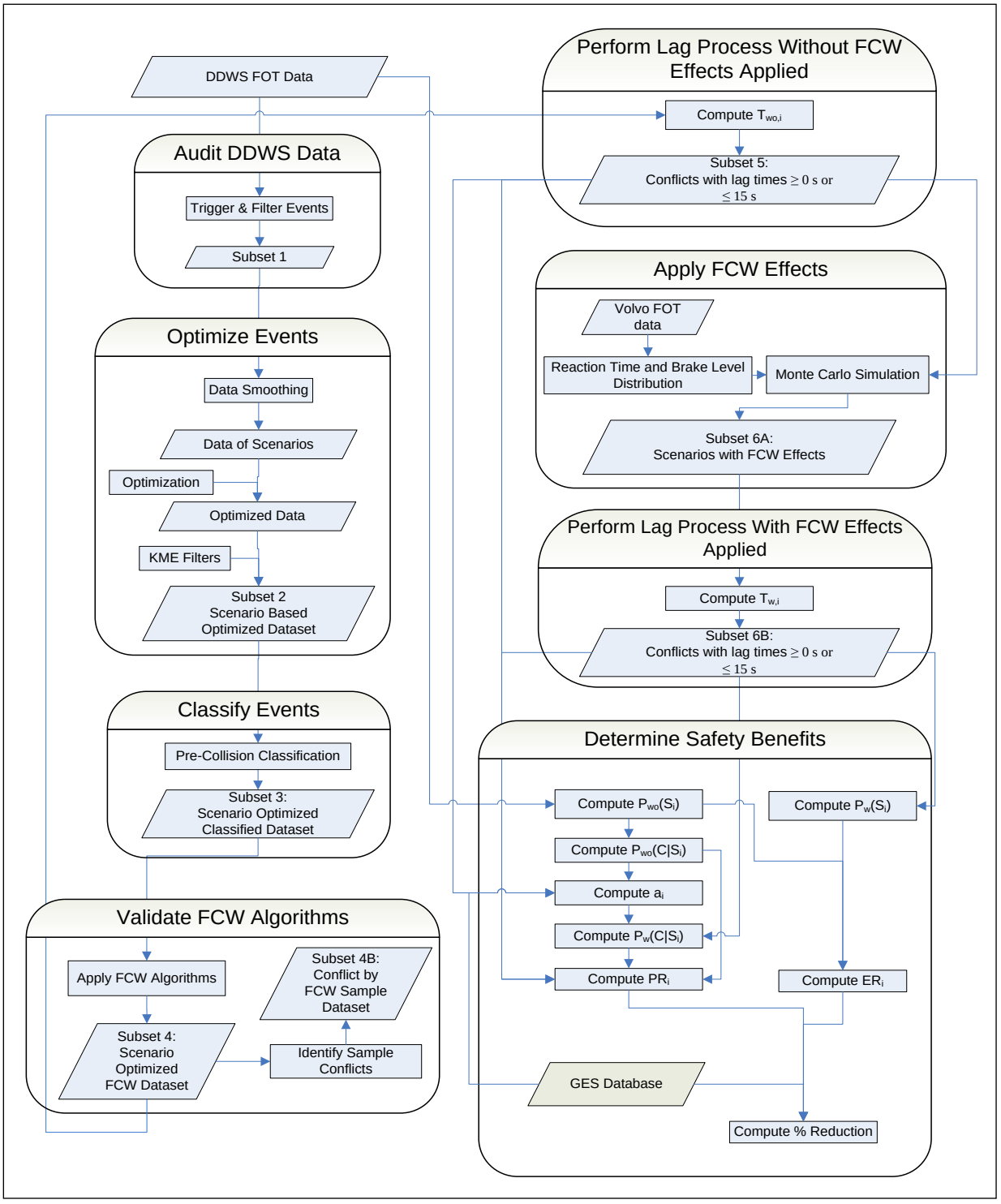

Figure 4. Data Flow for Safety Benefits Analysis(Fitch et al. 2008)

The algorithms used by the Eaton VORAD FCW system were applied to the identified RE conflict data. The auditory alarm severity and timing of the FCW alarms (that would have occurred had an FCW system been installed) were computed. Drivers' actual alarm perception-response times and braking-levels that were measured during the Volvo FOT were then used to

simulate driver response behavior to the theoretical FCW alarms using a Monte Carlo approach. An assumption was made that drivers made the best decision when receiving the FCW alarms, allowing them to apply the brakes sooner and possibly avoid a crash. Enhancing the approach used in Battelle, the number of conflicts avoided, as well as the additional response time available prior to encountering a crash, were both used to compute a prevention ration (PR) and exposure ratio (ER). The PR is a measure of drivers' ability to avoid a crash once they have encountered an RE conflict, while the ER is a measure of drivers' ability to avoid a conflict scenario altogether. The PR and ER were then combined to compute an overall crash reduction estimate.

This simulation determined that a nationwide deployment of FCW systems in heavy vehicles could reduce the number of RE crashes by 21 percent ($p < 0.001$). This study used a regression analysis to calculate the perception response time (PRT) which did limit the representation of individual drivers' behavior. Conflicts were said to exist when a following vehicle (FV) decelerated more than 3 m/s^2 (10 ft/s^2) to avoid a leading vehicle (LV) traveling in the same lane. The simulation modeled the drivers' course of action by selecting their optimum decisions (safest response) to the various alarm levels. The Monte Carlo simulation involved randomly selecting driver perception-response time (PRT) and deceleration levels from distributions generated using the Volvo FOT data.

An investigation was conducted by McLaughlin et al (McLaughlin et al. 2009b) using the 100-Car Naturalistic Driving Study database to identify factors associated with run-off-road (ROR) crashes. More specifically, investigating the ROR crash or near-crash events to identify conditions in which these events occurred and the contributing factors associated with the events. Crashes and near-crashes from the dataset that were marked as road departures or driver reactions other than braking were included in the ROR analysis. These events were reviewed and included in the analysis if the subject vehicle contacted or crossed a roadway boundary or a boundary defining direction of travel, or if a rapid, severe evasive maneuver was required to avoid departing the roadway or entering an oncoming traffic lane. In addition to crash and near-crash events, the dataset was used to develop a description of baseline driving. A stratified random review of the entire dataset was conducted to develop a baseline database that quantified the frequency with which different conditions and behaviors are present while driving. With the criteria identified, 122 ROR-related events were chosen for the investigation. These 122 ROR events were analyzed in order to discover the contributing factor of the event. The figure below illustrates the percentage of ROR events by contributing factor (**Figure 5**).

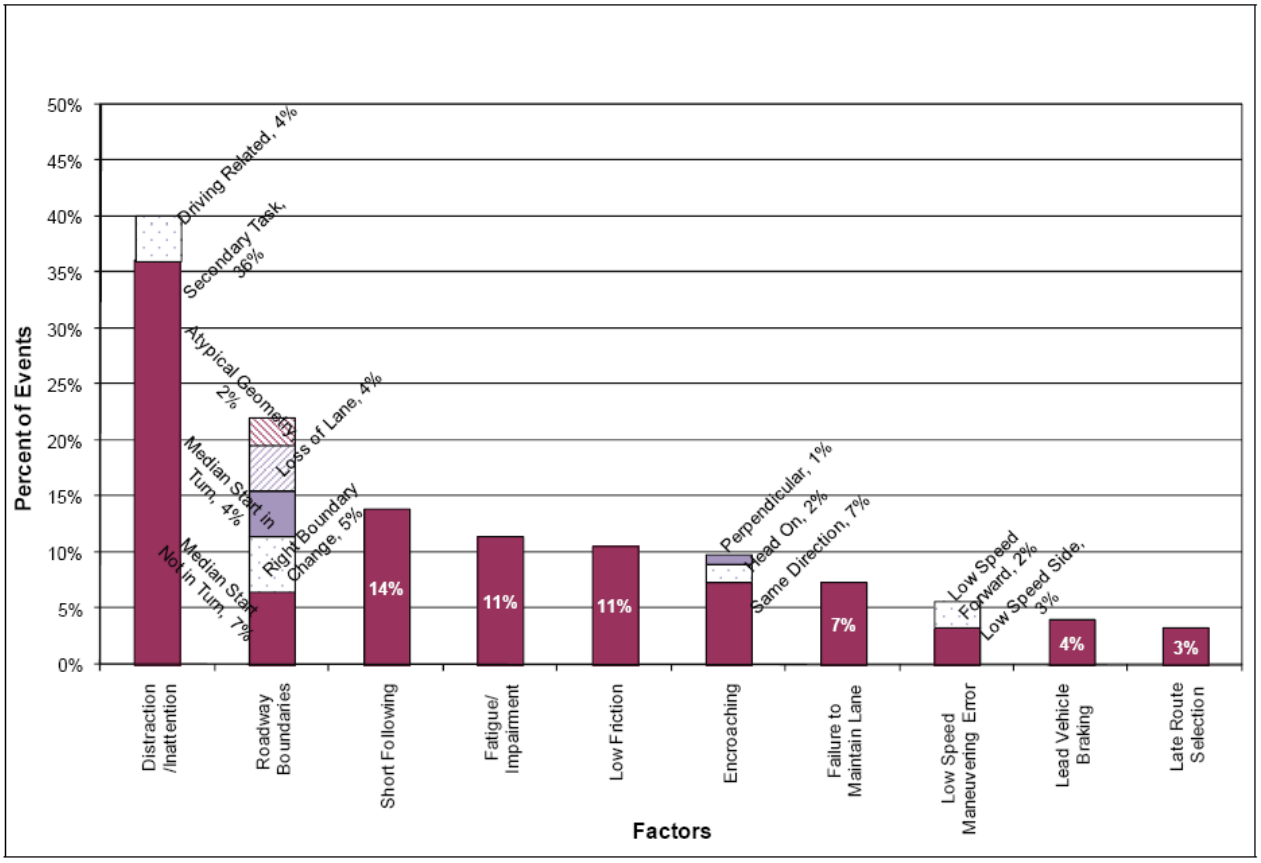

Figure 5. Percentage of ROR Events by Contributing Factors(McLaughlin et al. 2009b)

The process to identify the contributing factors required review of each event using data visualization software that allowed viewing the video frame-by-frame along with a graphical presentation of vehicle and kinematics measure such as speed, yaw rate, for example, all synced to video. An additional approach was used in the identification process based on the fact that accidents are frequently caused by multiple factors. Based on the idea that accidents are a result of the convergence of multiple factors a small number of contributing factors were also identified. In 75 percent of the ROR events, a single factor was identified as the contributing event and in 25 percent of the cases multiple factors were identified. In 22 percent of the ROR events, two factors are identified and in the remaining 3 percents of the events, three factors were identified.

The most frequently identified contributing factor of the ROR events was distraction/inattention (40 percent of ROR events), with 36 percent of the events involving secondary task distraction and 4 percent involving driving-related inattention to the forward roadway. Other factors included changes in roadway boundaries, short following distances, and lead vehicle braking. It was found that ROR events occur more frequently per mile in low visibility and low friction conditions than in clear and dry conditions.

The Naturalistic Data that is relevant to this project can be divided into two main categories: Safety Critical Events and Car Following Periods. The car following periods are representative of safe driving where one vehicle follows another without any conflicts arising. The safety

critical events are the cases where conflicts arise between two vehicles and evasive maneuvers or braking is taken as an action in order to avoid a collision.

Chapter 3. NATURALISTIC DATA PROCESSING

Naturalistic data collection is the collection of driver behavior and performance data in a natural environment. Naturalistic data collection has opened the door for tremendous possibilities in transportation research. It has allowed researchers to examine what happens in the final seconds before crash, near-crash, and safety critical events for which researchers would otherwise depend on eye witness accounts and police reports. Handled properly, data regarding vehicle speed, acceleration, range, range rate, headway, time to collision (TTC), brake pedal input, and qualitative data such as pre-incident maneuvers can be used to describe driver behavior. Qualitative data such as roadway type, number of lanes, traffic density, time of day, and weather can be used to describe the driving environment. Tying these data together allows researchers to understand the conditions that exist during event and attain baseline data. Naturalistic driving data collection is a powerful approach used by researchers to understand crash and near-crash causation. As opposed to traditional epidemiological and experimental/empirical approaches, this *in situ* process uses drivers who operate vehicles that have been equipped with specialized sensors as well as processing and recording equipment (**Error! Reference source not found.**). In effect, the vehicle becomes the data collection device. The drivers operate and interact with these vehicles during their normal driving routines while the data collection equipment continuously records numerous items of interest during the entire driving epoch (a prescribed driving period).

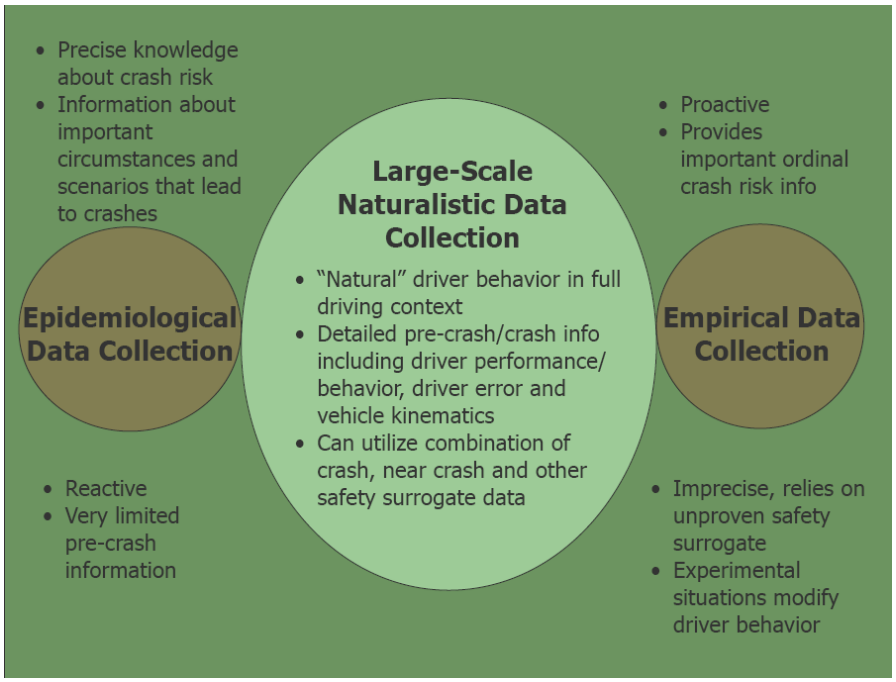

Figure 6. Advantages of the naturalistic data collection methodology data collection and reduction.

Naturalistic data collection methods require a sophisticated network of sensors, processing, and recording systems. The Data Acquisition System (DAS) provides a diverse collection of both on-road driving and driver (participant, non-driving) data, including measures such as driver input and performance (e.g., lane position, headway, etc.), four camera video views, and driver activity data. This information may be supplemented by subjective data, such as questionnaire data. A tremendous amount of data is acquired when carrying out these naturalistic studies.

Due to the tremendous amount of information collected during these procedures, naturalistic data collection methods require significant post-collection processing. VTTI processes this data using secure computing systems on isolated networks, which allows for both data processing and analysis. A storage area network is employed to store the data and to play multiple driving and video files simultaneously at several PC workstations.

Typically, the first step in the data reduction process is to identify events of interest, including crashes, near-crashes, and crash-relevant conflicts. To find events of interest, VTTI has developed a software program, Data Analysis and Reduction Tool (DART), that scans the dataset using user-defined threshold algorithms to identify notable actions (e.g., hard braking, quick steering maneuvers, short times-to-collision [TTC], and lane deviations (including median encroachments). All detected events are analyzed based on "instant replays" of video data and associated dynamic data recordings of the events. This analysis captures both the observable causal sequences leading to events as well as the conditions and correlates of event occurrence. **Figure 7** presents a screen shot from the DART software interface.

A = Plan view of subject vehicle and road position based on Road Scout lane tracker.
B = Video feed recording face shot of driver, with frame number, used to synchronize video and parametric data.
C = Video feed recording forward view.
D = Video feed recording of over-the-shoulder view of driver, which is used for determining driver distractions.
E = Video feed recording both rear-left and rear-right images.
F = Data feed from X/Y Acceleration sensor indicating longitudinal acceleration as measured in g's (the range to primary target as determined by VORAD).
G = Data feed from front VORAD sensor, indicating range to primary target as determined by VORAD, versus time.
H = Network speed is the vehicle's speed.
I = Data feed showing VORAD primary target closing rate (to subject vehicle).

Figure 7. Screen shot of DART software interface.

Table 4 identifies the DART components, tracking/recording mechanisms, and resulting data products. Using threshold values (triggers) developed by VTTI, the DART system was used to identify the different events.

Table 4. Data Analysis and Reduction Tool (DART) Components

Sensors	Vehicle Network	Incident Box	Video Cameras	On-Board Components	Data Products
• GPS • lane tracker • x/y acceleration • front VORAD • RF sensor • seat acceleration • sound level meter • in-cab temperature	• speed • distance • ignition signal • brake activation • turn signals	• light level • incident pushbutton • microphone	• face • forward • rear-left • rear-right	• data reliability check software	• truck performance data file (in .dat format) • digital video file (in .mpg format) • digital audio file (in .mp3 format)

The Drowsy Driver Warning System Field Operational Test (DDWS FOT) (Hanowski et al. 2008) conducted by VTTI was a naturalistic data collection study that collected data for 18 months from 103 commercial motor vehicle drivers (CMV). This database yielded approximately 2.2 million driving miles and, once reduced, found 1,217 valid safety critical events. The data was processed through software that flags potential events of interest based on trigger threshold values (Olsen et al. 2008). **Table 5** shows the triggers used for the DDWS FOT study.

Table 5. Trigger values used in the DDWS FOT (Hanowski et al. 2008).

Trigger Type	Trigger Values
Longitudinal acceleration (hard braking)	Deceleration greater than or equal to -0.35 *g*. Speed greater than or equal to 15 mi/h. Deceleration greater than or equal to -0.5 *g*. Speed less than or equal to 15 mi/h.
Time-to-collision	A forward time-to-collision (TTC) value of less than or equal to 1.8 s, coupled with a range of less than or equal to 150 ft, a target speed of greater than or equal to 5 mi/h, a yaw rate of less than or equal to $\lvert 4°/sec \rvert$, and an azimuth of less than or equal to $\lvert 0.8° \rvert$. A forward TTC value of less than or equal to 1.8 s, coupled with an acceleration or deceleration greater than or equal to $\lvert 0.35g \rvert$, a forward range of less than or equal to 150 ft, a yaw rate of less than or equal to $\lvert 4°/sec \rvert$, and an azimuth of less than or equal to $\lvert 0.8° \rvert$.
Swerve	Swerve value of greater than or equal to 3 rad/s². Speed greater than or equal to 15 mi/h.

Trigger Type	Trigger Values
Critical incident button	Activated by the driver upon pressing a button, located by the driver's visor, when an incident occurs that he/she deems critical.
Analyst identified	Event that was identified by a data analyst viewing video footage; no other trigger listed above identified the event (i.e., longitudinal acceleration, TTC, etc.).

When analyzing data, it is critical for the researcher to have a strong dataset of baseline data or epochs. Baseline epochs are described as brief time periods (e.g., 6 s) that are randomly selected from the recorded dataset. Baseline epochs are described using many of the same variables and data elements used to describe and classify crashes, near-crashes, and crash-relevant conflicts. Examples of such variables included ambient weather, roadway type, and driver behaviors. In a recent study utilizing the dataset from the 100-Car Naturalistic Driving Study (Dingus et al. 2006), one of the first large-scale studies, McLaughlin et al. (2009b) explain the process researchers used to utilize the 100-Car dataset in order to develop a baseline database that would quantify the frequency of different conditions and behaviors present while driving. During this process, researchers used the dataset to create a description of baseline driving. Conditions such as lighting and weather were then classified. Analysts used these random epochs to estimate miles traveled in different conditions by multiplying a participant's estimated miles traveled by the classified conditions. The sampling was stratified according to the involvement of each driver in crashes, near-crashes, and incidents of all types. From this analysis, it was determined that if a driver was involved in 3 percent of total events, then 3 percent of baseline epochs would be used from that driver's data.

NATURALISTIC TRUCK DATABASE AND TRIGGERS

The benefit of a naturalistic driving study is the ability to view the driver behavior and performance data that has been collected in a naturalistic environment. In the DDWS FOT study, a diverse set of measures were collected, including driver input/performance measures, video, actigrapy, and questionnaires. Of the 103 drivers, 24 were randomly assigned to the Control group (A^9) and 79 drivers were randomly assigned to the Experimental group (A^2B^9). *A* refers to the baseline condition, and *B* refers to the treatment condition. The superscripts refer to the number of weeks each participant drove an instrumented truck. When participants were driving in the baseline condition, the DDWS did not provide alerts. However, in the treatment condition the drivers were monitored and alerts were provided.

Three types of data were collected by the DAS: video, dynamic performance, and audio. Each driver drove 60 h in a 7-day period. Approximately 48,000 driving-data hours covering 2.2 million miles traveled were collected. Typically drivers would rotate into one of the 46 instrumented trucks, and each driver drove, on average, for 12 weeks. The DAS computer included five major components: the DDWS, sensors, vehicle network, incident box, and video cameras. Four digital video cameras were used to continuously record the driver and the driving environment. The four cameras were multiplexed into a single image. The four camera views were forward, driver's face, rear-facing-left, and rear-facing-right. A software program called Loki was developed to coordinate the data collection from the different DAS components and to integrate the data into a specific DAS output file linked to the video. More than 100 variables

were collected, such as PERCLOS (percent eye closure); output; and driving performance data such as lane position, speed, and longitudinal acceleration.

Similarly, VTTI performed another large-scale naturalistic data collection study investigating crash risk by identifying safety critical events. The objective of the Naturalistic Truck Driving Study (NTDS) (Olson et al. 2009), as well as Blanco et al. (in press), was to use the naturalistic driving data to investigate issues in the trucking industry such as driver sleep/rest cycles and crash countermeasures. Continuous driving performance data was collected for 100 commercial motor vehicle drivers during 4 months of their normal driving routine. Approximately 14,600 hours of driving data were collected. During this period of time, more than 2,800 safety critical events were identified: 13 crashes, 58 near-crashes, 1,595 crash-relevant conflicts, and 1,213 unintentional lane deviations. In addition to video and performance data, each participant was asked to wear an actigraph watch (i.e., a sleep monitor) and to fill out a daily activity log. These data provided information about the drivers' daily sleep patterns and amount of sleep, as well as other measure of interest such as time since last sleep period in reference to a critical incident. One or more crash countermeasures were also identified for each critical incident. The study recruited 100 participants from four participating trucking fleets. Each participant was observed for approximately 4 work weeks. Once the 4-week data collection period was over, another participant would start driving the instrumented truck.

Three forms of data were collected by the NTDS DAS: video, dynamic performance, and audio. As in the DDWS FOT, the DAS system for the NTDS consisted of five components: dynamic sensors, vehicle network, incident box, and video cameras. A lane tracker was included in the DAS system and consisted of a single analog black-and-white camera, a personal computer with a frame grabber card, and an interface-to-vehicle network for obtaining ground speed. Once installed, software automatically calibrated itself to determine the camera position. The following variables were reported:

- Distance from center of truck to left and right lane markings (estimated max error <6 inches, average error < 2 inches).
- Angular offset between truck centerline and road centerline (estimated max error <1°).
- Approximate road curvature.
- Confidence in reported values for each marking found.
- Marking characteristics, such as dashed versus solid and double versus single.
- Status information, such as in-lane or solid line crossed.

An additional camera view looking over the driver's shoulder into the lap was added to the NTDS after data analysts reported that in the DDWS FOT that the drivers would often reach for an object outside the camera view, so they were unable to determine what the driver was trying to reach. This additional view provided information on many potentially distracting driver behaviors. The data reduction process was similar to that of the DDWS FOT. The first step was to process data using modified trigger values (Table 6) to flag potential events of interest. The lane deviation trigger in the NTDS was not included in the DDWS FOT.

Table 6. Trigger values used in the NTDS (Olson et al. 2009).

Trigger Type	Trigger Values

Trigger Type	Trigger Values
Longitudinal acceleration (hard braking)	Deceleration greater than or equal to -0.20 *g*. Speed greater than or equal to 1 mi/h.
Time-to-collision	A forward time-to-collision (TTC) value of less than or equal to 2 s, coupled with a range of less than or equal to 250 ft, a target speed of greater than or equal to 5 mi/h, a gyro rate of less than or equal to $\left\vert 6°/sec \right\vert$, and an azimuth of less than or equal to $\left\vert 0.12° \right\vert$.
Swerve	Swerve value of greater than or equal to 2 rad/s². Speed greater than or equal to 5 mi/h.
Critical incident button	Activated by the driver upon pressing a button, located by the driver's visor, when an incident occurs that he/she deems critical.
Analyst identified	Event that was identified by a data analyst viewing video footage; no other trigger listed above identified the event (i.e., longitudinal acceleration, TTC, etc.).

Together, these CMV, heavy vehicle naturalistic studies yielded approximately 60,000 hours and 3 million miles of continuous data, which provide an extremely rich dataset. Further analysis on databases such as these will continue to answer questions on driver behavior.

The 100-Car Study

VTTI conducted another notable naturalistic data collection study (mentioned briefly above), the 100-Car Naturalistic Study (Dingus et al. 2006), which instrumented 100 light vehicles. The 100-Car study was unique in that it was the first study conducted in which vehicles were instrumented with the primary goal of collecting large-scale, naturalistic driving data. Also, 78 of the 100 instrumented vehicles were those of the participants, and the participants were given no special instructions. Approximately 2 million vehicle miles were included in the dataset which is rich with extreme cases of driving behavior and performance.

An in-depth analysis focusing on driver inattention conducted by Klauer et al. (2006), used data collected in the 100-Car Naturalistic Driving Study. The purpose of the analysis was to establish a relationship between driving behavior and crash or near-crash involvement. An additional database of baseline epochs was reduced from the raw 100-Car database to be used alongside the full 100-Car study event database. The event database consisted of crashes, near-crashes and incidents. The baseline database, created specifically for this study, consisted of a stratified sample of 20,000 6-second segments during which the vehicle maintained a velocity greater than 5 mi/h. Relative near-crash or crash risk was calculated by odds ratio using both crash and near-crash data compared to normal, baseline driving data for various sources of inattention. Additional analyses were performed of the corresponding attributable risk percentages to estimate the population's percentage of crashes and near-crashes occurring due to inattention. The analyses performed on this database allowed researchers to find the direct correlation between driver inattention and crash or near-crash incidents. More specifically, it was found that drivers engaging in secondary tasks that were visually or manually engaging had three times greater crash or near-crash risk than those who are attentive. Researchers also came to the conclusion

that brief glances from the forward roadway to scan the environment are safe, but glances that are 2 s or longer increase the crash or near-crash risk by twice that of normal baseline driving.

SAFETY CRITICAL EVENTS

The safety critical events for all three naturalistic data databases were identified using the triggers discussed earlier. Table 7 presents an enumeration of the types of crashes and near-crashes that were identified for each database. The safety critical events involving animals, pedestrians, or only a single vehicle are not relevant to this project because they are more representative of special case scenarios that are not recurrent. The crashes and near-crashes involving multiple vehicles is representative of conflicts between vehicles that have a recurring nature. The safety critical events that are not a forward or rear conflict will not have radar data, so the relative location and speed of the vehicles involved will not be known.

Table 7. Enumeration of crash and near-crash data.

	8-Truck Crashes	34-Truck Crashes	100-Car Crashes	8-Truck Near-crashes	34-Truck Near-crashes	100-Car Near-Crashes
Animal	1	4	2	1	9	10
Pedestrian					3	6
1 vehicle no objects	1		23	7	9	46
1 vehicle with objects	2	23	12	8	29	15
Multiple vehicles						
Not directly in front or behind	1		5	35	46	227
Directly in front or behind		2	27	9	22	457

Ten truck drivers and ten car drivers were selected for analysis and agent implementation based on the type of crashes and near-crashes they experienced. The drivers with the most crashes and near-crashes that were forward conflicts were selected. Figure 8 shows the demographic data of the chosen truck drivers, and Figure 9 shows their demographic data.

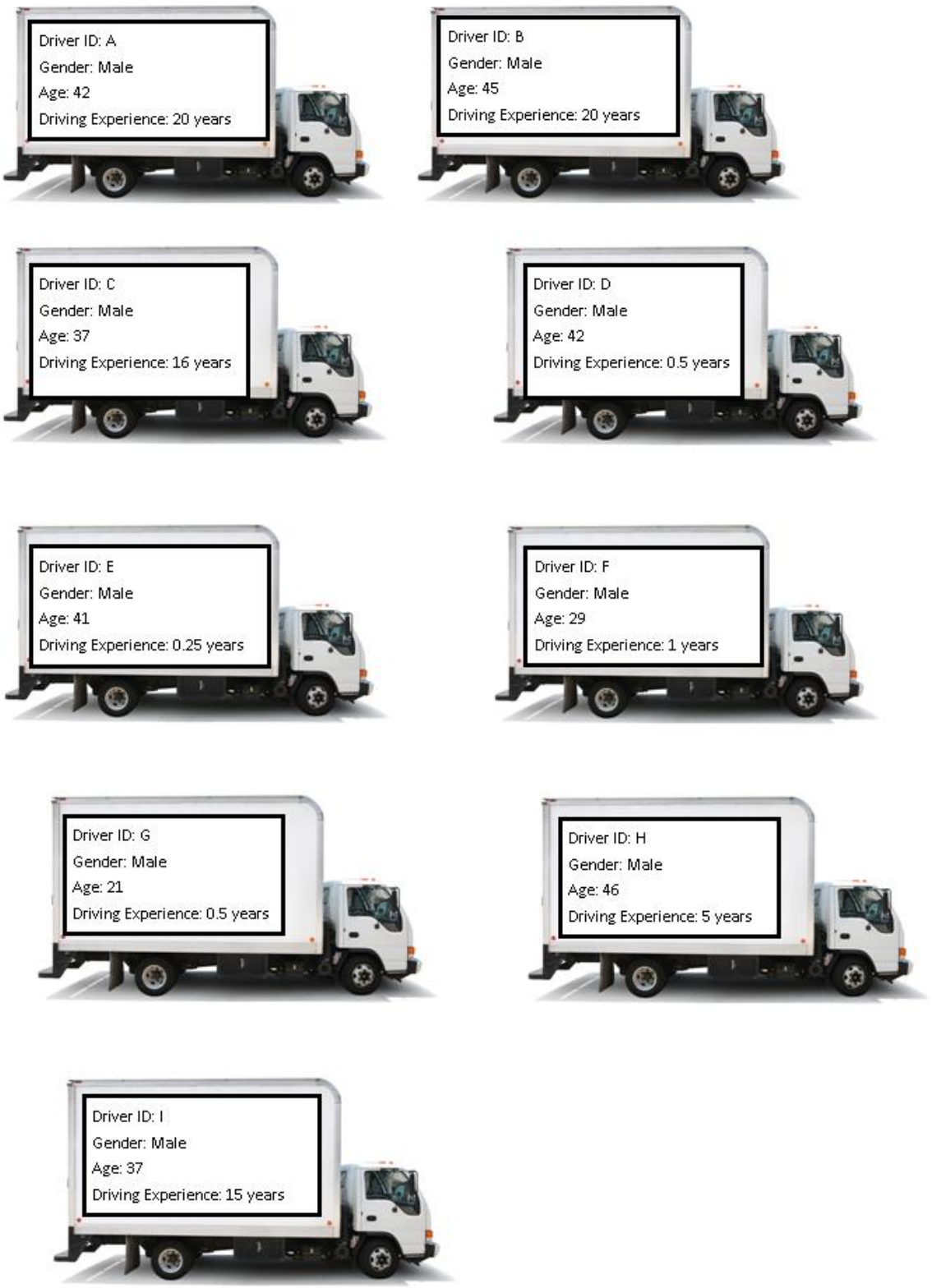

Figure 8. Truck driver demographic data.

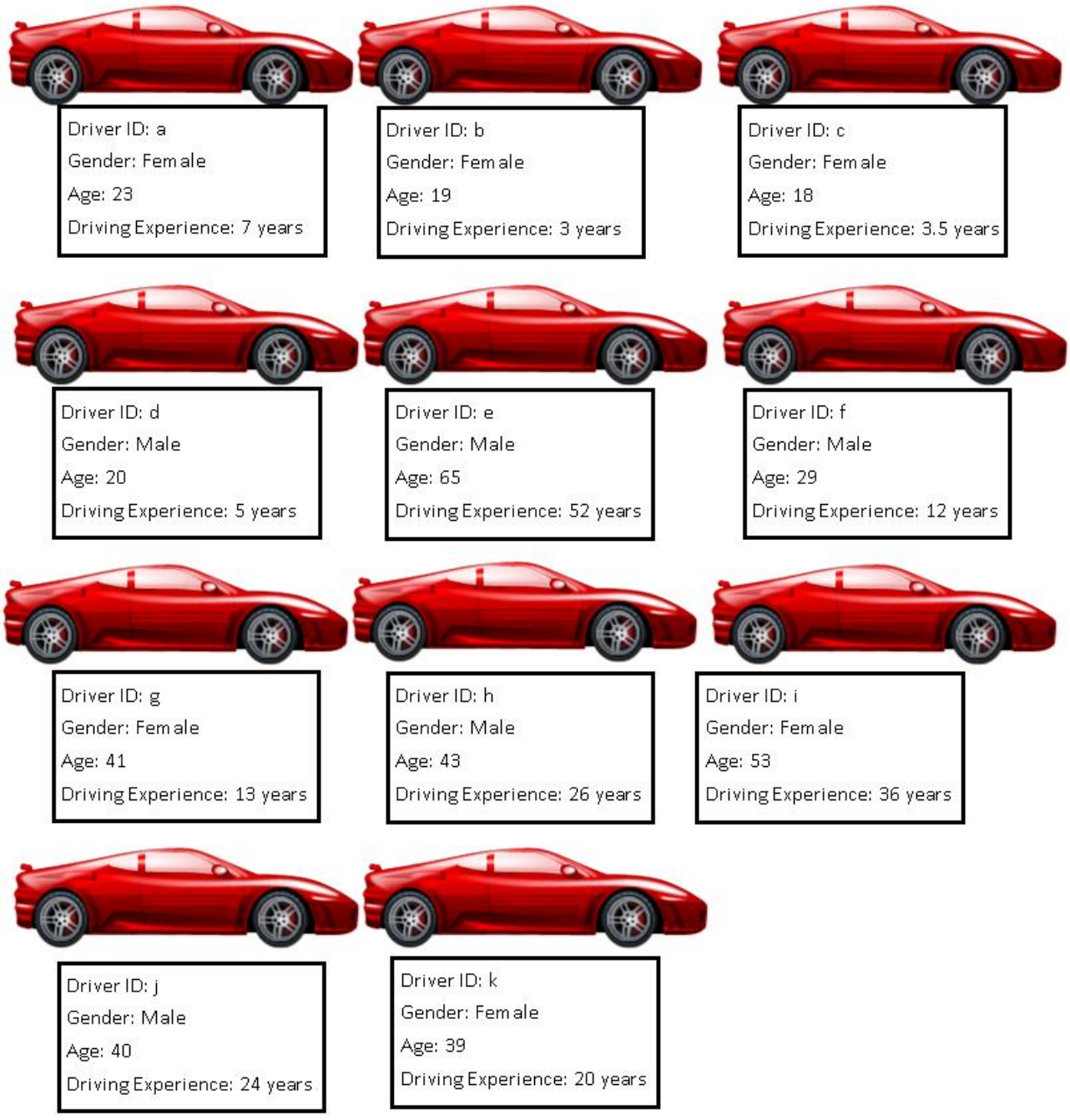

Figure 9. Car driver demographic data.

CAR-FOLLOWING FILTERING CRITERIA

Car-following situations were automatically extracted from the enormous volume of driving data in the database to analyze car-following behavior. This process made use of the tools SQL and MATLAB to query the databases and filter them down to the needed car-following periods.

SQL is a program that is useful for efficiently accessing data stored in a database. This program allows a user to write lines of code called a query, which is then used to find all the data in the database that matches the lines of code or filters in the query. This tool is a requirement for searching the huge volume of naturalistic data in the VTTI database. Even the data of a single driver is too large to analyze manually. For example, the data is collected at 10 hertz or 10 data points per second, so the 14,600 hours of data in the 8-Truck naturalistic study has 525.6 million data points.

An iterative process was used to filter for the car-following periods. Initial values and conditions were used, and after the events were flagged, they were reviewed in the video data to adjust the values accordingly to minimize the noise. Visual inspection of the first subsets created revealed some events that were not car-following, so additional filtering was performed to remove these events from the database.

Specifically, car-following periods were extracted automatically according to these conditions that arose as a result of the aforementioned process:

- Radar Target ID>0
  This eliminates the points in time without a radar target detected.
- Radar Range<=120 meters
  This represents 4 s of headway at 70 mi/h.
- -1.9 meters<Range*Sin (Azimuth) <1.9 meters
  This restricts the data to only one lane in front of the lead vehicle.
- Speed>=20km/h
  This speed was used to minimize the effect of traffic jams but still leave the influence of congestion in the data.
- Rho-inverse <=1/610 meters^{-1}
  This limits the curvature of the roadway such that vehicles are not misidentified as being in the same lane as the subject vehicle when roadway curvature is present.
- Length of car following period >= 30 seconds

All of the criteria except the length of car-following period criteria were applied in SQL. The length criteria posed difficulties in being applied in SQL. SQL evaluates each line or data point separately, but the criteria for length of car-following depend on the previous lines of data. MATLAB was used for this task because it offered simple and computationally efficient methods to enforce the length of car-following period criteria.

The results of these filtering criteria were verified through a video sample of 400 car-following periods, of which 396 were valid. Of the criteria listed above, it was found that the length of the car-following period criteria had the largest influence on the level of noise extracted. It was also found that increasing the length to more than 30 s reduced the number of car-following periods extracted to the point that the percentage of valid car-following periods in each sample would

have little improvement over the 30-s samples. The mentioned criteria result in numerous car-following periods for each driver, with the exact number found being vaguely related to the amount of data available for that driver. As a rough example, a driver with around 200 hours of data would yield in excess of 1,000 car-following periods, which is more than what is needed for a single driver.

CHAPTER CONCLUSION

The naturalistic data is unique in that it captures the behavior of drivers when they are not in a test environment. This allows analyses to give results that are more related to normal, natural driving behaviors. One of the strengths of the naturalistic dataset is its sheer amount of data, which allows for more in-depth analyses. This creates a problem, however, when sorting through the data to find the data relevant to the research objectives. For this project, SQL and MATLAB were used to find car-following periods in the data, which are representative of normal behaviors during which conflicts did not arise. In contrast, the times when conflicts did arise are called safety critical events, and these were identified in previous studies conducted by VTTI. Both datasets, safety critical events and car-following periods, are necessary for training an agent to mimic the range of behaviors of drivers.

Chapter 4. USING NATURALISTIC DATA TO CALIBRATE AND EXTEND EXISTING CAR-FOLLOWING MODELS—THE WIEDEMANN MODEL EXAMPLE

MODEL CALIBRATION

Wiedemann Car-Following Model

The Wiedemann car-following model was originally formulated in 1974 by Rainer Wiedemann (Wiedemann 1974). This model is known for its extensive use in VISSIM, the microscopic multimodal traffic flow simulation software (PTV-AG 2008). The simulation program chosen for the final implementation of trained agents as a result of this project was VISSIM. This makes the Wiedemann model a good illustrative example for this project. The Wiedemann model was constructed based on conceptual development and limited available data, and it must be calibrated to specific traffic stream data.

This project used the principal ideas behind the Wiedemann model, but the exact shape or formula used in the model were updated using the naturalistic data, which is deemed to be one of the best available sources of real-world data.

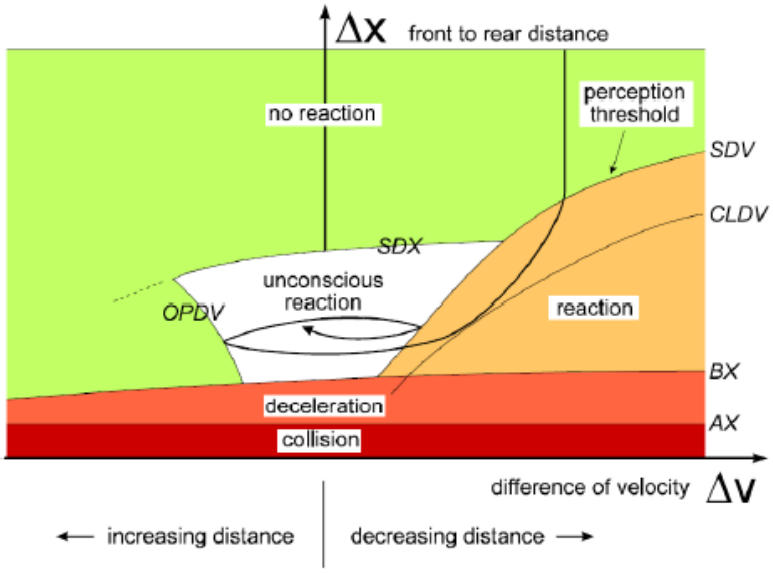

Figure 10. Diagram. Wiedemann 74 car-following logic (PTV-AG 2008).

Figure 10 shows the graphical form of the Wiedemann 74 model. The different thresholds are shown with a certain shape that can be amplified only during the calibration procedure. The figure shows the subject vehicle approaching a lead vehicle (ΔX decreasing due to the subject vehicle's greater speed, shown by a positive ΔV) and entering a perception area (crossing the SDV threshold) where it has to reduce speed. The subject vehicle then crosses another threshold (CLDV) where the driver reacts and reduces speed even further to enter an unconscious reaction car-following episode. The subject vehicle's driver then continues the unconscious car-following episode as long as it remains bounded by the OPDV, SDX, and SDV thresholds.

Reconstruction of the Wiedemann Model

The reconstruction process employed has two main sections. The first section is the evaluation and adaptation of the Wiedemann model according to the naturalistic data. The second section is the addition of new thresholds that aim to represent recurrent phenomena that were found in the naturalistic data.

Calibration and Adaptation of Existing Model

The Wiedemann model uses random numbers to create heterogeneous traffic stream behavior in VISSIM. These random numbers are meant to simulate the behavior of different drivers. The naturalistic data is a perfect match for this situation because the data is collected for individual drivers. Data for four different drivers was selected and processed to reconstruct the Wiedemann car-following model.

The reconstruction began by analyzing the equations for the different thresholds and different accelerations. This coincided with the design of a calibration framework tailored specifically to the naturalistic data. The analyzed and adjusted equations were implemented into the calibration framework, a Microsoft Excel spreadsheet. Microsoft Excel 2010 was chosen for this task due to the new evolutionary algorithm included in Excel's Solver Add-in. The framework expressed the logic of the Wiedemann model as a series of state transitions. The states are defined by the different thresholds, and each state has an equation or parameter for the acceleration. The optimization function was simply the minimization of the error between the values calculated in the Wiedemann model and the values obtained directly from the data.

The starting point for the Wiedemann model is the desired distance between stationary vehicles. The value calculated by the equation in Figure 11 is used in the calculations for the other thresholds. When analyzing this equation with an individual driver in mind, the calibration parameters can be combined with a driver-dependent random number to create a single calibration parameter as shown in Figure 12.

$$\boldsymbol{AX = L_{n-1} + AXadd + RND1_n * AXmult}$$

Figure 11. Equation. Desired distance between stationary vehicles.

L_{n-1} is the length of the lead vehicle.
$AXadd, AXmult$ are calibrated parameters.
$RND1_n$ is a normally distributed driver-dependent parameter.

$$\boldsymbol{AX = L_{n-1} + AXadd}$$

Figure 12. Equation. Desired distance between stationary vehicles new equation.

The desired minimum following distance threshold is calculated using the equations in Figure 13 and Figure 14. The calibration parameters and random number can be combined to produce the equation in Figure 15.

$$\boldsymbol{ABX = AX + BX}$$

Figure 13. Equation. Desired minimum following distance threshold.

$$\boldsymbol{BX = (BXadd + BXmult * RND1_n) * \sqrt{v}}$$

Figure 14. Equation. Desired minimum following distance threshold supporting equation.

$BXadd, BXmult$ are calibration parameters.
v is the minimum of the speed of the subject vehicle and the lead vehicle.

$$\boldsymbol{BX = BXmult * \sqrt{v}}$$

Figure 15. Equation. Desired minimum following distance threshold new supporting equation

The maximum following distance is calculated using the equations in Figure 16 and Figure 17. The second equation includes calibration parameters and random numbers, but it mainly serves as a multiplier to BX. Since it serves as only a multiplier, the first equation can be condensed into the equation shown in Figure 18.

$$\boldsymbol{SDX = AX + EX * BX}$$

Figure 16. Equation. Maximum following distance.

$$\boldsymbol{EX = EXadd + EXmult * (NRND - RND2_n)}$$

Figure 17. Equation. Maximum following distance supporting equation.

$EXadd, EXmult$ are calibration parameters.
$NRND$ is a normally distributed random number.
$RND2_n$ is a normally distributed driver-dependent parameter.

$$\boldsymbol{EX = EXmult}$$

Figure 18. Equation. Maximum following distance new supporting equation.

The perception threshold marks the point at which a driver will begin to react to the lead vehicle. This threshold is calculated by the equation in Figure 19**.** The equations in Figure 11 and Figure 20 are needed to calculate the equation in Figure 19.

$$\boldsymbol{SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX}\right)^2}$$

Figure 19. Equation. Perception threshold.

$$\boldsymbol{CX = CXconst * \left(CXadd + CXmult * (RND1_n + RND2_n)\right)}$$

Figure 20. Equation. Perception threshold supporting equation.

L_{n-1} is the length of the lead vehicle. $CXconst, CXadd, CXmult$ are calibrated parameters. $RND1_n$ and $RND2_n$ are normally distributed, driver-dependent parameters.

When looking at the equation in Figure 20 with a specific driver in mind, the equation can be reduced by combining all of the variables that would have a constant value. This collapses the equation to the one in Figure 21.

$$\boldsymbol{CX = CX}$$

Figure 21. Equation. Perception threshold new supporting equation.

This collapses the equations to have only one input variable and two calibrated parameters that are specific to a single driver.

The reaction curve marks the location of a second acceleration change point while the subject vehicle is still closing on the lead vehicle. In VISSIM this threshold is assumed to be equivalent to the perception threshold. Due to that similarity, the equation used for the reaction threshold, shown in Figure 22, is derived from the equation in Figure 19.

$$\boldsymbol{CLDV = \left(\frac{\Delta x - L_{n-1} - AX}{CLDVCX}\right)^2}$$

Figure 22. Equation. Reaction threshold.

$CLDVCX$ is a calibrated parameter specific to one driver.

The OPDV (opening difference in velocity) curve is primarily a boundary to the unconscious reaction region. It represents the point where the driver notices that the distance between his or her vehicle and the lead vehicle is increasing over time. When this realization is made, the driver will accelerate to maintain desired headway. This threshold is calculated using the equation in Figure 23.

$$\boldsymbol{OPDV = CLDV * (-OPDVadd - OPDVmult * NRND)}$$

Figure 23. Equation. OPDV threshold.

$OPDVadd$ and $OPDVmult$ are calibrated parameters.
$NRND$ is a normally distributed random parameter.

When considering only one driver, the equation changes to the one in Figure 24.

$$\boldsymbol{OPDV = CLDV * OPDVmult}$$

Figure 24. Equation. OPDV threshold for one driver.

The Wiedemann model reuses the perception threshold as a boundary to the unconscious reaction region. Again, this is the point at which the driver notices that the distance between his or her vehicle and the lead vehicle is decreasing over time. However, this second use of the threshold is used when the subject vehicle is already engaged in following the lead vehicle. In our reconstructed model, this reuse of the perception threshold was given its own equation to provide hysteresis control and to evaluate the adequacy of reusing the perception threshold. The equation in Figure 25 is of the same form as the one in Figure 19, but with a different calibrated parameter.

$$\boldsymbol{SDV2 = \left(\frac{\Delta x - L_{n-1} - AX}{CX2}\right)^2}$$

Figure 25. Equation. Boundary to the car-following regime.

$CX2$ is a calibrated parameter.

The first state is the free-driving regime, in which the subject vehicle is not reacting to a lead vehicle and is travelling at a desired speed or accelerating to a desired speed. The free-driving regime is defined as the area above the perception threshold and the maximum following distance threshold. If the subject vehicle enters the free-driving regime, the subject vehicle will then accelerate until the desired speed is reached. The value for this acceleration is calculated using the equations in Figure 26 and Figure 27. The first equation relates the maximum speed to the current speed times from the second equation, and it calculates an acceleration value accordingly to reach the maximum speed. The equation in Figure 28 is derived when the acceleration ends at

the desired speed and not the maximum speed. Also, the equation in Figure 29 is derived from the fact that equation in Figure 27 will reduce to a constant value.

$$\boldsymbol{b_{max} = BMAXmult * (v_{max} - v * FaktorV)}$$

Figure 26. Equation. Acceleration in free driving regime.

$BMAXmult$is a calibration parameter.
v_{max} is the maximum speed of the vehicle.

$$\boldsymbol{FaktorV = \frac{v_{max}}{v_{des} + FAKTORVmult * (v_{max} - v_{des})}}$$

Figure 27. Equation. Acceleration in free driving regime supporting equation.

$FAKTORVmult$ is a calibration parameter.
v_{des} is the desired speed.

$$\boldsymbol{b_{max} = BMAXmult * (v_{des} - v * FaktorV)}$$

Figure 28. Equation. New acceleration in free driving regime.

$$\boldsymbol{FaktorV = FAKTORVmult}$$

Figure 29. Equation. Acceleration in free driving regime new supporting equation.

The approaching regime occurs when a vehicle in the free driving regime passes the perception threshold. This vehicle will then decelerate according to the equation in Figure 30.

$$\boldsymbol{b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1}}$$

Figure 30. Equation. Approaching regime deceleration.

The closely approaching regime occurs only when a vehicle in the approaching regime passes the closing difference in velocity threshold. In VISSIM this regime is ignored, so the deceleration is still calculated by the equation in Figure 30.

The deceleration following regime occurs as a result of a vehicle in the approaching or closely approaching regime passing the perception threshold, or a vehicle in the acceleration following regime passing the second perception threshold. When a vehicle enters the deceleration following regime, the acceleration is calculated by the negative of the equation in Figure 31. When considering one driver, this equation can be reduced to the one in Figure 32.

$$\boldsymbol{b_{null} = BNULLmult * (RND4_n + NRND)}$$

Figure 31. Equation. Acceleration in the following regime.

$BNULLmult$ is a calibration parameter.
$RND4_n$ is a normally distributed driver-dependent parameter.

$$\boldsymbol{b_{null} = b_{null}}$$

Figure 32. Equation.Acceleration in the following regime, reduced.

The acceleration following regime occurs when a vehicle in the deceleration following regime passes the opening difference in velocity threshold, or when a vehicle in the emergency regime

passes the minimum following distance threshold. The acceleration for a vehicle in the acceleration following regime is simply the positive value of the equation in Figure 31. If a vehicle in this regime accelerates and crosses the maximum following distance threshold, then that vehicle will enter the free-driving regime. Also, the inverse is true: a vehicle will enter the acceleration following regime from the free-driving regime if the maximum following distance threshold is passed.

The emergency regime occurs any time that the space headway is below the minimum following distance threshold. The equations in Figure 33 and Figure 34 calculate the acceleration in the emergency regime. The second equation can be reduced to the one in Figure 35 when individual drivers are considered.

$$\boldsymbol{b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1} + b_{min} * \frac{ABX - (\Delta x - L_{n-1})}{BX}}$$

Figure 33. Equation. Acceleration in the emergency regime.

$$\boldsymbol{b_{min} = -BMINadd - BMINmult * RND3_n + BMINmult * v_n}$$

Figure 34. Equation. Acceleration in the emergency regime supporting equation.

$BMINadd$ and $BMINmult$ are calibration parameters.
$RND3_n$ is a normally distributed driver-dependent parameter.
v_n is the speed of the subject vehicle.

$$\boldsymbol{b_{min} = BMINadd + BMINmult * v_n}$$

Figure 35. Equation. Acceleration in the emergency regime reduced supporting equation

New Thresholds

Hook Following Threshold

The first and most prominent recurrent phenomenon in the naturalistic data is car-following periods that begin at low space headways, with the lead vehicle travelling at a higher speed than the subject vehicle. The low space headway and higher speed indicate the completion of a pass maneuver for the lead vehicle as it merges back into the same lane as the subject vehicle. The interesting phenomenon is that the completion of the pass maneuver initiates car-following behavior. Table 8 shows that, from a sample of the car-following periods, this phenomenon represents 45 percent of the data, which is too large to ignore.

Table 8. Random sample results for number of hook following periods.

	Driver J	Driver K	Driver F	Driver L	% of Total
Regular car-following	217	131	102	256	55%
Hook car-following	166	123	142	149	45%

Figure 36 shows an example of hook car-following behavior. It is deemed hook car-following because it is initiated by the subject vehicle "hooking" onto the faster lead vehicle. The figure

also shows that the vehicle moves into the unconscious reaction region and oscillates just like the regular Wiedemann model in Figure 10.

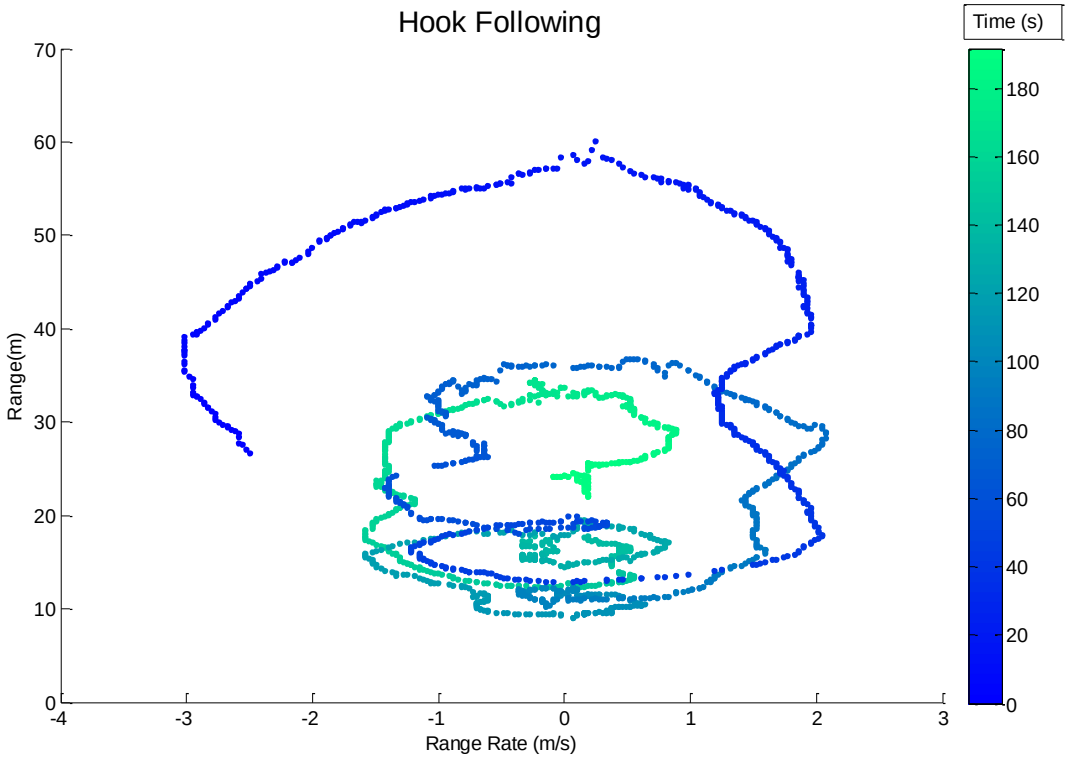

Figure 36. Graph. Example hook car-following period.

The process used was to plot two different sets of data and find the dividing line between them. The two datasets are (1) for vehicles that were not hooked onto for following and (2) for vehicles that were hooked onto and followed. The logic employed to differentiate between not hooking and hooking onto the lead vehicle operated on the fact that if a car-following period is initiated, then the subject vehicle will travel faster than the lead vehicle to catch up and follow it.

Passing Threshold

The Wiedemann model uses an external lane-changing logic for passing decisions. Trends in the naturalistic data show that this passing decision could be represented by an additional threshold to the Wiedemann model. This new threshold would operate much like the perception threshold, but the primary reaction would be to find a gap in an adjacent lane in order to pass. It will also operate like the perception threshold because the driver will need to react to the lead vehicle until an adequate gap is available.

The method employed here is the same as that employed for the hook threshold, except that in this case the subject vehicle will decelerate in order to follow.

RESULTS

Evaluation of the Existing Wiedemann Model

These new threshold equations were evaluated and calibrated using a genetic algorithm framework. The framework operated from state transitions, that is, where the driver would switch states when crossing a threshold. The progression of the states was as follows: free driving, approaching, closely approaching, following decelerate, following accelerate. The emergency regime and free driving were not evaluated in order to focus on car-following behavior. The evaluation consisted of 100 car-following periods for four different drivers. The results of the genetic algorithm were compared to the VISSIM default values obtained from the work of Olstam and Tapani (2004) shown in Table 9. The length of the lead vehicle was evaluated as a calibration parameter to provide verification that the calibration is feasible. The results of the optimization function are shown in Table 10.

Table 9. Calibration parameters by driver.

	Default	Driver J	Driver K	Driver F	Driver L
Ln-1	4.500	4.494	5.553	4.547	5.680
AXadd	2.500	2.525	5.044	2.873	5.601
BXmult	3.000	3.041	3.405	3.421	3.435
EXmult	2.500	2.514	3.096	3.427	2.676
CX	40.000	40.066	75.042	62.405	93.131
CX2	40.000	39.031	28.978	73.258	73.026
CLDVCX	30.000	31.762	56.798	33.159	51.080
OPDVmult	-2.250	-2.512	-6.580	-1.068	-3.307
bnull	0.100	0.122	0.075	0.241	0.072
bmaxmult	0.088	0.099	0.153	0.427	0.387
FaktorVmult	0.025	0.039	0.167	0.200	0.182
bminadd	-20.000	-20.898	-13.167	-23.408	-41.721
bminmult	0.025	0.033	0.091	0.320	0.378
Vdes	40.000	42.674	84.038	88.721	16.067
FaktorV	1.000	0.940	0.522	0.507	1.959

Table 10. Root mean square error for optimization function.

	Default	Calibrated	% change
Driver J	0.784762	0.7533074	4%
Driver K	4.016006	0.9688645	76%
Driver F	3.263291	1.2555831	62%
Driver L	17.55786	1.0745801	94%

The results show that Driver J's behaviors are very similar to the VISSIM default values. The other drivers all show drastic improvement when calibrated. Also, a number of the parameters change drastically in value between drivers, which suggests that drivers should be calibrated

individually to preserve accuracy. The large variation also suggests that a generic model would be inaccurate.

New Thresholds

Hook Following Threshold

Figure 37 shows both the hooked and not-hooked reactions. As can be seen in the figure, there appears to be a dividing line between the two datasets. This same process is repeated for the other drivers, and the equations are summarized by driver in Table 11.

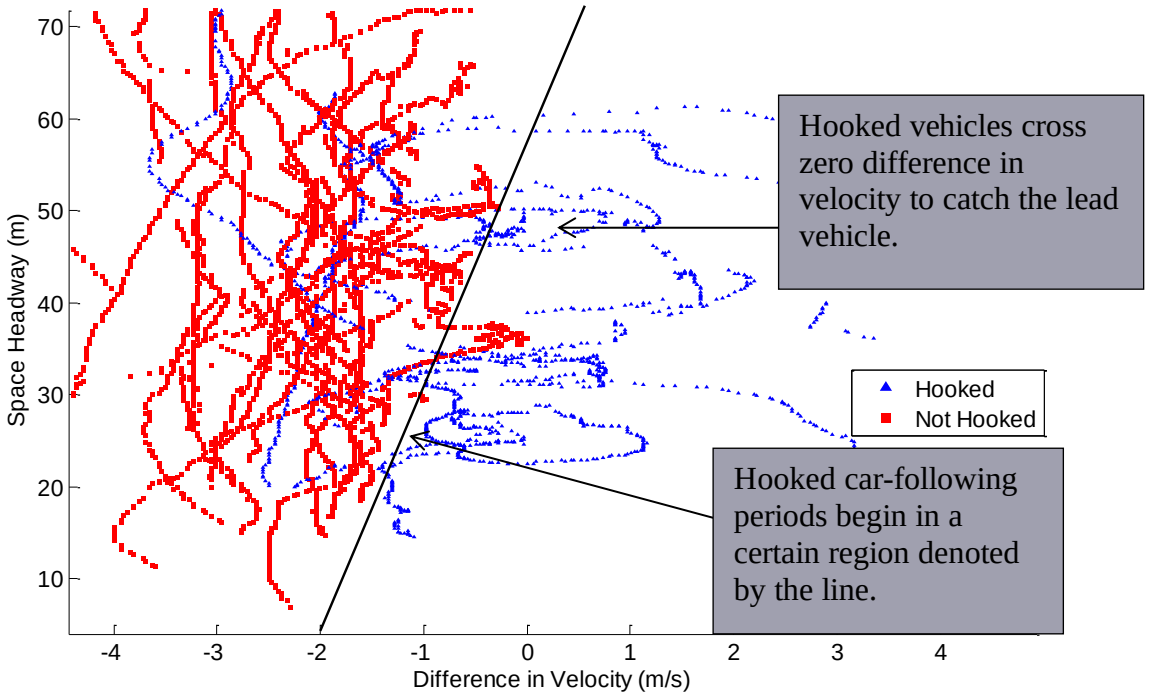

Figure 37. Graph. Hooked/not-hooked division line.

Table 11. Hook following threshold equations by driver.

Driver	Threshold Equation
J	11.667*ΔV+49.667
K	13.822*ΔV+43.239
F	13.333*ΔV+46.5
L	15.789*ΔV+54.772

Passing Threshold

The passing threshold is harder to discern than the hook threshold. Figure 38 shows data for Driver J, which is unclear until forced following is considered. In forced following, the driver intends to pass the lead vehicle, but there are no sufficient gaps allowing this maneuver. When forced following is considered, the division between passing and not passing is clearly above any hard deceleration at low space headways.

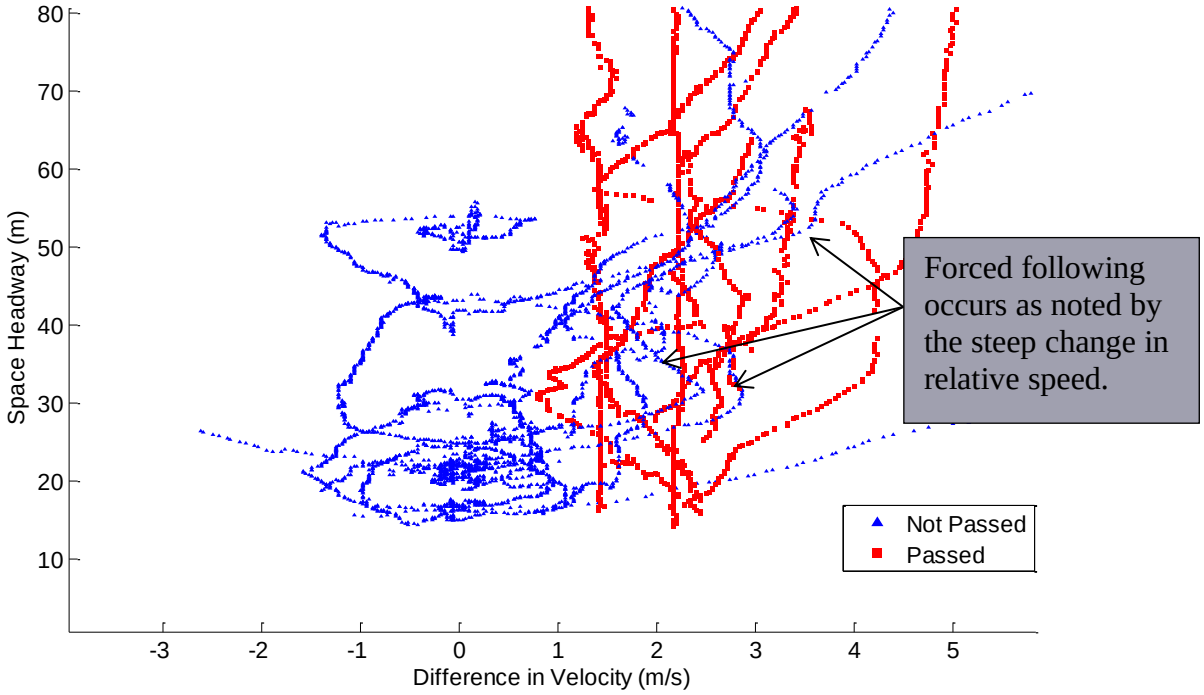

Figure 38. Graph. Pass decision curve.

Table 12. Pass threshold equations by driver.

	Pass Threshold Equation
Driver J	0.0017Δx^2-0.0421Δx+0.9981
Driver K	0.0007Δx^2+0.0113Δx+1
Driver F	0.0017Δx^2-0.0672Δx+1
Driver L	0.0002Δx^2-0.0007Δx+1.5

Reconstructed Wiedemann Model

The calibrated parameters and new thresholds were calculated for each driver and are presented in.Figure 39, Figure 40, Figure 41, and Figure 42. The SDV (perception threshold) and SDV2 (second perception threshold) are very similar for three of the drivers, but very different for Driver K. For Driver K, the CLDV (closing difference in velocity) threshold is greater than the SDV2, which means that Driver K prefers to decelerate at a harder rate when approaching a vehicle within the following regime. All of the drivers have SDV2s that minimize the effect of the acceleration following regime. The ABX (minimum following distance) and SDX (maximum following distance) thresholds show similarities between Drivers K and F and between Drivers J and L.

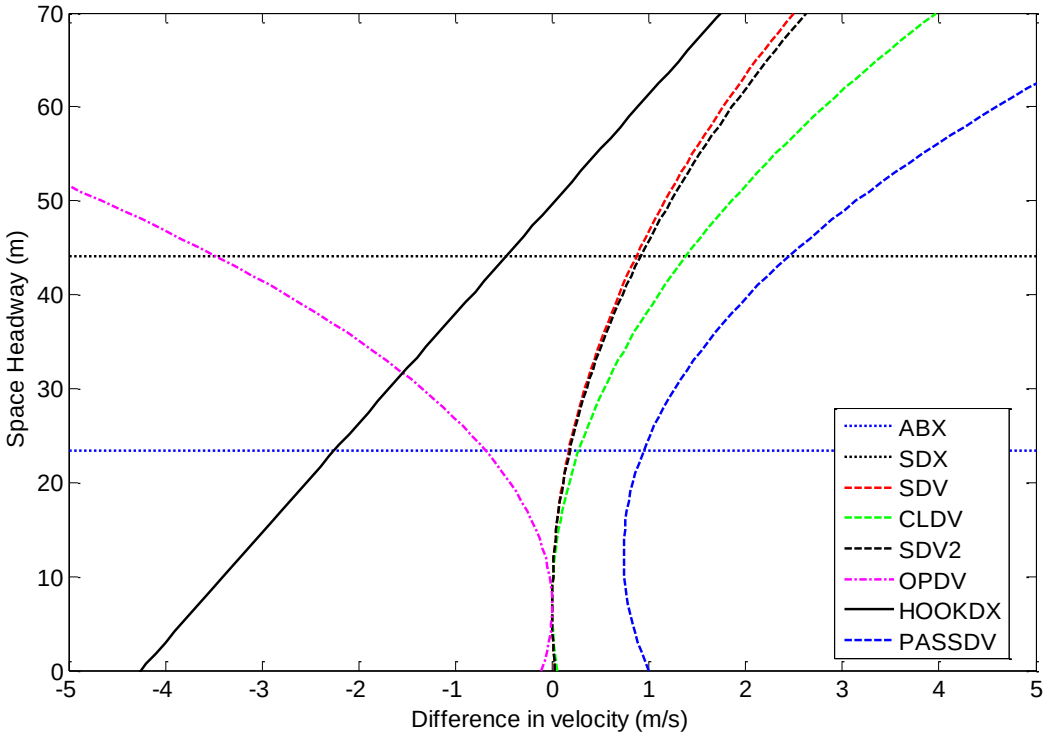

Figure 39. Graph. Thresholds for Driver J.

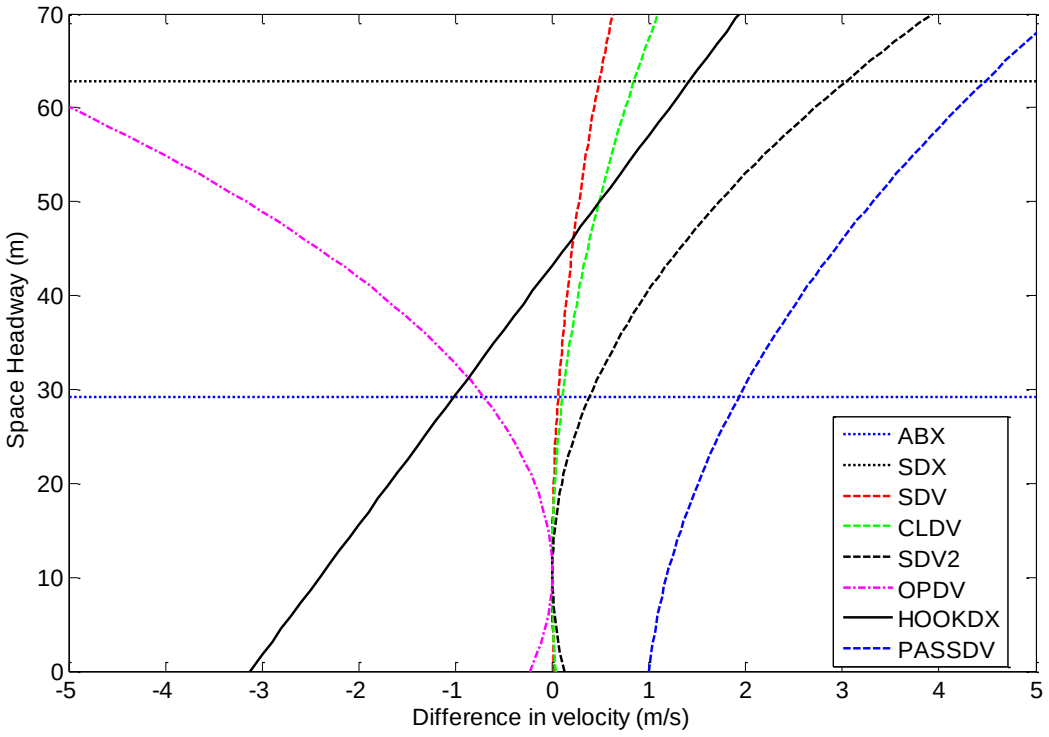

Figure 40. Graph. Thresholds for Driver K.

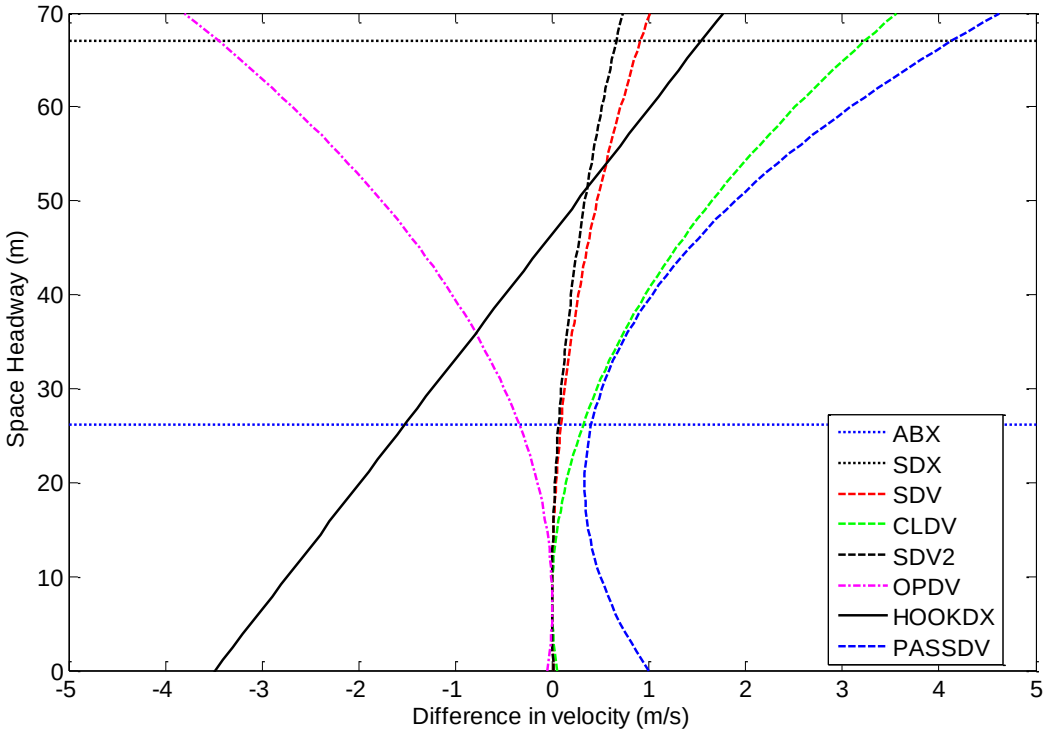

Figure 41. Graph. Thresholds for Driver F.

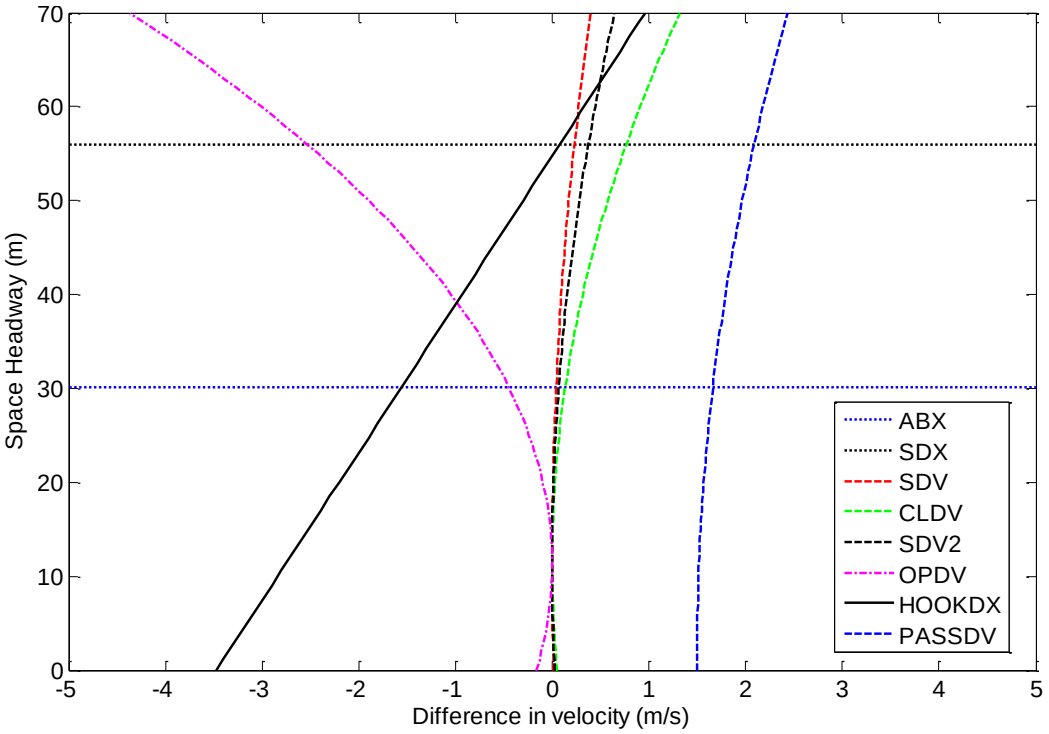

Figure 42. Graph. Thresholds for Driver L.

Conclusions

Modeling and calibrating drivers individually in this project added a new perspective to the Wiedemann model and allowed the authors to reconstruct the model by adding new thresholds. The inclusion of a hook following threshold adds value to the model by providing the ability to include a significant natural driving behavior. The addition of the passing threshold provides the model with the ability to easily transition from car-following to lane-changing behavior. This threshold also provides a way to force car-following behavior when a lane change is not possible. The addition of these new thresholds and the driver-specific equations gives the Wiedemann model a better ability to represent real-world driving behavior, which is the goal of all car-following models.

HYBRID MODEL DEVELOPMENT

GHR Model

The GHR model is shown in the equation in Figure 43. The GHR model relates the acceleration to the current speed, relative speed, and space headway.

$$\boldsymbol{a_n(t) = cv_n^m(t)\frac{\Delta v(t-T)}{\Delta x^l(t-T)}}$$

Figure 43. Equation. GHR model.

$a_n(t)$ is the acceleration of the subject vehicle at time t.
$v_n(t)$ is the speed of the subject vehicle at time t.
T is the perception reaction time of the driver.
$\Delta v(t-T)$ is the relative speed at time t minus T.
$\Delta x(t-T)$ is the space headway at time t minus T.
$c, l,$ and m are calibration parameters.

Wiedemann Model Equations

The starting point for the Wiedemann model is the desired distance between stationary vehicles. The value calculated by the equation in Figure 44 is used in the calculations for the other thresholds.

$$\boldsymbol{AX = L_{n-1} + AXadd + RND1_n * AXmult}$$

Figure 44. Equation. Desired distance between stationary vehicles.

L_{n-1} is the length of the lead vehicle.
$AXadd$ and $AXmult$ are calibrated parameters.
$RND1_n$ is a normally distributed driver-dependent parameter.

The desired minimum following distance threshold is calculated using the equations in Figure 45 and Figure 46.

$$\boldsymbol{ABX = AX + BX)}$$

Figure 45. Equation. Desired minimum following distance threshold.

$$\boldsymbol{BX = (BXadd + BXmult * RND1_n) * \sqrt{v}}$$

Figure 46. Equation. Desired minimum following distance threshold.

$BXadd$ and $BXmult$ are calibration parameters.
v is the minimum of the speed of the subject vehicle and the lead vehicle.

The maximum following distance is calculated using the equations in Figure 47 and Figure 48. The second equation includes calibration parameters and random numbers, but it mainly serves as a multiplier to BX.

$$\boldsymbol{SDX = AX + EX * BX}$$

Figure 47. Equation. Maximum following distance.

$$\boldsymbol{EX = EXadd + EXmult * (NRND - RND2_n)}$$

Figure 48. Equation. Maximum following distance supporting equation.

$EXadd$ and $EXmult$ are calibration parameters.
$NRND$ is a normally distributed random number.
$RND2_n$ is a normally distributed driver-dependent parameter.

The perception threshold marks the point at which a driver will begin to react to the lead vehicle. This threshold is calculated by the use of the equation in Figure 49. The equations in Figure 44 and Figure 50 are needed to calculate the equation in Figure 49.

$$\boldsymbol{SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX}\right)^2}$$

Figure 49. Equation. Perception threshold.

$$\boldsymbol{CX = CXconst * \left(CXadd + CXmult * (RND1_n + RND2_n)\right)}$$

Figure 50. Equation. Perception threshold supporting equation.

L_{n-1} is the length of the lead vehicle.
$CXconst, CXadd,$ and $CXmult$ are calibrated parameters.
$RND1_n$ and $RND2_n$ are normally distributed driver-dependent parameters.

The reaction curve marks the location of a second acceleration change point while the following vehicle is still closing on the lead vehicle. In VISSIM this threshold is assumed to be equivalent to the perception threshold. Due to that similarity, the equation used for the reaction threshold, shown in Figure 51, is derived from the equation in Figure 49.

$$\boldsymbol{CLDV = \left(\frac{\Delta x - L_{n-1} - AX}{CLDVCX}\right)^2}$$

Figure 51. Equation. Reaction threshold.

$CLDVCX$ is a calibrated parameter specific to one driver.

The OPDV (opening difference in velocity) curve is primarily a boundary to the unconscious reaction region. It represents the point at which the driver notices that the distance between his or her vehicle and the lead vehicle is increasing over time. When this realization is made, the driver

will accelerate to maintain the desired space headway. This threshold is calculated using the equation in Figure 52.

$$\boldsymbol{OPDV = CLDV * (-OPDVadd - OPDVmult * NRND)}$$

Figure 52. Equation. OPDV threshold.

$OPDVadd$ and $OPDVmult$ are calibrated parameters.
$NRND$ is a normally distributed random parameter.

The Wiedemann model reuses the perception threshold as a boundary to the unconscious reaction region. Again, this is the point at which the driver notices that the distance between his or her vehicle and the lead vehicle is decreasing over time. However, this second use of the threshold is used when the subject vehicle is already following the lead vehicle. In our model, this reuse of the perception threshold was given its own equation to provide hysteresis control and to evaluate the adequacy of reusing the perception threshold. The equation in Figure 53 is of the same form as the equation in Figure 49, but with a different calibrated parameter.

$$\boldsymbol{SDV2 = \left(\frac{\Delta x - L_{n-1} - AX}{CX2}\right)^2}$$

Figure 53. Equation. Perception threshold reuse.

$CX2$ is a calibrated parameter.

The first state is the free-driving regime, in which the subject vehicle is not reacting to a lead vehicle and is travelling at a desired speed or accelerating to a desired speed. The free-driving regime is defined as the area above the perception threshold and the maximum following distance threshold. If the subject vehicle enters the free-driving regime, the subject vehicle will then accelerate until the desired speed is reached. The value for this acceleration is calculated using the equations in Figure 54 and Figure 55. The first equation relates the maximum speed to the current speed times from the second equation, and it calculates an acceleration value accordingly to reach the maximum speed. This regime was not included in the calibration because it would only complicate the evaluation of car following.

$$\boldsymbol{b_{max} = BMAXmult * (v_{max} - v * FaktorV)}$$

Figure 54. Equation. Acceleration in the free-driving regime.

$BMAXmult$is a calibration parameter.
v_{max} is the maximum speed of the vehicle.

$$\boldsymbol{FaktorV = \frac{v_{max}}{v_{des} + FAKTORVmult * (v_{max} - v_{des})}}$$

Figure 55. Equation. Acceleration in the free-driving regime.

$FAKTORVmult$ is a calibration parameter.
v_{des} is the desired speed.

The approaching regime occurs when a vehicle in the free-driving regime passes the perception threshold. This vehicle will then decelerate according to the equation in Figure 56. The GHR model in Figure 43 replaces the equation in Figure 56. The GHR model, in this case, will have its own set of calibrated parameters specifically for approaching a lead vehicle.

$$b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1}$$

Figure 56. Equation. Deceleration in the approaching regime.

The closely approaching regime occurs only when a vehicle in the approaching regime passes the closing difference in velocity threshold. In VISSIM this regime is ignored, so the deceleration is still calculated by Figure 56. The closely approaching regime would typically result in a harder deceleration than the approaching regime due to the low space headway. The GHR model accounts for this difference by using a specialized set of calibration parameters.

The deceleration following regime occurs as a result of a vehicle in the approaching or closely approaching regime passing the perception threshold, or a vehicle in the acceleration following regime passing the second perception threshold. When a vehicle enters the deceleration following regime, the acceleration is calculated by the negative of Figure 57.

$$b_{null} = BNULLmult * (RND4_n + NRND)$$

Figure 57. Equation. Acceleration in following regime.

$BNULLmult$ is a calibration parameter.
$RND4_n$ is a normally distributed driver-dependent parameter.

The acceleration following regime occurs when a vehicle in the deceleration following regime passes the opening difference in velocity threshold, or when a vehicle in the emergency regime passes the minimum following distance threshold. The acceleration for a vehicle in the acceleration following regime is simply the positive value of the equation in Figure 57. If a vehicle in this regime accelerates and crosses the maximum following distance threshold, then that vehicle will enter the free-driving regime. Also, the inverse is true: a vehicle will enter the acceleration following regime from the free-driving regime if the maximum following distance threshold is passed.

The equation in Figure 57 lacks any form of reaction to the lead vehicle. The only reaction is to switch from acceleration to deceleration once a threshold is crossed. This relies solely on the assumption that a constant small acceleration or deceleration will be sufficient to account for actions of the lead vehicle while in the following regime. The GHR model can be used to provide clarity in the two following regimes. The two regimes were calibrated with a different set of parameters to account and test for any difference between the accelerations and decelerations of the two following regimes.

The emergency regime occurs any time that the space headway is below the minimum following distance threshold. The equations in Figure 58 and Figure 59 calculate the acceleration in the emergency regime. This regime is not included in the calibration because it resembles crash and near-crash reactions more than car following.

$$b_n = \frac{1}{2}\frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + b_{n-1} + b_{min} * \frac{ABX - (\Delta x - L_{n-1})}{BX}$$

Figure 58. Equation. Acceleration in the emergency regime.

$$b_{min} = -BMINadd - BMINmult * RND3_n + BMINmult * v_n$$

Figure 59. Equation. Acceleration in the emergency regime supporting equation.

$BMINadd$ and $BMINmult$ are calibration parameters.
$RND3_n$ is a normally distributed driver-dependent parameter.
v_n is the speed of the subject vehicle.

Calibration Framework

The car-following data for four different drivers from the naturalistic data was used to calibrate the models by using a genetic algorithm. Ten car-following periods per driver were used in the calibration. The framework consisted of calibrating the parameters for the Wiedemann model based on the equations shown earlier. This framework was then altered such that the GHR model replaced the acceleration equations in the following regimes: approaching, closely approaching, acceleration following, and deceleration following. The GHR model was given a different set of calibration parameters (c, l, m, and T) for each of those regimes. The reaction time, T, was used as a calibration parameter to obtain a measure of the attentiveness of the different drivers.

Results

Figure 60 shows how the Wiedemann model in its current state is very reactive to the changes in the speed of the lead vehicle. This reactivity can be attributed to the fact that the equation in Figure 56 is a relative acceleration equation. Also, the recovery from a hard deceleration does not match the data in Figure 60.

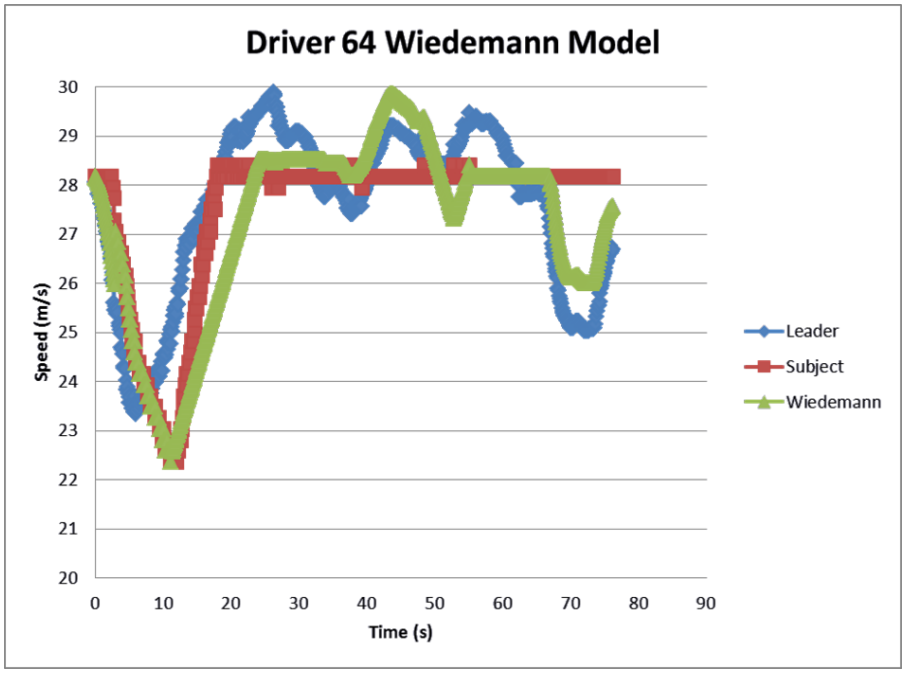

Figure 60. Graph. Driver F Wiedemann car-following period.

Figure 61 presents the same car-following episode as Figure 60, but with the GHR model integrated into the Wiedemann model. Figure 61 shows that the GHR model has removed the inaccurate reactivity to changes in the speed of the lead vehicle.

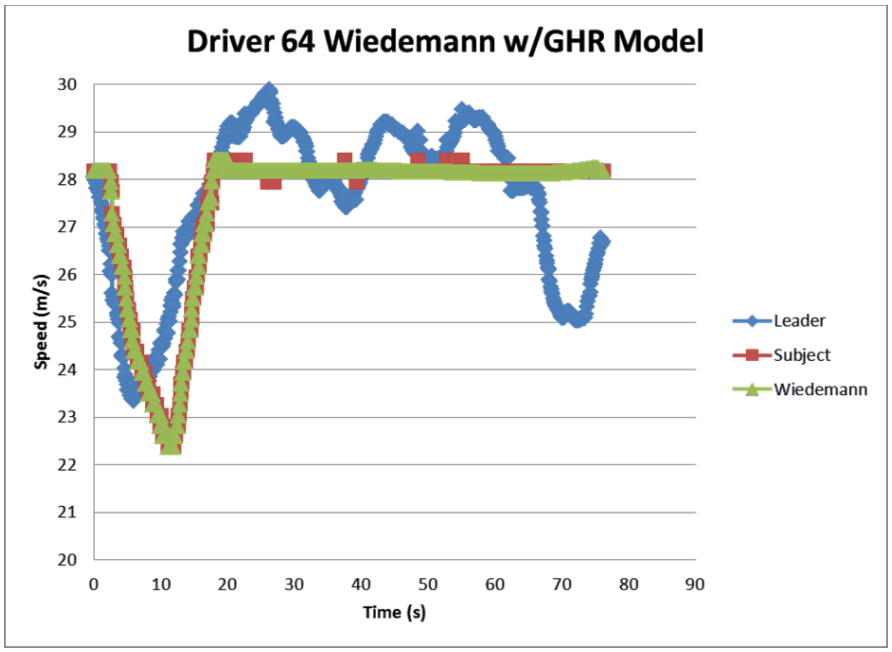

Figure 61. Graph. Driver F Wiedemann with GHR car-following period.

Figure 62 shows a comparison of the Wiedemann model and the Wiedemann model integrated with the GHR model. The GHR model corrects the recovery from a hard deceleration error seen in the Wiedemann model. The Wiedemann model appears to accelerate at a lower rate and for a longer period of time than the is found in the naturalistic data.

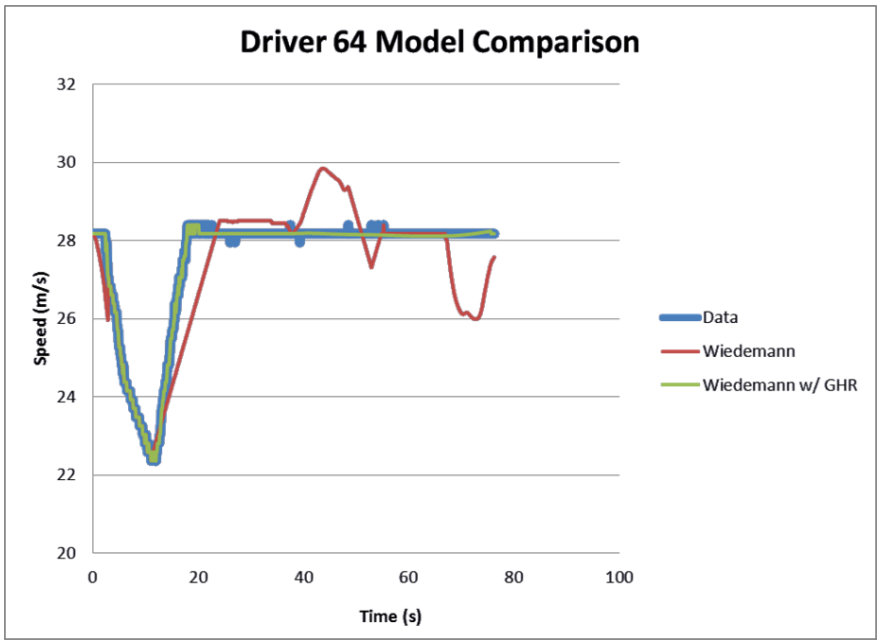

Figure 62. Graph. Comparison of Wiedemann and Wiedemann with GHR.

Figure 63 shows how the models compare in a graph showing range versus range rate. The Wiedemann model shows improper oscillation toward the end of the car-following period. Also,

the Wiedemann model continues producing small oscillations at a high range when the data shows a larger oscillation moving to a smaller range at the end of the car-following period.

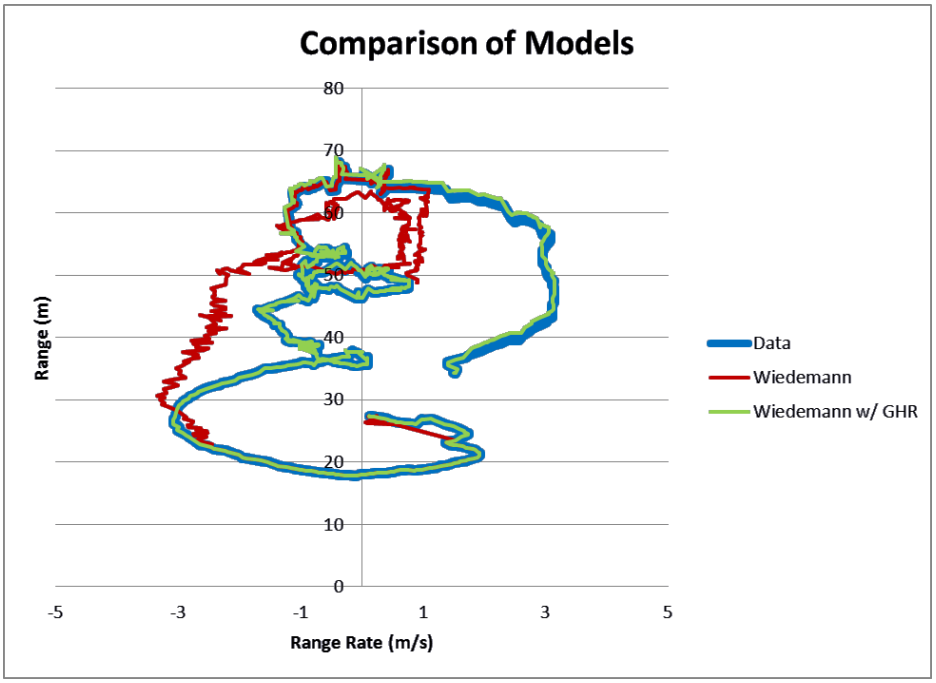

Figure 63. Graph. Comparison between models in range versus range rate.

Table 13 presents the results for the calibrated parameters of the GHR model by driver and regime. The values show variance between the drivers. The calibrated perception reaction times are low in some cases, which indicates that, in these cases, the thresholds serve as decision points and that actions from that point on are based on that decision.

Low reaction time values in the following regimes (OPDV and SDV2) indicate more active and focused following behavior. High reaction times in the same region indicate more relaxed and inattentive following behavior. This is indicated by the fact that the stimulus for the reaction is either a change in relative speed or a change in the space headway.

Table 13. GHR calibrated parameters by driver and regime.

	Driver J	Driver K	Driver F	Driver L
c SDV	0.064	0.935	0.446	0.434
m SDV	0.704	0.843	1.158	0.954
l SDV	2.135	1.997	2.046	1.933
T SDV	0.511	0.170	1.042	0.143
c SDV2	0.688	1.522	0.617	0.496
m SDV2	0.655	-0.592	0.734	0.337
l SDV2	2.613	2.565	1.821	2.002
T SDV2	0.227	0.511	0.498	0.505
c CLDV	0.130	0.532	0.602	0.706
m CLDV	0.880	-0.007	0.900	0.708
l CLDV	2.854	2.033	2.960	1.685

T CLDV	0.418	0.235	0.602	0.148
c OPDV	1.381	1.231	1.218	1.126
m OPDV	-0.521	-0.375	-0.064	-0.193
l OPDV	1.823	2.617	3.200	2.001
T OPDV	0.191	0.247	0.835	0.472

Table 14 presents the root mean square error values for both the Wiedemann model and the Wiedemann model integrated with the GHR model. The results show large improvement for two of the drivers but little improvement for the other two drivers. This indicates that the Wiedemann model calibrates to two of the drivers better than to the other two drivers. This means that the equations of the Wiedemann model reflect the behaviors of two of the drivers, but they do not accurately reflect the behaviors of the other two drivers.

Table 14. Root mean square error by driver and model.

	Wiedemann	Wiedemann with GHR	% improvement
Driver J	0.6684	0.6269	6%
Driver K	0.6042	0.4628	23%
Driver F	0.5980	0.3423	43%
Driver L	0.8106	0.7715	5%

Conclusions

The car-following data for four different drivers from the naturalistic data was used to calibrate the models using a genetic algorithm. Ten car-following periods per driver were used in the calibration. The framework consisted of calibrating the parameters for the Wiedemann model based on the equations for the thresholds and accelerations in the Wiedemann model. This framework was then altered such that the GHR model replaced the acceleration equations in the following regimes: approaching, closely approaching, acceleration following, and deceleration following. The GHR model was given a different set of calibration parameters for each of those regimes. The reaction time, T, was used as a calibration parameter to obtain a measure of the attentiveness of the different drivers. The combination of the Wiedemann model and the GHR model presents advantages when calibrating to the car-following behaviors of individual drivers. The hybrid Wiedemann-GHR model calibrated to four individual drivers' results with 5 percent to 43 percent less error than the Wiedemann model alone. Future research should look into the transition between following and free-driving regimes and the driver actions in the emergency regime.

MODEL COMPARISON

The method employed by this research effort selected, calibrated, and compared various car-following models. The models chosen for comparison, shown in Table 15, are GHR, Wiedemann, Fritzsche, Gipps, IDM, and VDIFF. These specific models were chosen because they represent a variety of the types of commonly used car-following models. Each model was calibrated by the use of a genetic algorithm. A genetic algorithm was used because of its ability to adequately and accurately find the optimal solution when multiple parameters are present, as in some of the models.

These models were calibrated using more than 100 car-following periods for each of the four truck drivers. These four truck drivers were chosen because they represent a wide spectrum of the population. When classifying by the number of driving conflicts each driver experienced, Driver 49 represents a driver with a low number of conflicts, Drivers 48 and 97 represent drivers with an average number of conflicts, and Driver 64 represents a driver with a high number of conflicts. Table 16shows the root mean squared error that resulted from calibrating these car-following models to the four chosen drivers.

Table 15. Car-following models and parameters.

GHR	$a_f(t+T_r) = cv_f^m(t+T_r)\frac{\Delta v(t)}{\Delta x^l(t)}$	T_r: time between the observation of a certain stimulus and the reaction to that stimulus $a_f(t+T_r)$: acceleration of the following vehicle at time $t+T_r$ $v_f(t+T_r)$: speed of following vehicle at time $t+T_r$ $\Delta v(t)$: relative speed between the following car and the car immediately in front $(v_{leader} - v_{follower})$ $\Delta x(t)$: relative distance between following car and car immediately in front $(x_{leader} - x_{follower})$ m, l, c: parameters describing car-following behavior
Gipps		a_n^{max} : maximum desired acceleration, vehicle *n* (m/s^2) d_n^{max} : maximum desired deceleration, vehicle *n* (m/s^2)

	$v_a^n(t+T) = v_c(t) + 2.5\, a_n^{max}\ T\left(1 - \frac{v_n(t)}{v_n^{desired}}\right).\sqrt{0.025 + \frac{v_n(t)}{v_n^{desire\ d}}}$ $v_n^b(t+T) = d_n^{max}.T$ $.\sqrt{(d_n^{max}.T)^2 - d_n^{max}.\left[2\{x_{n-1}(t) - s_{n-1}(t)\} - v_n(t).T - -\frac{v_n(t)^2}{\hat{d}_{n-1}}\right]}$	$\hat{d}_{n-1}$: estimation of maximum deceleration desired by vehicle *n-1* (m/s^2) s_{n-1} : effective length of vehicle T: reaction time $\hat{d}_{n-1}$: leader desired deceleration x_{n-1} : vehicle spacing
Intelligent Driver Model (IDM)	$a_n(t+\tau) = a\left[1 - \left(\frac{v_n(t)}{v^*}\right)^{\delta} - \left(\frac{\Delta x^*_{min}\,(v_n(t), \Delta v_{n,n-1(t)})}{\Delta x_{n-1,n}(t)}\right)^2\right]$ $\Delta x^*_{min}\left(v_n(t), \Delta v_{n,n-1(t)}\right) = d + T_{safe} * v_n(t) + \frac{v_n(t)\Delta v_{n,n-1(t)}}{2\sqrt{a^{max} * b^{\max\ abs}}}$	$\Delta v_{n,n-1}(t)$: approaching rate of following vehicle v^*: desired speed T_{safe}: safe time headway a^{max}: maximum desired acceleration of following vehicle $b^{\max\ abs}$: absolute maximum desired deceleration of following vehicle δ: acceleration component $\Delta x_{n-1,n}(t)$: distance headway d: vehicle length
Velocity difference model (VDIFF)	$a(s, v, \Delta v) = \frac{v_{opt}(s) - v}{\tau} - \lambda \Delta v$ $v_{opt}(s) = \frac{v_0}{2}\left[tan\left(\frac{s}{l_{int}} - \beta\right) - \tan(-\beta)\right]$	s: distance headway Δv: relative velocity τ: relaxation time λ: sensitivity

		parameter v_0: desired velocity l_{int}: interaction length β: form factor
Wiedemann model	$AX = L_{n-1} + AXadd + RND1_n \times AXmult$ $ABX = AX + BX$ $BX = (BXadd + BXmult \times RND1_n) \times \sqrt{v}$ $v = \begin{cases} v_{n-1} \; for \; v_n > v_{n-1} \\ v_n \quad for \; v_n \leq v_{n-1} \end{cases}$ $SDX = AX + EX \times BX$ $EX = (EXadd + EXmult \times (NRND - RND2_n))$ $SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX}\right)^2$ $CX = CXconst \times (CXadd + CXmult \times (NRND1_n + RND2_n))$ $OPDV = CLDV \times (- OPDVadd - OPDVmult \times NRND)$ $AD = S_{n-1} + T_D \times v_n$ $AR = S_{n-1} + T_r \times v_{n-1}$ $AS = S_{n-1} + T_s \times v_n$ $\Delta b_m = \lvert b_{min} \rvert + a_{n-1}$ $AB = AR + \frac{\Delta v^2}{\Delta b_m}$	AX: the desired distance between stationary vehicles ABX: the desired minimum following distance at low speed differences SDX: the threshold for maximum following distance SDV: the point which the driver notices that he or she is approaching a slower vehicle OPDV: increasing speed difference L_{n-1}: leading vehicle length *AXadd* and *AXmult*: calibration parameters $RND1_n$: a normally distributed random number for vehicle *n* v: vehicle speed *CXconst*, *CXadd*, and *CXmult*: calibration parameters *NRDV*: a normallydistributed random number

		OPDVadd, OPDVmult: calibration parameters
Fritzsche	$AD = S_{n-1} + T_D \times v_n$ $AR = S_{n-1} + T_r \times v_{n-1}$ $AS = S_{n-1} + T_s \times v_n$ $\Delta b_m = \lvert b_{min} \rvert + a_{n-1}$ $AB = AR + \frac{\Delta v^2}{\Delta b_m}$	AD: desired distance threshold AR: risk distance threshold AS: safe distance threshold AB: risk braking distance threshold T_D: desired time gap S_{n-1}: effective length of vehicle v_n: following vehicle speed T_r: risky time gap T_s: safe distance gap b_{min} , a_{n-1} : parameter controlling maximum deceleration

Table 16. Root mean squared error for each model by driver.

	Driver J	Driver K	Driver F	Driver L
GHR	0.9294	0.9533	1.1892	4.3815
Gipps	1.3894	1.2036	2.3112	3.8024
IDM	1.2260	1.2215	1.9211	1.0985
VDIFF	1.1082	1.0848	1.1990	1.2454
Wiedemann	0.7541	1.0916	1.4735	1.0764
Fritzsche	2.2094	1.3413	1.4332	1.1242

Table 17 shows the sum of squared error for each model by driver. The rightmost column presents the root mean squared error if the sum of squared error is summed for all the drivers and then converted to a root mean squared error. This provides a means of comparing the different models, if a single model were to be used for all of the drivers. The results show that the Wiedemann model results in the least root mean squared error, with the velocity difference model (VDIFF) not far behind.

Table 17. Sum of squared error for the models by driver with a root mean squared error of the total.

	Driver J	Driver K	Driver F	Driver L	Total	RMSE
GHR	78774	104980	177220	1752741	2113714	2.2345
Gipps	176030	167333	669436	1320099	2332898	2.3475
IDM	137063	172342	462506	110167	882078	1.4435
VDIFF	111990	135938	180169	141616	569713	1.1601
Wiedemann	51749	108426	197567	105430	463172	1.0460
Fritzsche	445138	207813	257427	115386	1025765	1.5566

Table 18 shows the parameter values of the GHR model for each driver. The GHR model showed the least root mean squared error for two of the drivers, but with the inclusion of the other drivers, this model shows a greater root mean squared error than some of the other models. This means only that the GHR model can accurately mimic some drivers, but it is not sufficient to model all the drivers presented here. The structure of the model causes this to occur because some of the drivers exhibit behavior that is more complex than the GHR model can handle.

Table 18. GHR parameter values by driver.

	Driver J	Driver K	Driver F	Driver L
c	-3.553	-0.959	-0.542	0.300
m	-0.253	0.264	1.068	0.000
l	0.644	0.576	1.000	1.000
T	1.791	0.272	1.087	1.000
RMSE	0.929	0.953	1.189	4.381

Table 19 presents the calibrated parameters of the Gipps model for each of the drivers. The Gipps model does not calibrate as well as the other models, as shown by the root mean squared error. The Gipps model shows a large increase in error for Drivers F and L as compared to Drivers J and K. Table 19 suggests that the Gipps model is not as adequate as the other models when representing multiple drivers. The Gipps model is originally a two-regime macroscopic traffic flow model. That characteristic is obvious when the Gipps model is compared to other microscopic models, as this model tends to represent steady-state behavior.

Table 19. Gipps parameter values by driver.

	Driver J	Driver K	Driver F	Driver L
b	0.2105	0.2740	-0.0164	-0.4231
b'	6.3986	9.7948	0.5938	0.0916
Ln-1	90.1440	19.3718	10.5249	3.0981
Un	92.1683	135.6518	26.4435	25.0118
an	7.3557	7.9477	-0.0004	0.5487
T	1.9306	0.9083	0.9974	0.2282
RMSE	1.3894	1.2036	2.3112	3.8024

Error! Reference source not found. presents the calibrated parameters of the intelligent driver m odel for the four drivers. The intelligent driver model shows a root mean squared error very

similar to that of the Wiedemann model for Driver L, but greater root mean squared error values for the other drivers. In Table 20, the intelligent driver model shows a root mean squared error that suggests it is better at representing multiple drivers than some other models, but it is not as adequate as they are. This means that the intelligent driver model is not as accurate as other models when representing one driver, but when multiple drivers are considered, it can outperform some other models.

Table 20. IDM parameter values by driver.

	Driver J	Driver K	Driver F	Driver L
a	2.6519	3.3372	5.8950	0.8724
vdes	106.5559	97.3445	144.7275	96.0030
s0	27.5776	8.8639	2.5513	15.6821
T	0.0831	2.2913	2.3686	2.1768
b	1.0143	1.0593	67.6662	0.3113
RMSE	1.2260	1.2215	1.9211	1.0985

Table 21 presents the calibrated parameters for the velocity difference model of the four drivers. The values of the free-flow speed and desired velocity suggest that their use in this model will only cause error, and that a replacement with a calibration parameter would yield a more accurate representation of the actual behavior of drivers. The velocity difference model shows a root mean squared error that is similar and not too distant from that of the Wiedemann model. The velocity difference model also shows very stable behavior across the different drivers in Table 21. This shows that the velocity difference model can adequately represent the behavior of different drivers in a manner that exceeds or is on par with the other models.

Table 21. VDIFF parameter values by driver.

	Driver J	Driver K	Driver F	Driver L
FFS	48.5177	79.9246	123.4765	155.2547
Ln	5.0480	50.4894	36.6056	17.3944
Lag time	0.9870	3.5970	0.2989	2.4494
Sensitivity	12.7509	13.6386	1.0580	8.2774
Vdes	93.1806	106.0174	33.6502	76.6298
Form factor	4.6059	6.9404	0.9134	7.1708
l	8.2814	6.0162	6.6070	5.2652
tau	4.4006	3.4847	1.3662	2.9527
RMSE	1.1082	1.0848	1.1990	1.2454

Table 22 shows the calibrated parameters of the Wiedemann model for the four drivers. The drivers show different null acceleration values and different threshold parameters, which highlights that the drivers have different preferences for accelerating and decelerating while following. This also shows that the drivers make the decision to change accelerations at different points. The flexibility available in the Wiedemann model undoubtedly contributes to its performance in mimicking the behavior of real drivers. The Wiedemann model shows stable behavior across the different drivers, just like the velocity difference model, but with a small hiccup for Driver F. The intelligent driver model and the Gipps model showed this same hiccup

for Driver F. Nevertheless, the Wiedemann model represents different drivers well, as shown in Table 22.

Table 22. Wiedemann parameter values by driver.

	Driver J	Driver K	Driver F	Driver L
Ln-1	4.4938	5.5533	4.5469	5.6797
AX	2.5254	5.0441	2.8730	5.6009
BX	3.0408	3.4054	3.4211	3.4349
EX	2.5139	3.0961	3.4273	2.6760
CX	40.0659	75.0421	62.4053	93.1305
CX2	39.0314	28.9785	73.2578	73.0260
CLDVCX	31.7622	56.7976	33.1589	51.0795
OPDV	-2.5116	-6.5796	-1.0677	-3.3071
bnull	0.1224	0.0747	0.2408	0.0716
bmaxmult	0.0991	0.1533	0.4271	0.3871
FaktorVmult	0.0389	0.1670	0.2002	0.1820
bminadd	-20.8984	-13.1674	-23.4076	-41.7209
bminmult	0.0334	0.0913	0.3202	0.3784
Vdes	42.6738	84.0379	88.7207	16.0671
FaktorV	0.9396	0.5216	0.5065	1.9585
RMSE	0.7541	1.0916	1.4735	1.0764

Table 23 presents the calibrated parameters of the Fritzsche model for the four drivers. The model shows similar null acceleration values for three of the drivers. Three of the drivers also show similar *an-1* and *an+* values. The Fritzsche model is very similar to the Wiedemann model in the way it operates, but the root mean squared error shows that the Fritzsche model does not perform similarly to the Wiedemann model. The Fritzsche model shows a competitive root mean squared error for Driver L, but it is lacking with the other drivers. This inconsistency shows in with the Fritzsche model having mediocre performance compared to the other models, which highlights its inadequacy for mimicking multiple drivers.

Table 23. Fritzsche parameter value by driver.

	Driver J	Driver K	Driver F	Driver L
Sn-1	22.6295	2.0673	2.0673	1.8332
kPTN	5.4213	4.8326	4.8326	4.0063
kPTP	1.1493	3.0874	3.0874	1.9313
fx	19.3430	-62.8779	-62.8779	-62.5547
Td	4.4129	4.8104	4.8104	3.9579
Tr	1.8111	0.8955	0.8955	1.0082
Ts	2.9147	3.6792	3.6792	3.8147
bmin	-2.4743	-0.0955	-0.0955	-0.8210
an-1	-4.5296	6.8066	6.8066	6.0010
bnull	-6.0129	6.7731	6.7731	6.4286
an+	-5.3233	-5.3827	-5.3827	-3.5638
RMSE	2.2094	1.3413	1.4332	1.1242

$$FaktorV = \frac{v_{max}}{v_{des} + FAKTORVmult * (v_{max} - v_{des})}$$

Figure 55 presents an example fitted car-following period for Driver L that is representative of the observations seen in other car-following periods for all four drivers. Most of the models show a behavior that is heavily influenced by the lead vehicle, as shown by the lines for the models overlapping the line for the lead vehicle in the figure. The actual behavior of the driver shows less dependence on the behavior of the lead vehicle, as shown by the black line in

$$FaktorV = \frac{v_{max}}{v_{des} + FAKTORVmult * (v_{max} - v_{des})}$$

Figure 55. The GHR and Gipps models show large error in this car-following period. The error of the Gipps model is the constant velocity, while the error of the GHR model appears to be that this car-following period represents behavior that the calibrated GHR model does not adequately capture. The rest of the models show similar behavior that, as mentioned before, is heavily influenced by the actions of the lead vehicle.

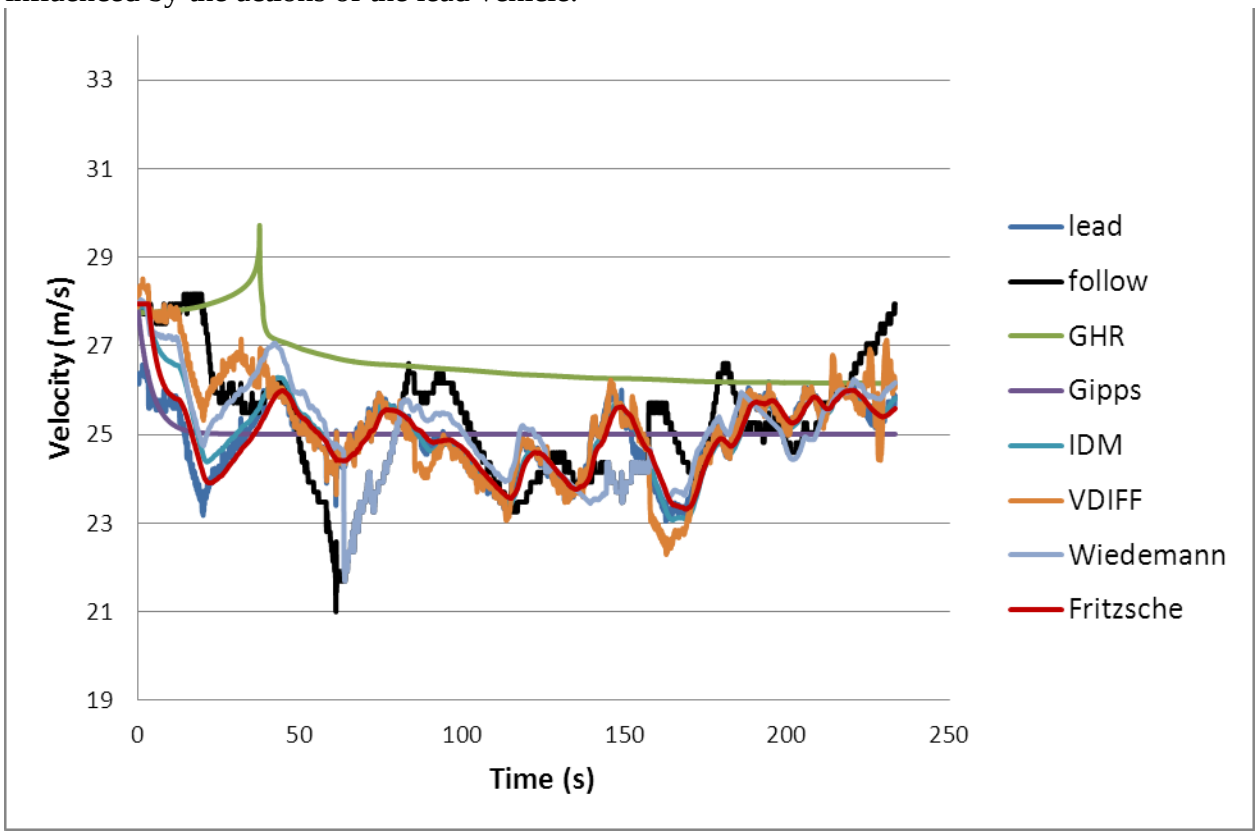

Figure 64. Graph. Plot of models as compared to the data for Driver L.

Conclusions

The results show that some of the microscopic traffic flow models calibrate to match real drivers better than others. The results also show that some of the models are more adequate at mimicking different truck drivers. Most of the models show a behavior that is heavily influenced by the actions of the lead vehicle. The results of this research effort suggest that the velocity difference model and the Wiedemann model can adequately represent the behavior of different drivers. This means that if a single car-following model is used with data from different drivers

individually, these two models show the most promise in being up to the task. It is important to note that these findings are based solely upon data for truck drivers, so the applicability of these findings to car drivers' behavior would be questionable and thus is recommended as an area of future research. Further research is also recommended in testing these two models with a larger number of individual drivers to clarify the strengths and weaknesses of each model.

CHAPTER CONCLUSION

The naturalistic data offers new insights into driver behavior due to the sheer amount of car-following data that is available for each driver. The amount of data allows for the discovery of new thresholds for the Wiedemann model: a passing threshold and a hook following threshold. These two new thresholds increase the usability and accuracy of the Wiedemann model when natural or naturalistic driver behaviors are concerned because these thresholds were developed based on observations made in the naturalistic data. Combining the strengths of both the GHR model and the Wiedemann model into a hybrid yields a more accurate and highly flexible model. The only downside to this combination is the introduction of additional calibration parameters, which makes the model more difficult to calibrate. The comparison of different car-following models resulted in finding that different models have the least error for different drivers, but the velocity difference model and the Wiedemann model both perform adequately across all the drivers tested.

Chapter 5. AGENT DEVELOPMENT AND TRAINING WITH REINFORCEMENT LEARNING

Background Information

The goal of this research task is to model naturalistic vehicle actions during safety critical events. This was to be achieved by simulating driving policy to associate observed traffic state to vehicle actions. Driving policy should be driver independent and consist of multiple driving rules that are associated with different states. Reinforcement learning is used to determine the sub optimal action for driving rules. Fuzzy logic and reinforcement learning are coordinated to extract driving policy and embed the policy into an agent simulator.

Safety critical events in this study are located in the emergency regime. Most car-following models, such as the Wiedemann model, assume that drivers will always keep a safe distance so that vehicles should always be able to stop before hitting lead vehicles in an emerging conflict. However, in real-world cases, driver actions are not always just longitudinal deceleration. For example, a driver may make a maneuver and execute a lane change. Also, as traffic stimuli and causalities sometimes vary case by case, it is very hard to establish predefined longitudinal and lateral action models. So far, no lateral action models have been used to simulate actions for safety critical events.

Traffic States and Actions during Safety Critical Events

Drivers behave according to the traffic state they experience. Traffic state is defined by a set of variables that can represent a vehicle's kinematic conditions and its environment. In fact, in most safety critical events, a vehicle's action is closely related to its interaction with a leading vehicle. As most of car-following model suggests, vehicle speed, distance from the leading vehicle, and relative speeds are considered traffic state variables. Additionally, we assume driving during safety critical events is a sequential task in which driver actions are continuous, so that earlier actions may contribute to current action decisions. For example, when a driver decides to decelerate in the previous state, he or she is more likely to continue decelerating in the current state instead of executing a maneuver. Therefore, actions of the previous state are considered to be variables of the current state.

Agent-based Modeling of Driver Behavior

Behavior rules of individual drivers should be used to teach an agent simulator (of an individual driver). Behavior rules associate actions with traffic state and provide a driver-dependent driving policy for its agent to follow. So when an agent experiences a certain traffic state, the policy will map the traffic state to associated actions. By using naturalistic driving data in training, agents will learn to adopt driving rules during the training procedure and should be capable of replicating driver and vehicle actions when training is completed. Accordingly, if data on safety critical events are used in training, agents will perform driver-specific naturalistic actions, which could probably result in a crash or near-crash.

Reinforcement Leaning: Brief Introduction

Reinforcement learning is a relatively new methodology for developing artificial agents. Reinforcement learning is a subarea of machine learning in computer science that determines the actions an intelligent agent should take in an environment to maximize some notion of long-term goals (Sutton 1988). The objective of using reinforcement learning algorithms is to find a policy that maps traffic states to their optimal actions. In our research scope, the agent driving policy should be close enough to the "guiding" driver. One of the remarkable research papers by Jouffe (1998) illustrates the basic mechanism of reinforcement learning. Reinforcement learning reinforces agent actions when they perform approximately close to naturalistic actions and penalize actions that are far away from them. The only information available for learning is the system feedback, which describes the agent's task in terms of reward and punishment. At each time step, the agent receives a reinforcement signal regarding the last action it performed in the previous state. The challenge is optimizing not only the direct reinforcement, but also the total amount of reinforcements the agent can receive in the future. Finally, reinforcement learning should extract driving rules from a naturalistic dataset and establish similar driver-specific state action mapping rules.

Potential Shortcomings of Existing Reinforcement Learning Algorithms

Although these research efforts have achieved great accomplishments, many limitations still exist. The dimension problem is a great difficulty of most agent controllers and simulators. If the number of traffic state variables is large, computation time would become a burden in training. Because the reinforcement learning mechanism has to associate state with action and update mapping rules, more dimensions in traffic state will cause an exponentially increasing number of mapping rules.

Moreover, the existing reinforcement learning efforts mainly deal with discrete state-action mapping problems so that mapping rules can always could refer to a look-up table to relate a discrete traffic state to a discrete action. However, traffic state variables (vehicle speed, for example), especially in our study, have continuous distributions, which makes it impossible to build a look-up table to relate states to actions.

Another significant drawback of the established reinforcement learning algorithm in transportation is the inability to generate continuous actions. Instead, most studies use discrete action sets consisting of a limited number of actions for the agent to select. Abdulhai's (2003) Q-learning mechanism has a binary action option phase: extension or end. Zhang (2007) improved this action selection methodology and enabled controller agents to select one of the six actions. This problem limits established reinforcement learning algorithms in continuous action simulation.

The Proposed Reinforcement Learning Methodology

In this research task, we proposed a revised reinforcement learning methodology to solve the traffic state dimensional problem and continuous action generation problem. In fact, this revised neuro-fuzzy actor-critic reinforcement learning (NFACRL) approach can alleviate most of the computation burden that was associated with the previously developed approaches. Using naturalistic traffic states and driving actions during crash and near-crash events, this approach can reproduce actual driver behavior during safety critical events.

This chapter first proposes a revised NFACRL methodology, including its framework as well as input and output components. Second, the naturalistic driving database and safety critical events extraction process are described. Third, the agent training results are presented with a performance evaluation. Finally, cross validation between different agents is performed, driver heterogeneity is illustrated, and the idea of a mega-agent is proposed.

AGENT TRAINING METHODOLOGY: NFACRL

NFACRL Structure

NFACRL is a type of actor-critic reinforcement learning method. In this study, NFACRL uses current traffic state as inputs and generates vehicle actions as outputs. NFACRL then loads the state data of following states and decides the following actions. As its name indicates, NFACRL has a neural network structure and uses fuzzy logic to cluster traffic state variables and transfer information between layers. Accordingly, continuous traffic states are converted into a limited number of discrete fuzzy sets. Fuzzy rules are considered state-action mapping rules to associate fuzzy sets and actions. Reinforcement learning is the training process to obtain fuzzy rules.

As Figure 65 shows, our proposed NFACRL structure has four layers. The first layer is the input layer. Each of its nodes represents a state variable. The second layer is the fuzzy membership layer. Continuous state variables are fuzzified into linguistic terms as “Speed is High” and “Speed is Low” in this layer. Each node is a fuzzy set with a membership function as output. Notice that one state input variable should have more than one fuzzy set so that continuous input variables can be transferred to discrete sets without losing much information. Fuzzy membership functions can be triangular, trapezoidal, and Gaussian (Jiang 1997), depending on the distributions of state variables. The third layer is the fuzzy rules layer. Each rule is connected with a number of antecedents (discrete fuzzy sets) from fuzzy set layer. The firing strength is the output of a fuzzy rule and indicates the rule’s strength. The fourth layer is the actor-critic layer, which includes actor and critic nodes. Critic nodes are associated with the value of the next state after the agent chooses an action. Actor nodes are from one discrete action set. Each fuzzy rule selects one discrete action. NFACRL output action is the weighted average of all the actions selected by fuzzy rules in which fuzzy rule strengths are the weights. The following several paragraphs elaborate more on the framework design.

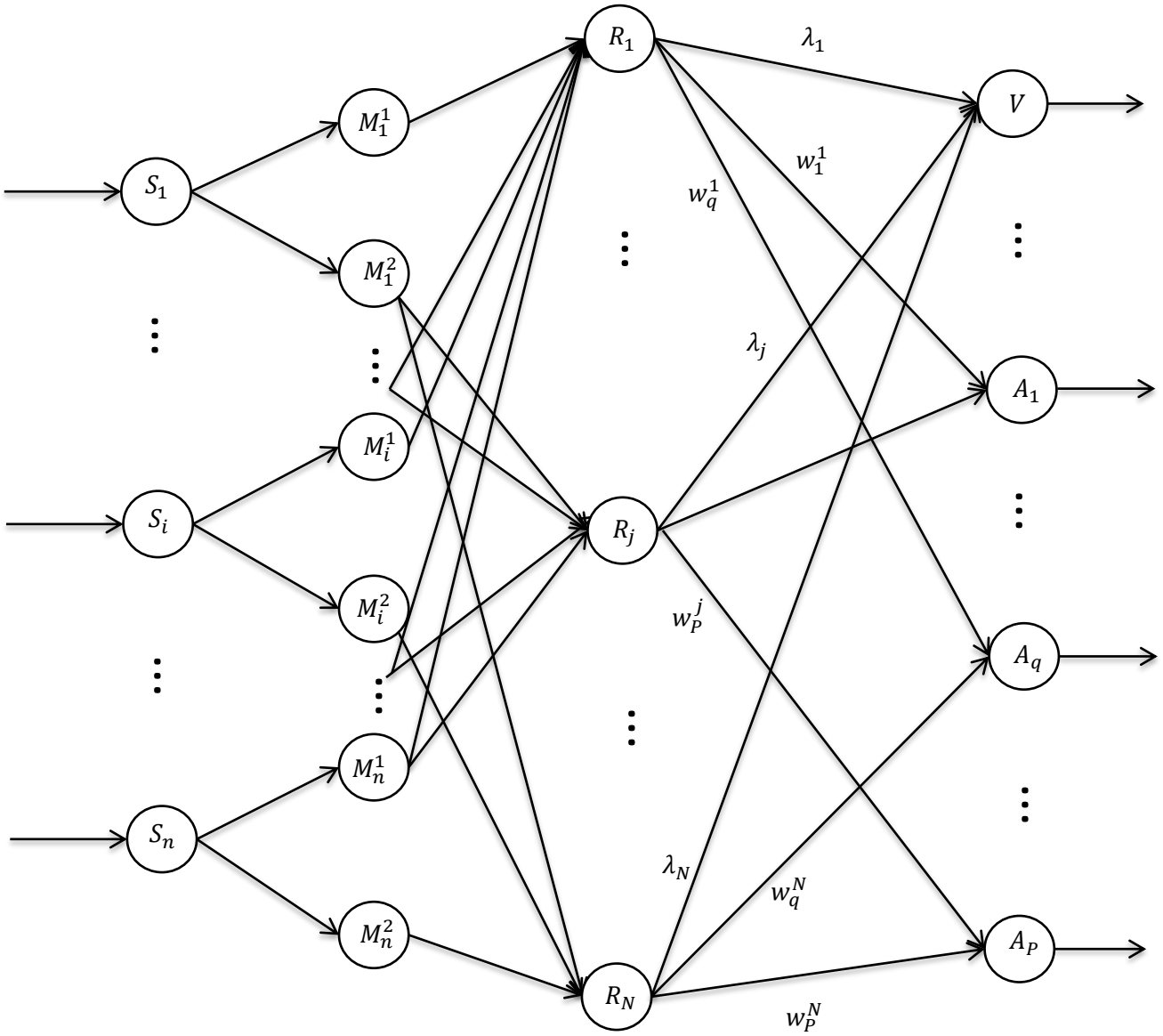

Figure 65. Diagram. NFACRL structure.

S_i is the i^{th} input variable (state).

n is the number of input variables.

NM_i is the number of fuzzy sets or membership functions for the S_i.

$M_i^{a(i)}$ is the $a(i)^{th}$ fuzzy set or membership function for the i^{th} input variable.

R_j is the j^{th} fuzzy rule.

N is the number of fuzzy rules.

λ_j is the weight between the j^{th} fuzzy rule and critic.

w_q^j is the weight between the j^{th} fuzzy rule and action q.

V is the critic output.

A_q is the output of the q^{th} discrete action.
Where $i = 1, \dots n, a(i) = 1, .. NM_i, j = 1, \dots, m$ and $q = 1, .., P$

Layer Design

1) State Layer

Most safety-distance-based car-following models assume speed, relative speed, and distance as important stimuli when leading and following vehicles are interacting with each other. In addition, during safety critical events, these three variables are also incorporated into state variables because the lateral lane offset (lateral position relative to the center of the lane), acceleration, and vehicle yaw angle (angle of vehicle longitudinal relative axis to lane markings) of the previous state are also important to the driver action decision process. Traffic state variables in the NFACRL structure are defined as in the equation in Figure 66.

$$S_1 = v$$
$$S_2 = \Delta x$$
$$S_3 = \Delta v$$
$$S_4 = a'$$
$$S_5 = y'$$
$$\boldsymbol{S_6 = o}$$

Figure 66. Equation. Traffic state variables.

where S_i is the ith state variable, v is the vehicle speed, Δx is the space headway (relative distance from the leading vehicle), Δv is the relative speed (speed of the leading vehicle minus the speed of the following vehicle), a' is the previous acceleration, y' is the previous yaw angle, and o is the vehicle lateral position offset.

2) Fuzzy Sets Layer

Each state variable in this study is linked to two fuzzy sets in the second layer. These two fuzzy sets are defined in linguistic terms: "S_i is high" and "S_i is low" for state variable S_i. Each fuzzy set is associated with a membership function. A triangular fuzzy membership function is used to make uniform a state variable and transfer its information to the two fuzzy sets. The triangular membership function requires upper and lower bounds of each state variable, which could be extracted from safety critical events that the driver experienced. For each state, the lower bound and upper bound are the maximum and minimum state variable values. During the training process, upper and lower bounds are considered to be constant.

Membership functions for fuzzy sets Low and High are defined as:

$$\mu_{Low}(S_i) = \begin{cases} 1 & S_i \le S_{lb,i} \\ \dfrac{S_{ub,i} - S_i}{S_{ub,i} - S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i}\ i = 1, 2 \dots 6 \\ 0 & S_i \ge S_{ub,i} \end{cases}$$

Figure 67. Equation. Membership function for Low fuzzy set.

$$\mu_{High}(S_i) = \begin{cases} 0 & S_i \le S_{lb,i} \\ \dfrac{S_i - S_{lb,i}}{S_{ub,i} - S_{lb,i}} & S_{lb,i} < S_i < S_{ub,i}\ i = 1, 2 \dots 6 \\ 1 & S_i \ge S_{ub,i} \end{cases}$$

Figure 68. Equation. Membership function for High fuzzy set.

where $\mu_{Low}(S_i)$ is the membership function for fuzzy set "S_i is Low", $\mu_{High}(S_i)$ is the membership function for fuzzy set "S_i is High", $S_{lb,i}$ is the lower bound of state variable S_i, and $S_{ub,i}$ is the upper bound of state variable S_i.

3) Fuzzy Rules Layer

Fuzzy rules in the third layer provide a state action mapping policy to determine the actions to select from in the fourth layer. Each fuzzy rule is applied to one combination of fuzzy sets from the second layer. For example, a fuzzy rule can be represented as

WHEN "S_1 is low" and "S_2 is low" and "S_3 is high" and "S_4 is high" and "S_5 is high" and "S_6 is high", THEN Action "Deceleration $a = -0.05g$"

In our designed neural network, each fuzzy rule is associated with six fuzzy sets originated from six continuous state variables. Because each state variable has two fuzzy sets, "Low" and "High," the number of fuzzy set combinations should be $2^6 = 64$. Equivalently, there are 16 fuzzy rules in this study.

The product of the membership functions connected with each rule is used to compute its firing strength. Firing strength for the j^{th} fuzzy rule is shown in Figure 69.

$$FS_{Rj} = \prod_{i=1}^{6} \mu^{j}_{a(i)}(S_i)$$

Figure 69. Equation. Firing strength for the j^{th} fuzzy rule.

where $a(i)$ is the linguistic term of the fuzzy set (either "Low" or "High") for the i^{th} input state variable, and j represents the j^{th} fuzzy rule.

4) Actor and Critic Layer

Critic nodes represent the value of the next state under the current policy. Actor nodes represent actions to be selected by the fuzzy rules. Two types of actions, acceleration and yaw angle, are used as equivalents to vehicle longitudinal and lateral actions. Notice that acceleration and yaw

angle are continuous variables. However, due to the constraints of neural network structure, it is impossible to enumerate all the possible continuous actions. Instead, discrete actions sets are defined here to include a number of representative action values for fuzzy rules to select. In this study, five discrete longitudinal acceleration values and five lateral yaw angle values are considered action candidates. Two sets are used to store actions, one for acceleration A and one for yaw angle Y (see Figure 70). Ten action values are considered as constant parameters during training.

$$A = \{a_{d1}, a_{d2}, a_{d3}, a_{d4}, a_{d5}\}$$
$$Y = \{y_{d1}, y_{d2}, y_{d3}, y_{d4}, y_{d5}\}$$

Figure 70. Equation. Acceleration and yaw angle sets.

5) Neuron Weights

Weights are located between the fuzzy rule layer and the action layer. Critic weight λ^j links the j^{th} fuzzy rule to critic output. Action w_k^j links the j^{th} fuzzy rule to the k^{th} action output. Weights w_k^j show competition between discrete actions in the same action set and are used as references to the action selection methodology for fuzzy rule j. Weights λ^j are used to compute the value of traffic state V_S, which is the sum product of firing strength FS_{R_j} and weight λ^j (see Figure 71).

$$\boldsymbol{V_s = \sum_{j=1}^{64} FS_{R_j} \lambda_j}$$

Figure 71. Equation. Traffic state V_S.

where FS_{R_j} is the firing strength for the j^{th}fuzzy rule R_j and V_s is the value of state S.

NFACRL Output Actions

1) Greedy Selection Algorithm

Fuzzy rule R_j selects discrete action a_{dk} from set A when weight w_{ak}^j is the largest among accelerations $a_{d1}$to a_{d5}. Similarly, R_j selects discrete action y_{dk} from set Y when weight $w_{yk}^j$is the largest in set Y.

2) Continuous Action Output

For each fuzzy rule, one discrete acceleration and one discrete yaw angle are selected. Therefore, the output actions acceleration a and yaw angle y are the weighted average of all the selected discrete actions by all the rules. Firing strengths of fuzzy rules are used as weights, so NFACRL agent actions are generated as seen in Figure 72 and Figure 73.

$$\boldsymbol{a = \sum_{j=1}^{64} FS_{R_j} * a_{dk}^j}$$

Figure 72. Equation. Continuous longitudinal acceleration.

$$y = \sum_{j=1}^{64} FS_{R_j} * y_{dk}^{j}$$

Figure 73. Equation. Lateral yaw angle.

where a is continuous longitudinal acceleration and y is lateral yaw angle.

Instead of generating discrete actions, this NFACRL method is able to generate continuous action variables.

Weights Update: Reinforcement Learning Algorithm

The reinforcement learning algorithm updates the weights between the third layer and the fourth layer.

1) Weights Update

Weights $\lambda^{j}$and w_k^j are updated according to temporal difference (TD) error (Sutton 1988). TD error consists of the value of the current state, the value of the following state, and a predefined reward function. The reward function is an estimation of how close the selected actions are at state S_t.

Temporal difference (TD) error is calculated as seen in Figure 74 and Figure 75.

$$\delta_{a,t} = r_{a,t+1} + \gamma V_{s_{t+1}} - V_{s_t}$$

Figure 74. Equation. Temporal difference for acceleration.

$$\delta_{y,t} = r_{y,t+1} + \gamma V_{s_{t+1}} - V_{s_t}$$

Figure 75. Equation. Temporal difference for yaw angle.

where $r_{a,t+1}$is the reward function when action a is taken at state S_t, $r_{q,t+1}$is the reward function when action y is taken, γ is the discounting factor, V_{s_t} is the value of the current state, and $V_{s_{t+1}}$ is the value of the following state.

$V_{s_{t+1}}$ is calculated as seen in Figure 76.

$$V_{s_{t+1}} = \sum_{j=1}^{N} FS_{R_j,t+1}\lambda_{j,t}$$

Figure 76. Equation. Value of the following state.

Notice that as the NFACRL agent generates two actions, two responding reward functions are computed and two temporal difference errors are calculated.

λ and w are updated by TD errors (see Figure 77, Figure 78, Figure 79, and Figure 80), where

$$\lambda_{a,t+1}^{j} = \lambda_{a,t}^{j} + \beta\delta_{a,t}FS_{R_j,t}$$

Figure 77. Equation. Critic weights update

$$\lambda_{y,t+1}^{j} = \lambda_{y,t}^{j} + \beta\delta_{y,t}FS_{R_j,t}$$

Figure 78. Equation. Critic weights update

$$w^j_{ak,t+1} = w^j_{ak,t} + \beta\delta_{t,a}FS_{R_j,t}$$

Figure 79. Equation. Actor weights update

$$w^j_{yk,t+1} = w^j_{yk,t} + \beta\delta_{y,a}FS_{R_j,t}$$

Figure 80. Equation. Actor weights update

where β is the learning rate.

TD errors update only the weights connected to the selected discrete actions a and y. From the equations in Figure 79 and Figure 80, when discrete action a_{dk}/y_{dk} is selected by rule R_j, only weight $w^j_{ak,t}/w^j_{yk,t}$ is updated.

2) Reward Function

The reward function provides guidance for an agent to follow. The reward function encourages an agent to take actions that are close to driver actions and penalizes actions that are dissimilar. The agent will know the outcome of the actions only when they have been chosen. When the performance of an action outcome is good, the reward function should be positive to provide a greater probability of the agent choosing those actions in the future. When performance is bad, the reward function should be negative to reduce the probability of the agent making those choices again. In this project, actions from the naturalistic driving database were used as references.

First, relative error is calculated as in Figure 81

$$e_a = |\frac{a-a_n}{a_n}|$$

Figure 81. Equation. Relative error.

for acceleration and as in Figure 82

$$e_y = |\frac{y-y_n}{y_n}|$$

Figure 82. Equation. Yaw angle.

for yaw angle, where e_a is the absolute relative error of acceleration, e_y is the absolute relative error of yaw angle, and a_n and y_n are the naturalistic actions from database.

Then, a nonnegative parameter e_{th} is defined as an acceptance threshold. When e_a or e_y is less than e_{th}, the reward function (see Figure 83 and Figure 84) is positive and related weights will increase, which increases the probability of fuzzy rules selecting a and y in the future.

$$r_a = \alpha(e_{th} - e_a)$$

Figure 83. Equation. Reward function of acceleration.

$$r_y = \alpha(e_{th} - e_y)$$

Figure 84. Equation. Reward function of yaw angle.

where r_a is the reward function of acceleration, r_y is the reward function of yaw angle, and α is the scaling factor.

Initially, when an agent does not know which discrete action to take, it may choose an extremely inappropriate action, causing e_a and e_y to become too large. When that condition occurs, the reward function becomes negative enough to interrupt the weights update. Therefore, reward functions need to be adjusted to make sure they have a lower bound (see Figure 85 and Figure 86).

$$\boldsymbol{r_a = -\alpha} \quad \text{when } \boldsymbol{e_a \geq 1}$$

Figure 85. Equation. Reward function of acceleration with lower bound.

$$\boldsymbol{r_y = -\alpha} \quad \text{when } \boldsymbol{e_y \geq 1}$$

Figure 86. Equation. Reward function of yaw angle with lower bound.

NATURALISITIC DATA EXTRACTION AND AGENT TRAINING PROCEDURE

Database Description

In agent training, we use safety critical events from the 8-Truck database of the Naturalistic Truck Driving Study (NTDS), the 34-Truck database, and the 100-Car database collected by VTTI. See Chapter 3 for further descriptions of these studies.

In our research focus, the following vehicle was the instrumented vehicle. The measured subject vehicle data included speed, yaw angle, lane offset, and accelerations. Range and range-rate were collected by instrumented forward viewing radar from the following vehicle. Speed was collected from the speedometer. Yaw angle and lane offset were extracted from video recording. Acceleration from the accelerometer was used as longitudinal traffic action, and yaw angle was used as lateral action.

Safety Critical Events Extraction

The safety critical events were identified and analyzed in previous work by VTTI (Olson 2009). The method used to identify the safety critical events were triggers or thresholds on individual variables that were collected. For an event to be flagged, only one of the triggers had to be met. Those triggers are as follows:

- Longitudinal acceleration greater than or equal to -0.2 *g*.
- Forward time-to-collision of less than or equal to 2 s.
- Swerve greater than or equal to 2 rad/s^2.
- Lane tracker status equals abort (lane deviation).
- Critical incident button press.
- Analyst-identified event.

The safety critical events selected in this study are rear-end crash and near-crash conflict, especially when a following vehicle reacts to sudden braking or a sudden merge of the leading vehicle from the adjacent lane. When an event happens, the driver of the following vehicle brakes or swerves to avoid the incoming conflict.

State and Action Variables Selection

Before training, the training parameters (upper and lower bounds of fuzzy sets and discrete action set values) were determined from the naturalistic data. In this experiment, we used minimum and maximum state variables as lower and upper bounds.

Because discrete action sets should represent naturalistic driving actions, one basic method of selecting state and action variables is to extract representative actions from the naturalistic driving data. Statistically, the five representative action values of accelerations and yaws can be the minimum, the lower quartile (25^{th} percentile, cuts off the lowest 25 percent of data), the median (50^{th} percentile, cuts off 50 percent of data), the upper quartile (75^{th} percentile, cuts off the highest 25 percent of data), and the maximum. Notice that the agent output action is the weighted average of discrete actions. When using minimum and maximum discrete actions, the weighted average will make it impossible for the agent action to reach a naturalistic maximum and minimum. Therefore, the maximum value is multiplied by 1.2 when the maximum is positive and by 0.9 when the maximum is negative. The minimum value is multiplied by 0.9 when the minimum is positive and by 1.2 when the minimum is negative. This change extends the range of agent action.

Training Process

In our designed training process, at one time step of one event, fuzzy rules scan their associated weights, select the optimal actions, and update the weights. NFACRL continues updating the weights from the beginning of the events until the end. Then, NFACRL loads initial state variables from the same events. Theoretically, when the differences between critic and actor weights for two consecutive iterations become very small, the training process is considered to be finished. However, the proposed NFACRL is a heuristic methodology, so no global optimal agent behavior is guaranteed. Therefore, it is possible that the convergence of the weights may be premature and that NFACRL results in a local optimal solution. One way to avoid this premature convergence is to give the agent sufficient training iterations. In this event example, we tried 400 iterations in agent training. Because an average safety critical event has about 1,500 timing steps (150 s), driver fuzzy rules have been trained 600,000 times (1,500*400), and the resulting agent should produce a near-optimal approximation of naturalistic driver behavior.

During the learning process, a memory discount factor γ, a learning factor β, and a reward function scaling factor α affect the learning speed of NFACRL. γ controls the memory fade speed, in which the value of recently occurring states are weighted more. β shows how fast the

agent adjusts to the new information. α controls the magnitude of the reward function and weight change, and e_{th} controls the sign of the reward function. In this test, $\beta = 0.6$, $\gamma = 0.9$, $\alpha = 10$, and $e_{th} = 0.2$.

Based on our preliminary efforts, 10 truck drivers and 10 car drivers were selected. We used all the safety critical events available from database. Therefore, training parameters for each driver should be global, especially fuzzy thresholds of the state variables and discrete actions in the two action sets A and Y. By using all safety critical events, we attempted to enrich agent experiences and teach agents to react as drivers did. In addition, a greater number of events may prevent bias effects. Although conditions and causalities of different events can be totally different, NFACRL, theoretically, should still capture rules of individual drivers quite well because different events are located in different state spaces dominated by different fuzzy rules.

AGENT TRAINING RESULTS

Training Results

As an example, Figure 87 and Figure 88 show the agent's longitudinal action (acceleration) and lateral action (yaw angle) during one event of Driver Agent A. The blue scatter plot represents the naturalistic driving actions, and the green curves show the agent actions. NFACRL captures naturalistic driver behavior quite well in this event, with an R squared degrees of accuracy of 0.981 and 0.967 for acceleration and yaw angle, respectively. We also tested NFACRL performance on another event of the same driver (see Figure 89 and Figure 90).

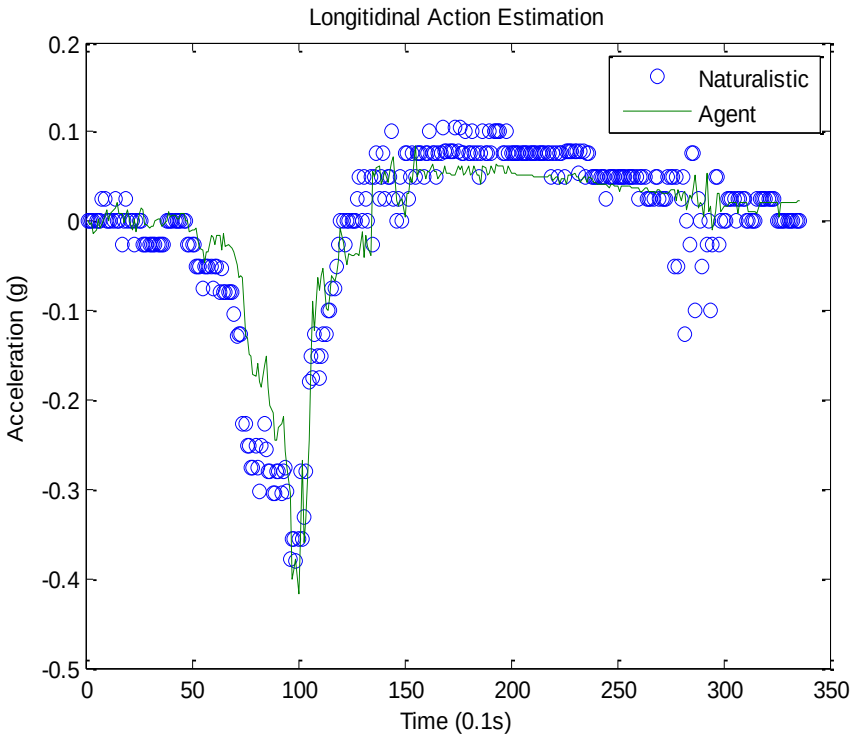

Figure 87. Graph. Acceleration of Agent A, event A1.

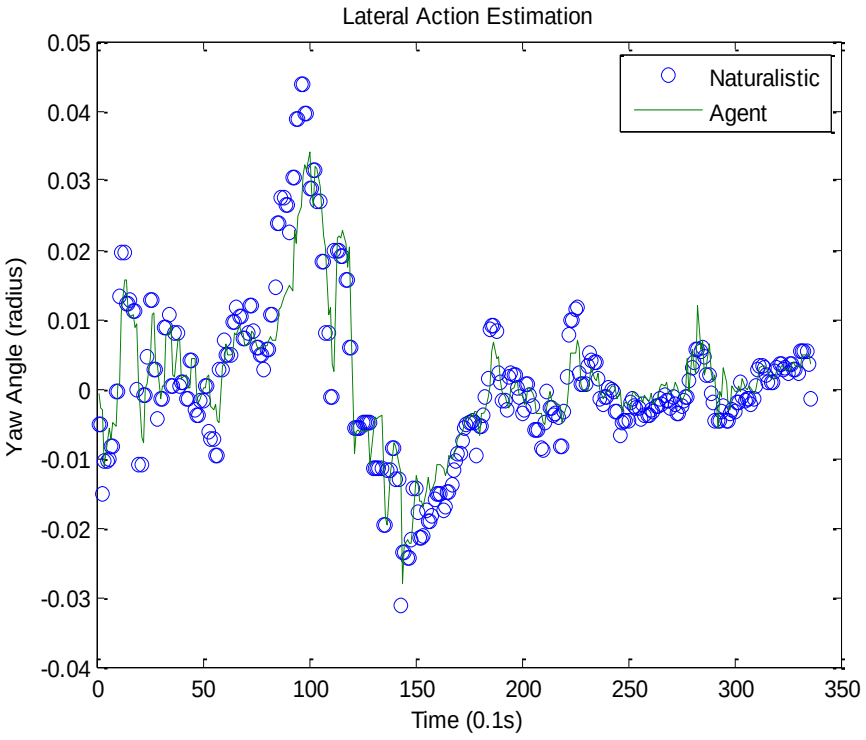

Figure 88. Graph. Yaw angle of Agent A, event A1.

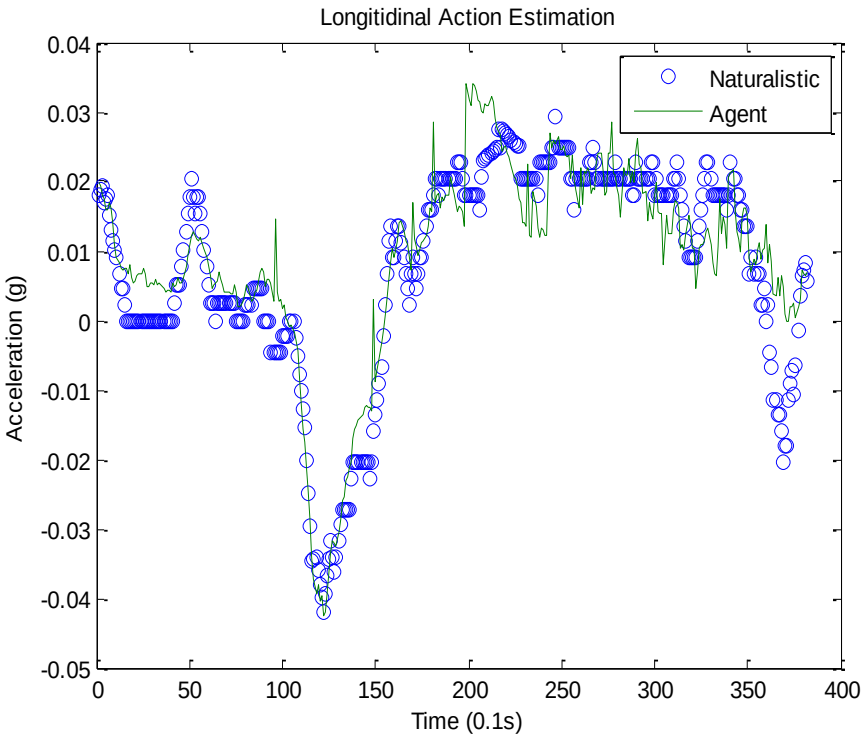

Figure 89. Graph. Acceleration of Agent A, event A2.

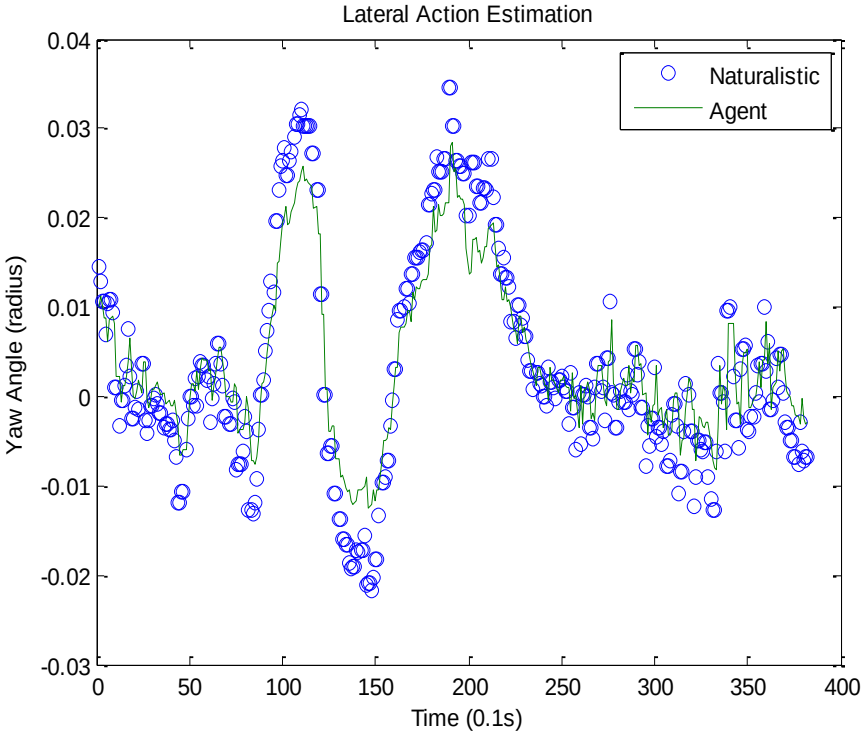

Figure 90. Graph. Yaw angle of Agent A, event A2.

Agent A captures Driver A's naturalistic behavior during events A1 and A2 quite well. It seems that even if events A1 and A2 vary, Driver A behaves similarly. This may prove that NFACRL

methodology will not result in an average driving behavior, even when using more events as in training.

The four figures above show that during some part of the events, such as $t = 28s$ to $30s$ in Figure 87, the action of Agent A diverges from naturalistic data by a small amount. Several reasons, including data collection errors in traffic state variables and unstable driver behavior within the events, may explain these differences. Occasionally, the leading vehicle does not fall into the range of the radar detection zone, so the agent assumes there is no vehicle in front of it. Consequently, the wrong traffic state leads to the wrong action. Human behavior is another phenomenon that is difficult to model. Within one event, a driver may react differently in similar traffic states. Also, the training factors may contribute to the error. From the reward functions, shown in Figure 83 and Figure 84, we tolerate NFACRL errors within a smaller amount of e_{th}, which may result in deviations from naturalistic actions.

CROSS VALIDATION

Cross Validation Results

The purpose of cross validation is to show the different behavior of different drivers when they experience the same hypothetical event. Driving behavior of one driver (for example, Driver A) must be obtained by using his or her events in training, and then states from another event (for example, event B) experienced by another driver (Driver B) should be used to demonstrate Driver A's actions during event B.

In our test, we trained Agent A and simulated Driver A's actions but using the event states from Driver B. Similarly, we simulated Agent B's actions using Driver A's events. Figure 91 and Figure 92 show an event example (event B1) from Agent B trained by using all its safety critical events.

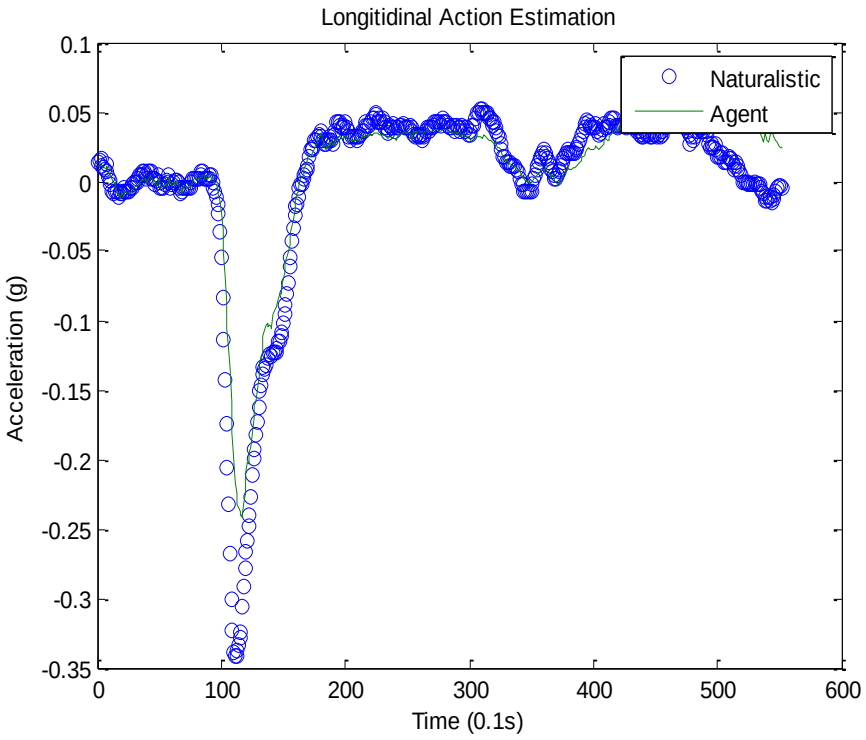

Figure 91. Graph. Acceleration of Agent B, event B1.

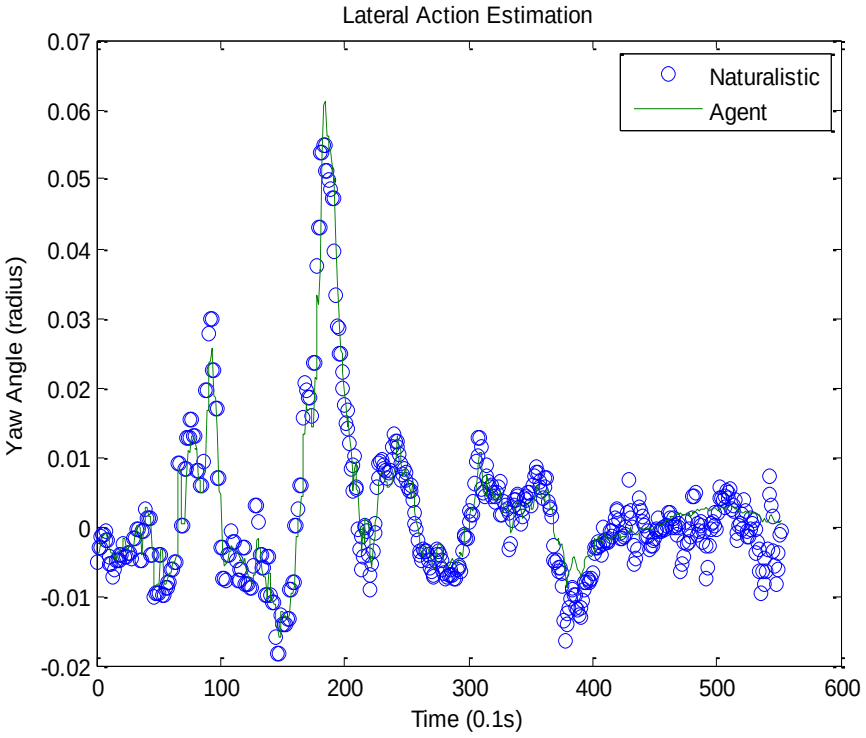

Figure 92. Graph. Yaw angle of Agent B, event B1.

Figure 93 and Figure 94 show the longitudinal and lateral actions of Agent B in one event from Driver A. Figure 95 and Figure 96 show the longitudinal and lateral actions of Agent A in one event from Driver B. It is very clear that Driver A and Driver B perform heterogeneous behaviors.

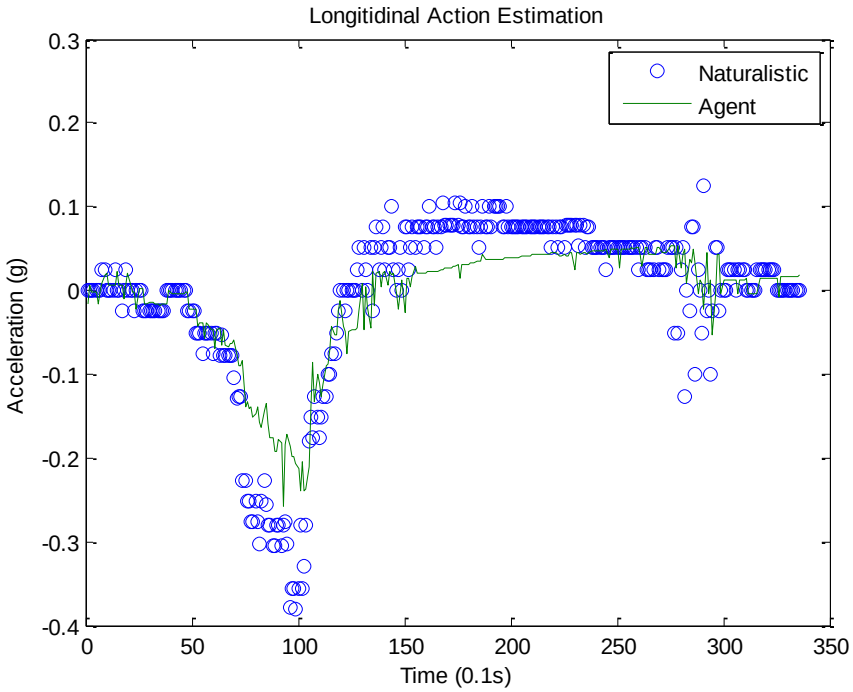

Figure 93. Graph. Acceleration of Agent B, event A1.

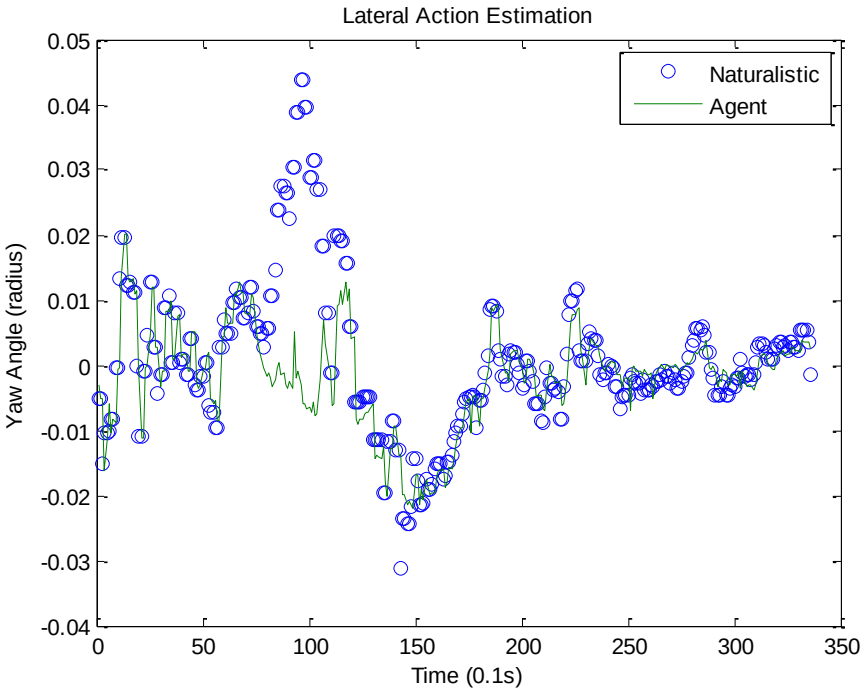

Figure 94. Graph. Yaw angle of Agent B, event A1.

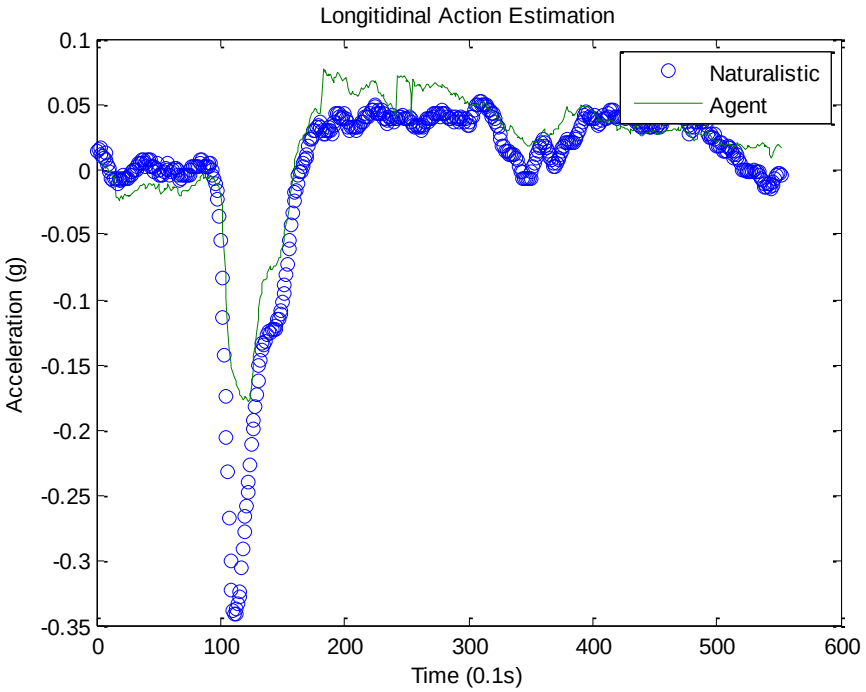

Figure 95. Graph. Acceleration of Agent A, event B1.

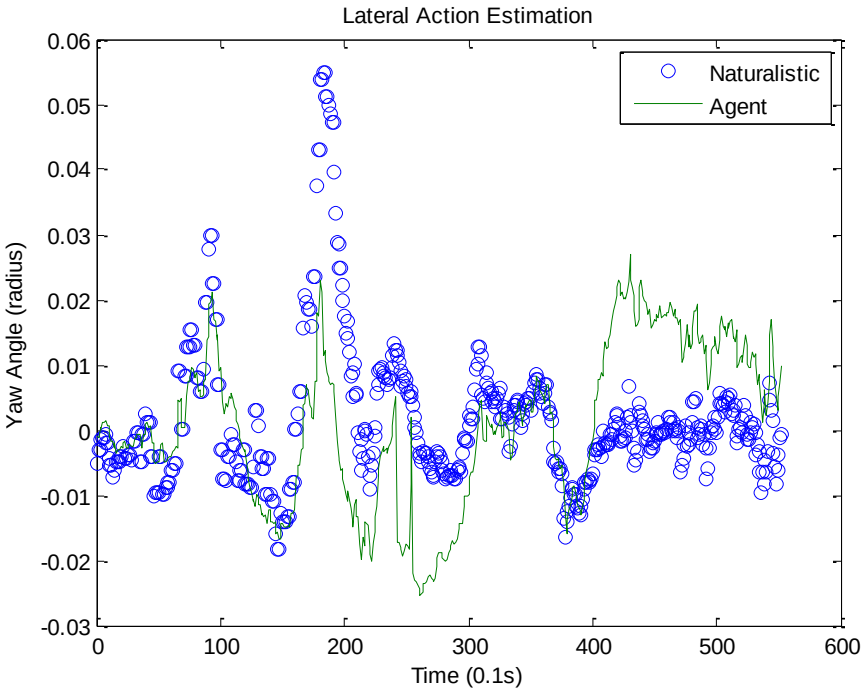

Figure 96. Graph. Acceleration of Agent A, event B1.

Table 24 uses R squared values as a statistical representation of the degree of accuracy of agent performance. In the table, the intersection of Driver A and Agent A and the intersection of Driver B and Agent B represent the degree of approximation when using events of the same driver in training and validation. The table shows that the degree of approximation when using events of

the same driver in training and validation is much greater than the degree of approximation in the cross validation (upper right and lower left), which also shows clear heterogeneities between the two drivers.

Table 24. R squared values for cross validation.

Event	Agent A		Agent B	
	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.98	0.97	0.81	0.83
Driver B	0.82	0.60	0.97	0.92

The Mega-Agent Concept

We attempted to design an imaginary agent that can capture the behaviors of both Driver A and Driver B but would not result in an average behavior between A and B. Our revised NFACRL methodology may handle this challenge. According to the nature of traffic state variables, the state space in this problem has six dimensions. Differences in one dimension (lower speed and high speed, for example) can make two separate states in the state space. From the naturalistic data, as we found out, state variables from different events vary greatly, so it is less likely that two events would have many overlaps in the state space. Notice that in NFACRL, different fuzzy rules dominate different regimes of the state space divided by fuzzy sets of state variables. Therefore, the training process of the hypothetical mega-agent involves adjusting the fuzzy rules that are taking charge of relevant state-space regimes in which safety critical events are located. Similarly, the fuzzy set thresholds and discrete action sets of training parameters are adjusted.

We used all the events of Driver A and Driver B to train the mega-agent. Performances of the mega-agent (using events A1 and B1) are presented in Figure 97, Figure 98, Figure 99, and Figure 100.

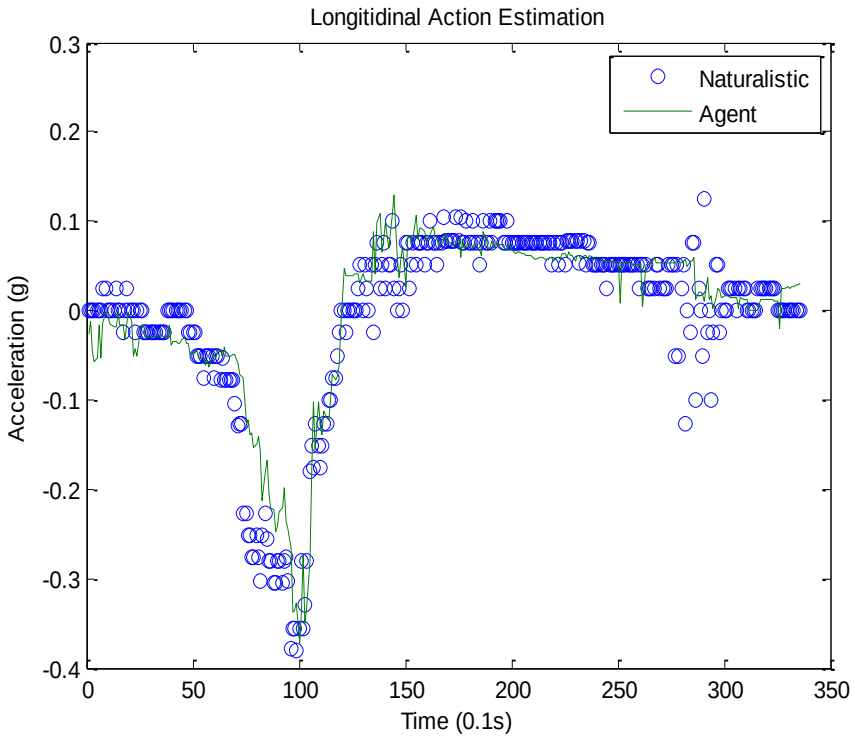

Figure 97. Graph. Acceleration of mega-agent, event A1.

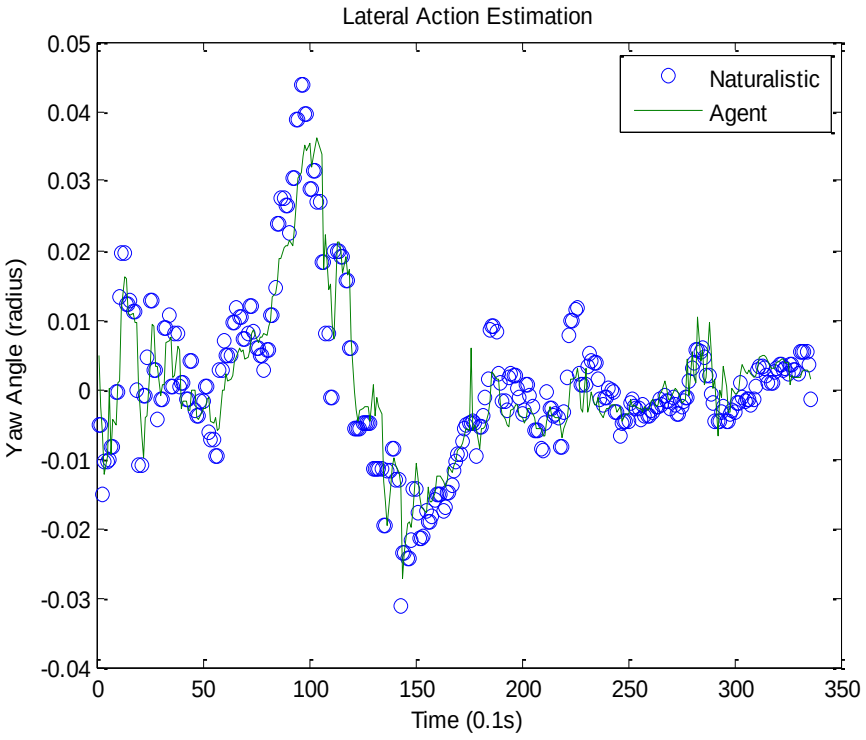

Figure 98. Graph. Yaw angle of mega-agent, event A1.

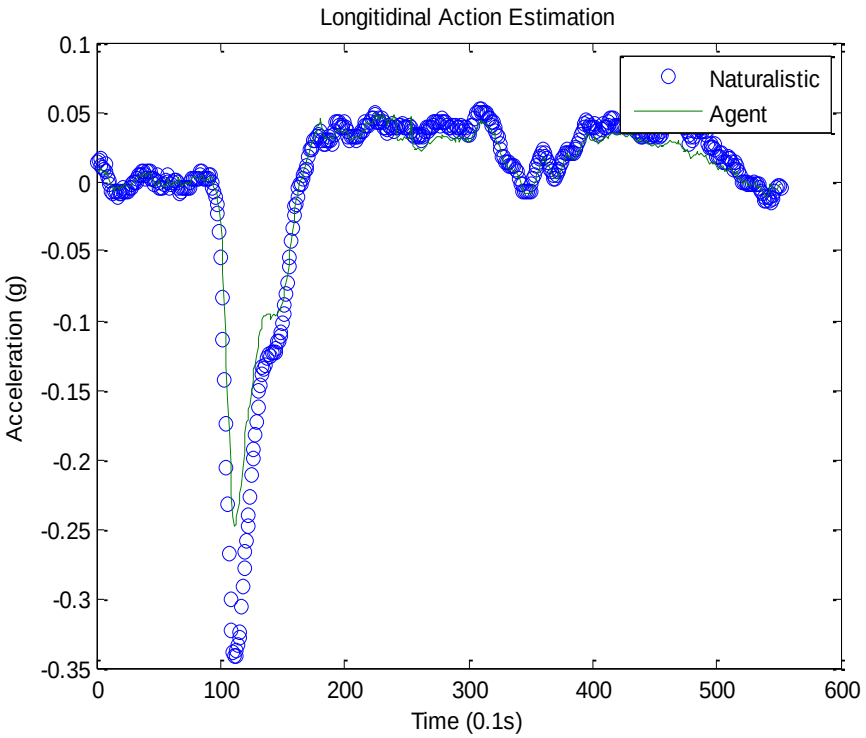

Figure 99. Graph. Acceleration of mega-agent, event B1.

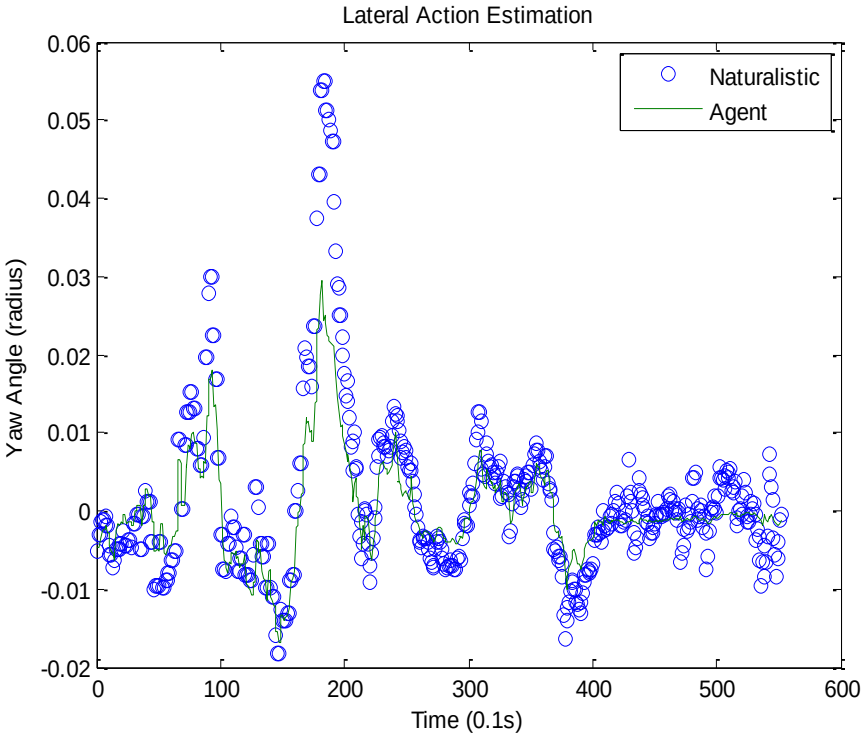

Figure 100. Graph. Yaw angle of mega-agent, event B1.

Compared to Figure 93, Figure 94, Figure 95, and Figure 96, the mega-agent performs better in cross-validation and can differentiate between Agent A and Agent B most of the time.

In reality, a conservative driver might never experience safety critical events, but an aggressive driver may experience many. In that case, the conservative driver has no idea what actions to take to get out of a sudden emergency regime. Through this mega-agent training, the conservative driver will learn the crash avoidance actions from the aggressive driver and be better equipped to evade upcoming crashes.

Table 25 shows the R squared values of the mega-agent. They are quite high, although they are slightly lower than those when training and validation use the same events. In fact, the mega-agent is capable of mimicking the behaviors of Driver A and B at the same time without losing much driver specificity.

Table 25. R squared values of the test mega-agent.

Event	**Agent A**		**Agent B**		**Mega-Agent**	
	Acceleration	Yaw	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.98	0.97	0.81	0.83	0.98	0.95
Driver B	0.82	0.60	0.97	0.92	0.97	0.91

From our preliminary efforts, we designed two mega-agents: one truck mega-agent and one car mega-agent. We selected safety critical events from 10 truck drivers and 10 cars drivers to train the two agents separately. Each of them used all the event data from their 10 selected drivers in training. In fact, each of the two agents should know all the evasive behaviors of their 10 guiding drivers.

Table 26. R squared values of the truck mega-agentTable 26 shows the performance of the truck mega-agent. Two first two columns (under “Agent Itself”) show the optimal performance when training data and validation events come from the same driver. As expected, compared to the agent itself, the performance of the mega-agent is approximately the same, although it is slightly worse in some of the cases because of the noise.

Table 26. R squared values of the truck mega-agent.

Event	Agent Itself		Truck Mega-Agent		Cross Validation with Agent A	
	Acceleration	Yaw	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.97	0.97	0.98	0.97	0.98	0.97
Driver B	0.97	0.94	0.97	0.91	0.82	0.60
Driver C	0.98	0.96	0.97	0.96	0.93	0.86
Driver D	0.99	0.92	0.99	0.88	0.86	0.64
Driver E	0.88	0.96	0.81	0.95	0.47	0.76
Driver F	0.98	0.96	0.94	0.96	0.83	0.43
Driver G	0.86	0.98	0.84	0.88	0.86	0.62
Driver H	0.96	0.99	0.95	0.99	0.63	0.98
Driver I	0.95	0.98	0.93	0.98	0.48	0.75
Driver J	0.85	0.98	0.85	0.97	0.66	0.32

Similarly, the degree of accuracy of the car mega-agent is presented in Table 27.

Table 27. R squared valued of the car mega-agent.

Event	Agent Itself		Car Mega-Agent		Cross Validation with Agent D	
	Acceleration	Yaw	Acceleration	Yaw	Acceleration	Yaw
Driver A	0.94	0.94	0.93	0.70	0.90	0.50
Driver B	0.97	0.94	0.96	0.79	0.82	0.30
Driver C	0.93	0.97	0.97	0.83	0.93	0.77
Driver D	0.97	0.93	0.96	0.93	0.97	0.93
Driver E	0.95	0.98	0.91	0.98	0.68	0.95
Driver F	0.97	0.92	0.96	0.89	0.92	0.86
Driver G	0.98	0.93	0.99	0.88	0.98	0.79
Driver H	0.98	0.93	0.99	0.89	0.98	0.90
Driver I	0.97	0.95	0.98	0.91	0.90	0.85
Driver J	0.95	0.92	0.99	0.89	0.98	0.65

CHAPTER CONCLUSION

This chapter reported a framework that simulates individual driver actions during safety critical events. Neuro-fuzzy actor-critic reinforcement learning (NFACRL), an agent-based artificial intelligence machine-learning technique, was used to model driving behavior. The naturalistic driving database was used to train and validate driver agents.

The advantages of NFACRL lie mainly in its ability to simulate heterogeneous vehicle actions in complicated traffic environments. It is worth mentioning that this research was an attempt to apply reinforcement learning techniques to solving high-dimensional state problems and continuous-action simulation problems simultaneously in transportation research, especially in traffic flow theory. From the perspective of microscopic traffic behavior modeling, the proposed methodology is able to simulate lateral action, which brings new insights to the modeling of driver maneuvering behavior during safety critical events. The proposed methodology also simulated events from different drivers and proved behavior heterogeneities.

This task focused on safety critical events. The next step of this research is the extension of NFACRL to simulate other traffic behavior, such as lane-changing behavior, merging behavior in the upstream and downstream of ramps, and evacuation behavior. It would be interesting to model individual driver behavior and the decision-making process under these traffic conditions.

For the technical part of this task, the training parameters speed, memory, and scaling factor were fixed for all the events, but it would be interesting to see how NFACRL performs under different combination of these factor sets. As with most artificial intelligence methods, NFACRL is a heuristic approach, so there is no guarantee that training results would be optimal. Therefore, training parameters play an important role in guiding NFACRL to find a near optimal result. During our test, we found that agent performance is very sensitive to the driver-dependent training parameters, discrete action sets, and state bounds. Theoretically, it would be better if these parameters were preoptimized before training. However, because the only way to test agent performance is through NFACRL training, parameter optimization and training forms a cycle, and computation time would exponentially increase. Our approach set the parameters based on statistical quartiles, and this yielded good results. Future research will focus on the optimization methodology for training parameters and will improve agent performance.

DISCRIMINANT ANALYSIS

Discriminant analysis is a statistical method that uses information known a priori about the classification of observations from different datasets, to distinguish between future observations from the same datasets (in this case two datasets: safety-critical and normal). The technique works by finding coefficients for the input variables that when multiplied by the variables and summed, create a value that can be reliably used to classify future variables. In this chapter, we use the discriminant analysis technique to distinguish between safe (normal) and unsafe (safety-critical) driving patterns before safety-critical event occurs. This is a concept that we present here as a potential trigger for agent activation. In other words, if agent-based simulation were to be activated only to handle the simulation of safety-critical events, then the agents should be activated just before a safety-critical event is expected. The trigger-based activation could also be set with values that can ensure the activation of agents during the whole simulation period if it was so desired.

For this analysis, 30 data points from different car-following periods were used for each driver, along with the event data for that driver. This number of car-following or normal-driving points was used to produce a fair representation of normal driving behavior, while not overpowering the safety critical event data in the analysis. Six variables were used for the discriminant analysis: longitudinal acceleration, vehicle speed, yaw angle, lane offset, range, and range rate. Figure 101 describes the mathematical form of the resulting discriminant score and how it relates to the coefficients for each variable.

$$Discriminant\ Score = \sum \beta_i * X_i$$

Figure 101. Equation. Discriminant score.

where β_i is the coefficient for variable i and X_i is variable i.

Car Drivers

Table 28 presents the coefficient values resulting from a discriminant analysis at different time steps before the occurrence of a safety critical event. The table shows that consistency exists in some of the coefficients over time, while the rest of the coefficients appear to be time dependent. Figure 102 shows the percentage of misclassification error at each time step. The time step at 6 s is chosen because it is the greatest amount of time that still maintains relatively low error. Figure 103 shows the results of applying the coefficients of the different time steps to the near-crash data as well as the normal car-following data. The results show some separation between the datasets at each time step, with some overlap occurring. Figure 2 shows that the error weighs on the side of misclassifying safety critical events as normal driving behavior, which is not an acceptable error. Time step 6 shows a clear separation between the two datasets, which means that the coefficients belonging to that time step would be the optimal choice for use in further study.

Table 28. Discriminant coefficients for Driver l.

Time	Longitudinal Acceleration	Lateral Acceleration	Speed	Lane Offset	Yaw Angle	Range	Range Rate
0	5.4	2.7	0.051	0.019	-0.4	0.013	0.12
1	1.5	-13.1	0.013	0.019	1.6	0.013	0.15
2	-10.3	-10.6	0.030	0.022	-5.4	0.030	0.18
3	-4.0	-5.9	0.057	0.017	2.1	0.024	0.19
4	1.8	-8.7	0.053	0.014	15.3	0.000	0.13
5	-17.9	14.2	-0.005	-0.003	-16.3	-0.004	-0.15
6	16.5	-28.1	-0.026	0.014	38.0	0.001	0.15
7	9.8	-12.2	0.006	0.009	23.2	0.014	0.16
8	22.0	-17.5	-0.009	0.016	37.5	0.008	0.20
9	-14.8	15.3	0.006	-0.007	-3.0	-0.004	-0.15
10	14.3	-12.3	-0.004	0.014	-3.5	0.007	0.14

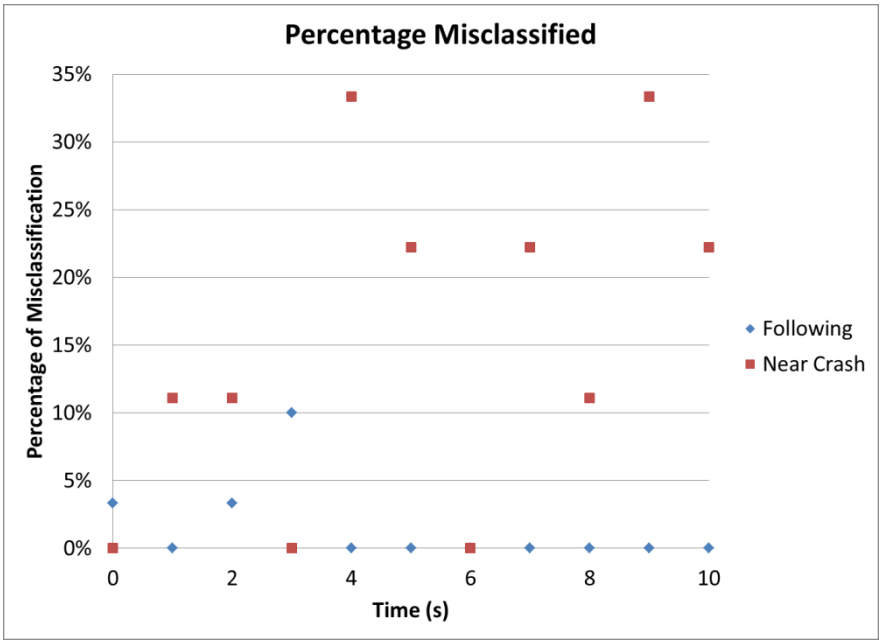

Figure 102. Graph. Percentage of misclassified data points for Driver l.

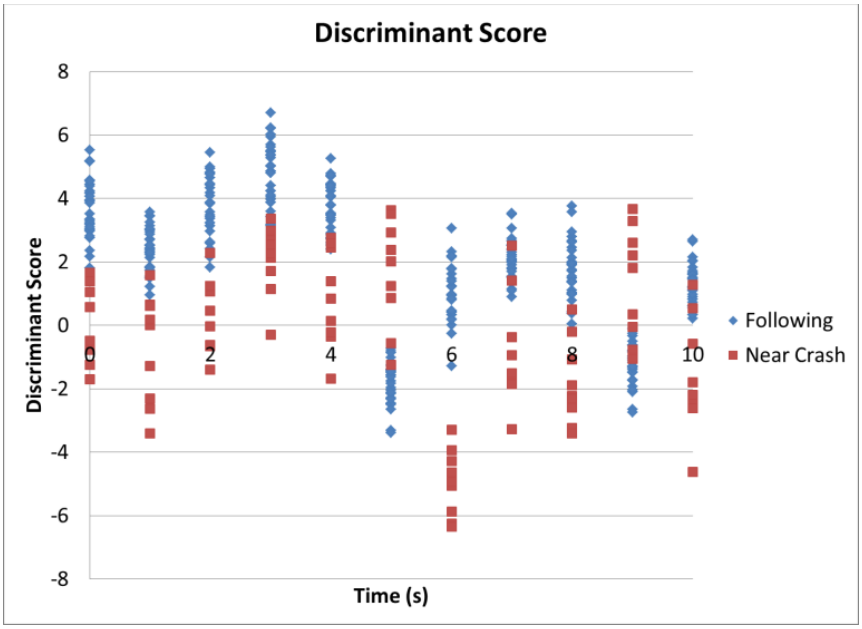

Figure 103. Graph. Discriminant scores for Driver l.

Figure 104, Figure 105, and Figure 106 show the graphs of some of the terms that constitute the discriminant score for Driver l. These graphs are the result of multiplying the input variables by the corresponding discriminant coefficient. The graphs show that the observable results of the discriminant analysis are not due to any one variable. Rather, the results are highly reliant on the specific combination of all of the variables that the discriminant coefficients create.

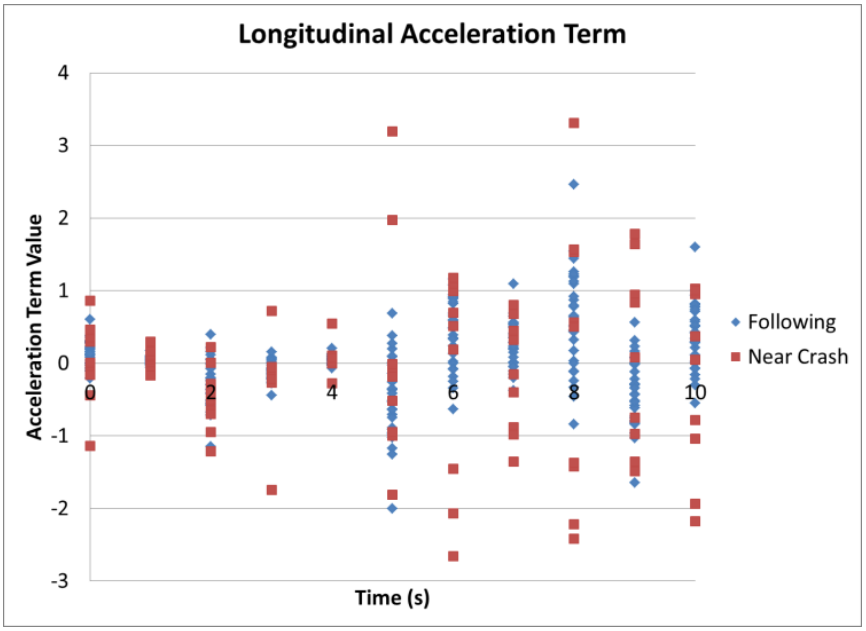

Figure 104. Graph. Longitudinal acceleration term for Driver 1.

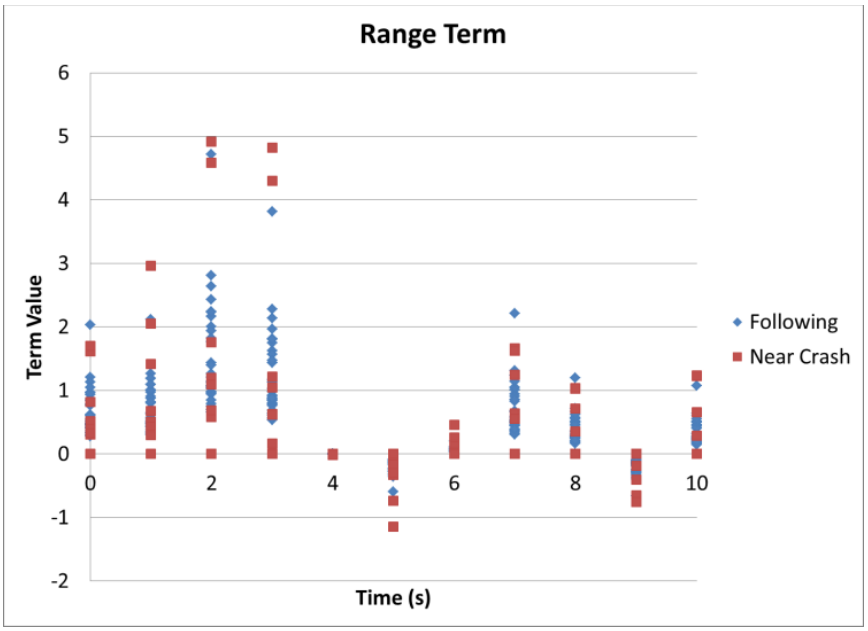

Figure 105. Graph. Range term for Driver 1.

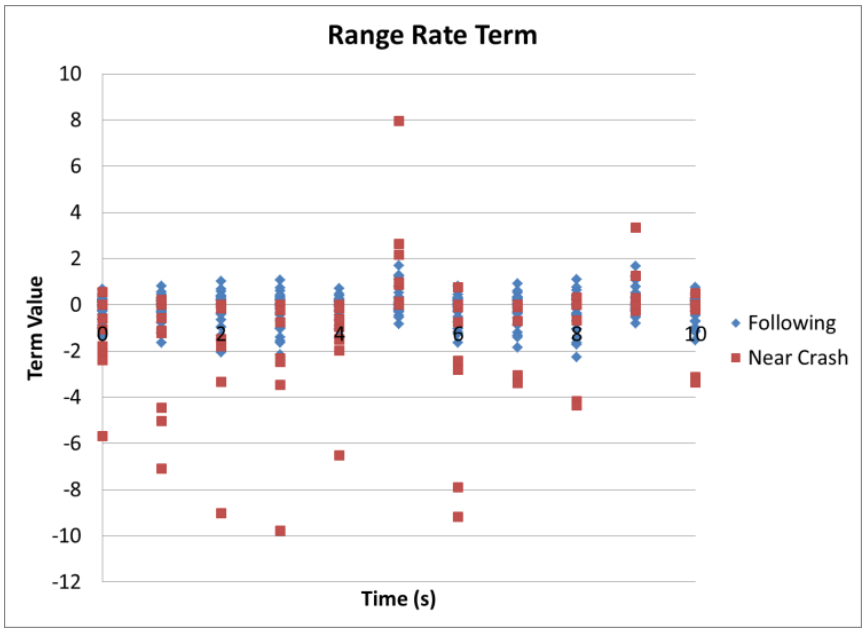

Figure 106. Graph. Range rate term for Driver 1.

Figure 107 shows the resulting discriminant scores when the coefficients from the 6-s time step are applied to a full car-following period. The figure shows some erratic behavior, but the discriminant scores are still above −3, which is the highest point of the near-crash data inFigure 107. Figure 108 shows the results of applying the same coefficients to a sample near-crash event. The discriminant scores seem similar to a normal car-following period until the score suddenly jumps below −5. This jump corresponds to a vehicle suddenly changing lanes to be in front of the subject vehicle.

Figure 109 shows the speed trajectory of the same near-crash event along with a marker of the 6-s prediction given by the discriminant coefficients. The 6-s prediction is well before the driver's reaction. Also, the discriminant score plateaus around −5 immediately after the 6-s prediction, which means that the driving conditions remain dangerous, but the driver has not adjusted to the change in conditions. When the driver decelerates, as seen in the trajectory, the discriminant score increases, indicating a change from dangerous conditions to safe conditions.

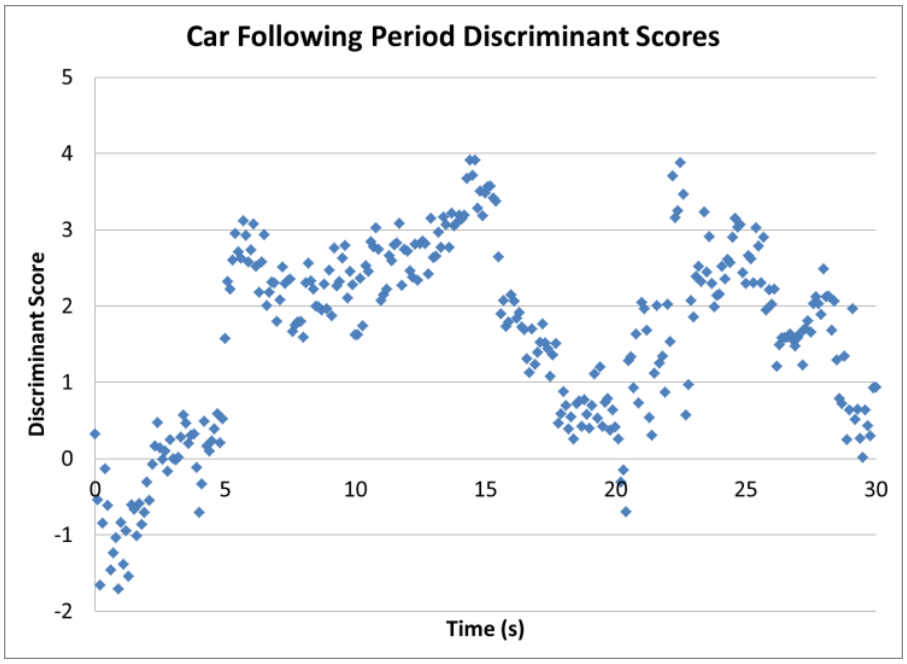

Figure 107. Graph. Discriminant scores of a sample car-following period.

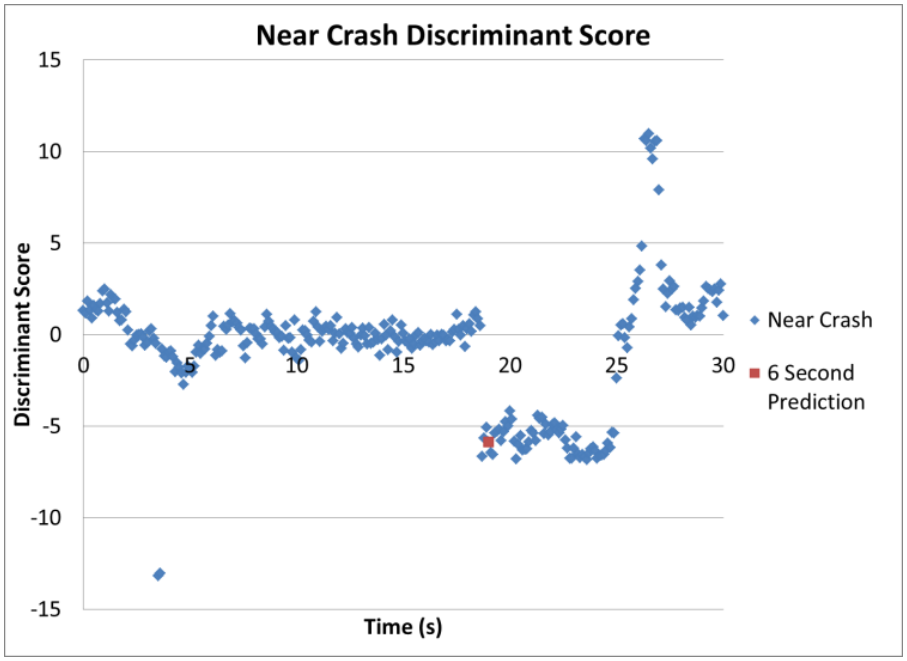

Figure 108. Graph. Discriminant scores of a sample near-crash event.

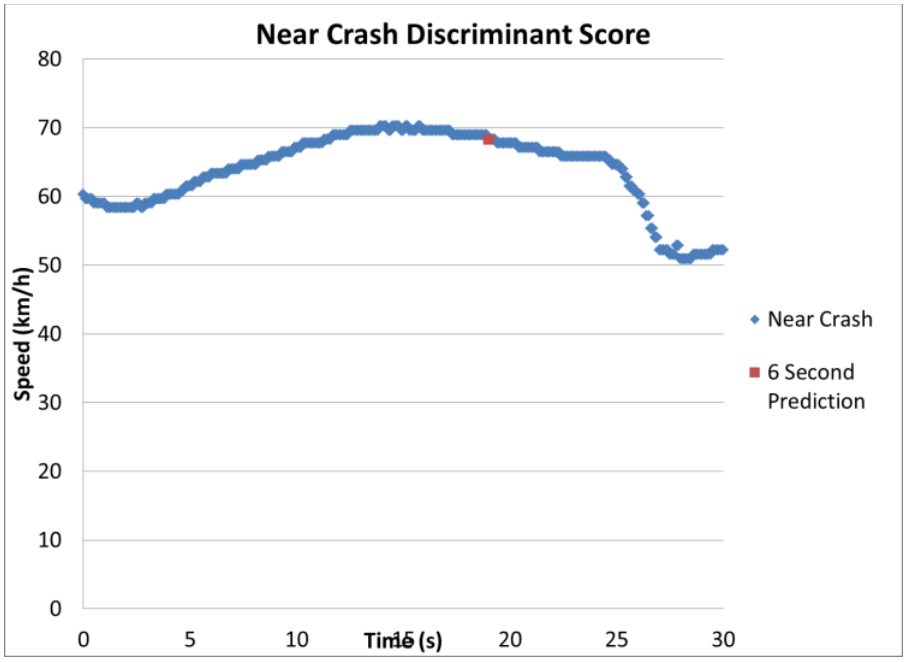

Figure 109. Graph. Trajectory of a sample near-crash event.

Table 29 shows the discriminant coefficients for Driver m. The acceleration coefficients show that the discriminant score for this driver relies on the longitudinal acceleration more than on the lateral acceleration. Driver 103-A showed more balance between the two accelerations, which shows that every person has unique behaviors. Figure 110 shows the percentage of misclassification associated with each set of discriminant coefficients.

Table 29. Discriminant coefficients of Driver m.

Time	Longitudinal Acceleration	Lateral Acceleration	Speed	Lane Offset	Yaw Angle	RANGE	Range Rate
0	-15.1	-2.3	0.64	-0.02	14.9	-0.02	0.21
1	12.1	2.8	0.39	0.04	-9.0	0.01	-0.10
2	-26.8	0.0	0.32	0.02	-13.3	-0.01	0.01
3	-32.3	-0.1	0.66	-0.01	2.6	-0.02	0.05
4	-28.7	0.9	0.33	0.00	4.2	-0.01	0.02
5	-30.6	0.2	0.31	0.00	-1.8	-0.02	0.04
6	-27.4	1.0	0.21	0.00	0.7	-0.01	0.03
7	-31.2	0.3	0.29	-0.01	5.8	-0.02	0.04
8	28.2	-0.7	-0.21	0.00	4.2	0.01	-0.01
9	-11.6	2.4	0.14	0.03	-21.4	0.00	-0.07
10	12.3	3.3	0.14	0.02	-14.3	0.02	-0.10
11	3.1	2.8	0.14	0.02	-19.3	0.02	-0.06
12	-10.3	3.3	0.14	0.01	-1.5	0.00	-0.01
13	-7.1	0.0	0.14	0.01	-5.7	0.01	0.05
14	-8.5	-3.4	0.14	0.00	-7.6	0.01	0.08
15	5.1	-1.7	0.15	0.02	-28.9	0.01	-0.02

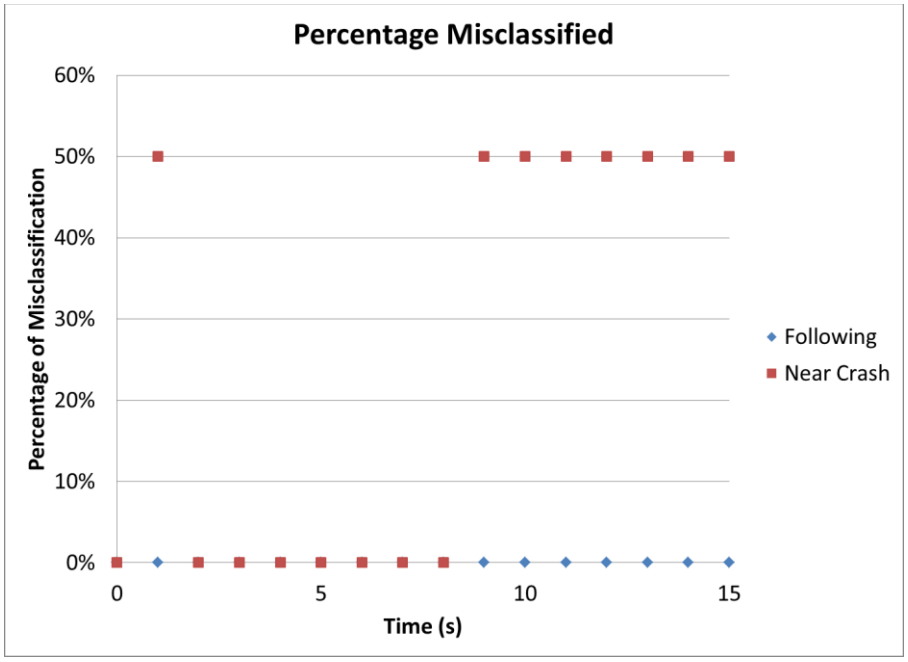

Figure 110. Graph. Percent misclassification for Driver m.

Figure 111 shows the discriminant scores for the near-crash data and normal car-following data of Driver m. These scores are calculated by multiplying the discriminant coefficients by the appropriate data values and then taking the summation. There is a separation between the normal car-following periods and the near-crash events in Driver m at most time steps until time step 9. There, the analysis begins to show a misclassification of one of the near-crash events as normal driving behavior. To avoid this error, the coefficients at time step 7 should be considered optimal.

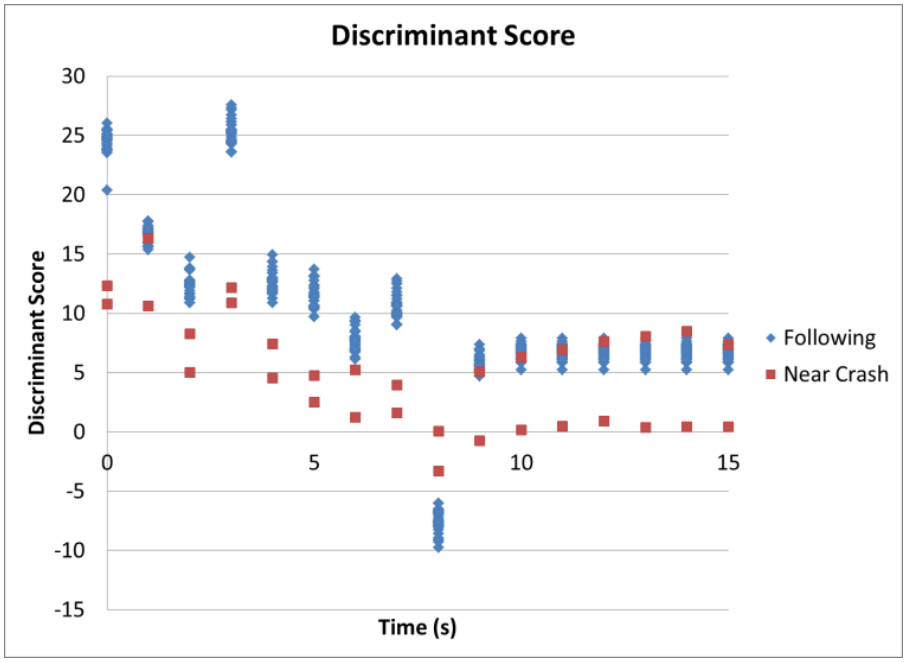

Figure 111. Graph. Discriminant scores for Driver m.

Figure 112, Figure 113, and Figure 114 show a further breakdown of the discriminant score for Driver m into three of the seven terms: longitudinal acceleration, range, and range rate. Just like Driver l, the graphs support the finding that the resulting discriminant score is dependent upon the specific combination of the input variables that the discriminant coefficients provide.

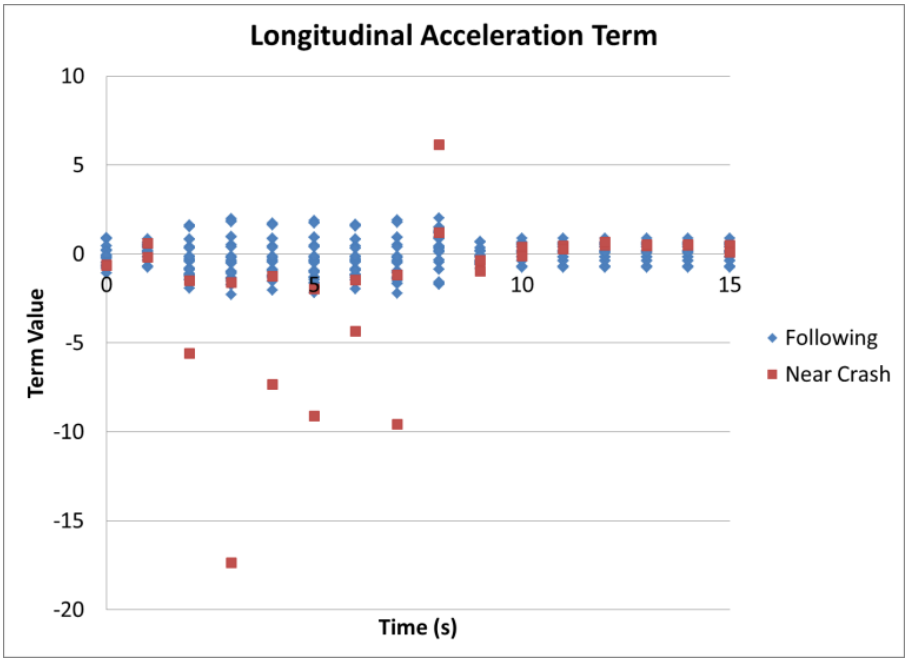

Figure 112. Graph. Longitudinal acceleration term for Driver m.

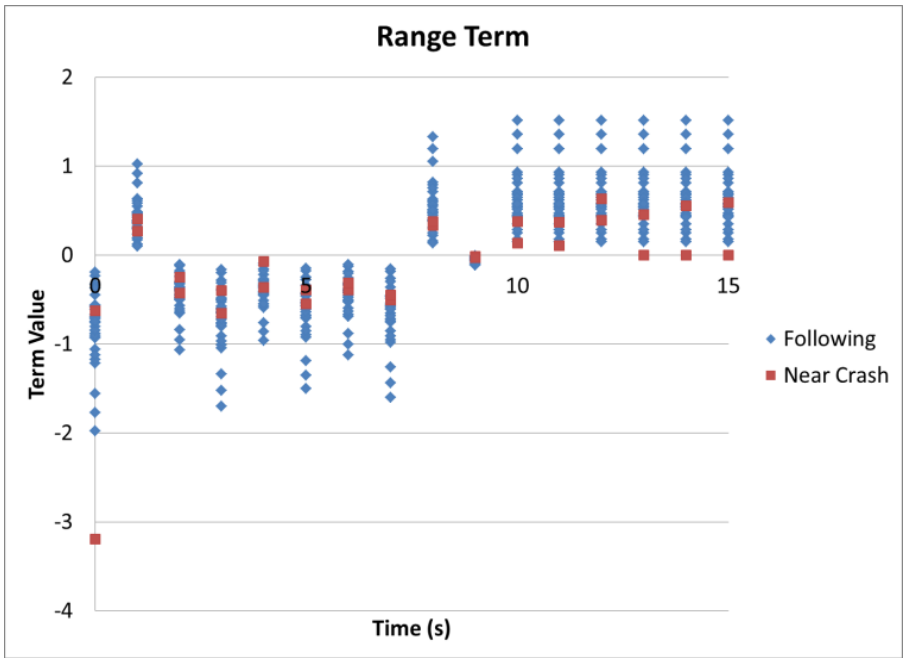

Figure 113. Graph. Range term for Driver m.

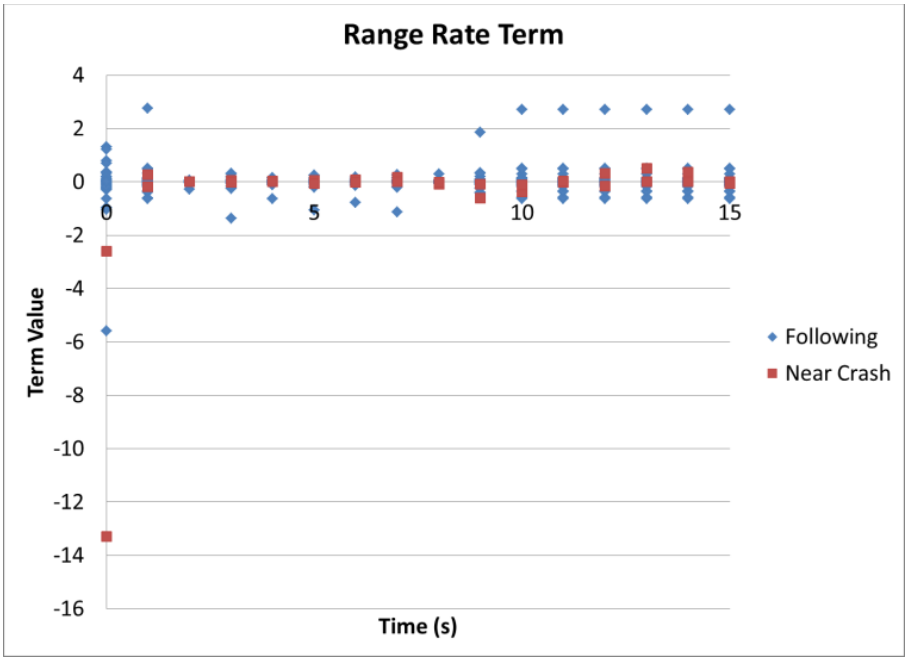

Figure 114. Graph. Range rate term for Driver m.

Figure 115 shows the resulting discriminant scores when the coefficients of the 7-s time step are applied to a full car-following period. Figure 116 shows the results of when the same coefficients are applied to the data from a near-crash event. In Figure 117 the scores remain between approximately 10 and 12, while in Figure 118 the scores decrease immediately before the event occurs. Also, the discriminant scores from 10 to 12 s in Figure 117 appear to be similar to the scores seen in the normal car-following period. This means that a deterioration of the discriminant score can be seen as the data before the near-crash event transitions from safe to unsafe. Figure 117 shows the trajectory during the near-crash event as the driver decelerates suddenly to avoid a collision. Figure 117 also shows that the 7-s prediction given by the discriminant scores appears well before the reaction of the driver.

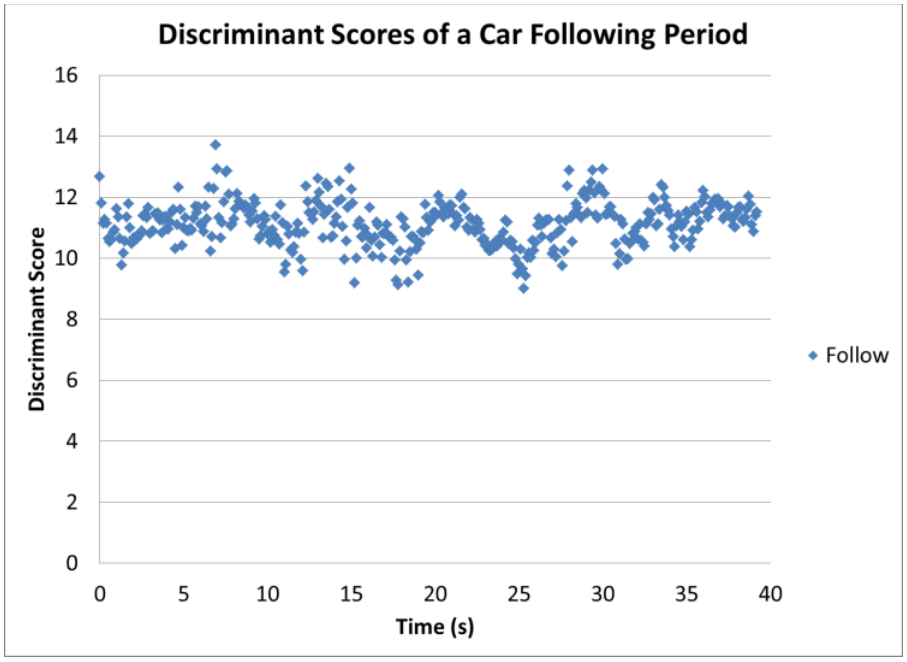

Figure 115. Graph. Discriminant scores of a sample car-following period.

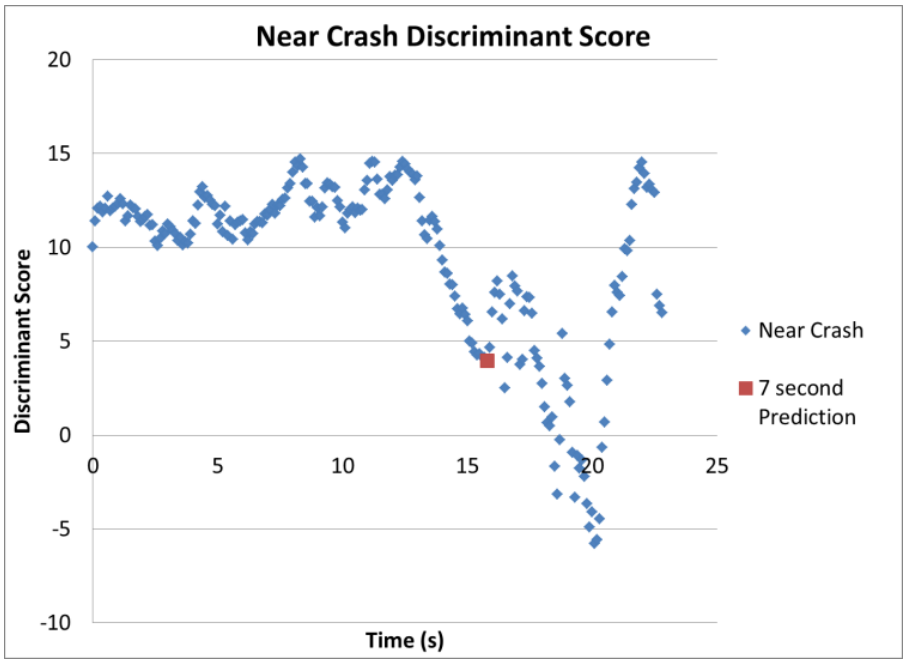

Figure 116. Graph. Discriminant scores of a sample near-crash event.

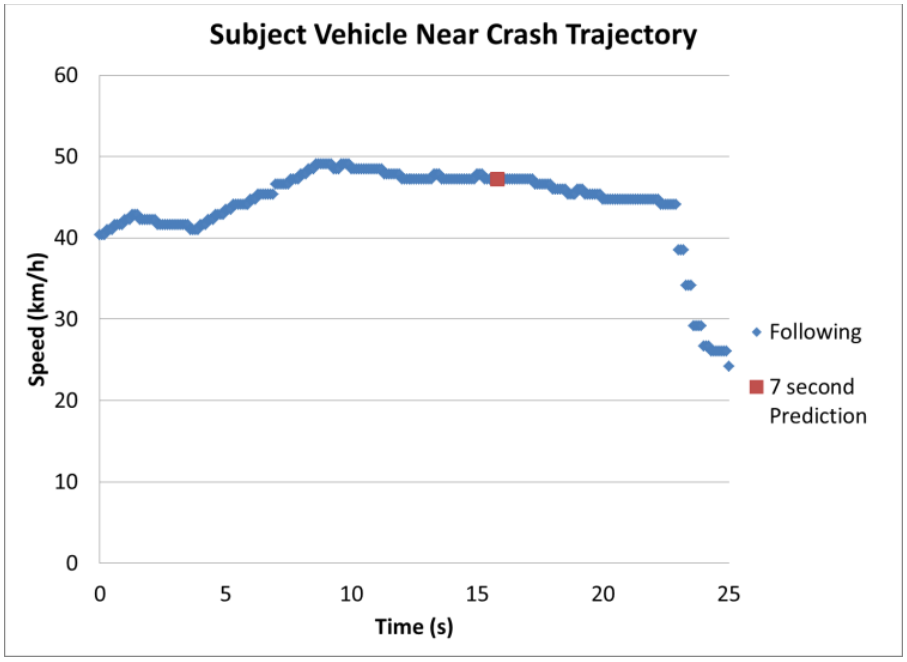

Figure 117. Graph. Trajectory of a sample near-crash event with the 7-s crash prediction point highlighted.

Figure 118 and Figure 119 show the results of attempting to use the coefficients of one driver for another driver. Both figures show that using a different set of parameters that was not optimized for that driver results in losing the clear distinction between normal car-following behavior and safety critical behavior. It is also interesting to note that in Figure 118 and Figure 119, the normal car-following discriminant scores are highly variable as opposed to the clustering seen in the results in Figure 118 and Figure 119.

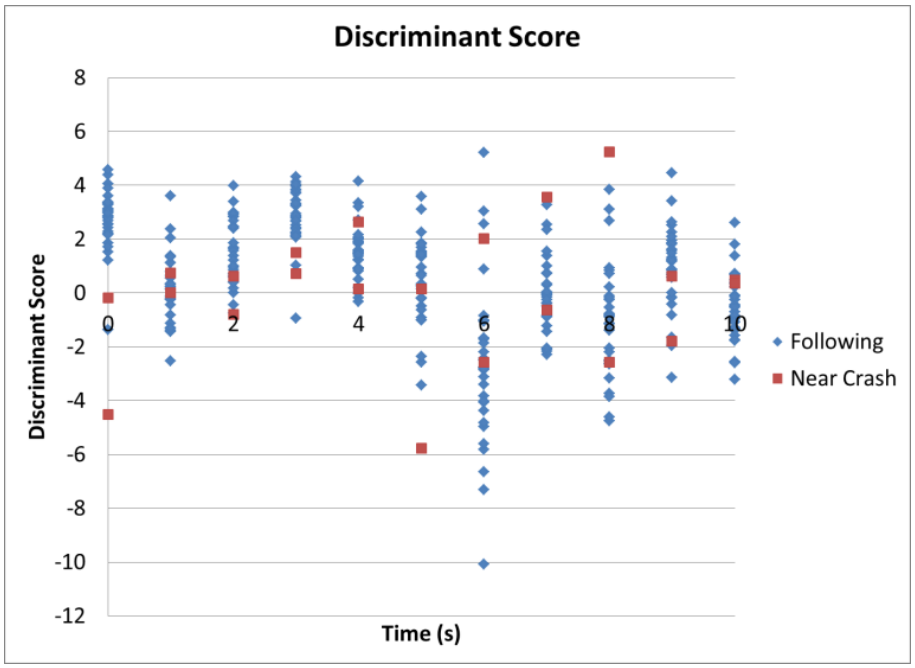

Figure 118. Graph. Discriminant score for using Driver 1's coefficients for Driver m.

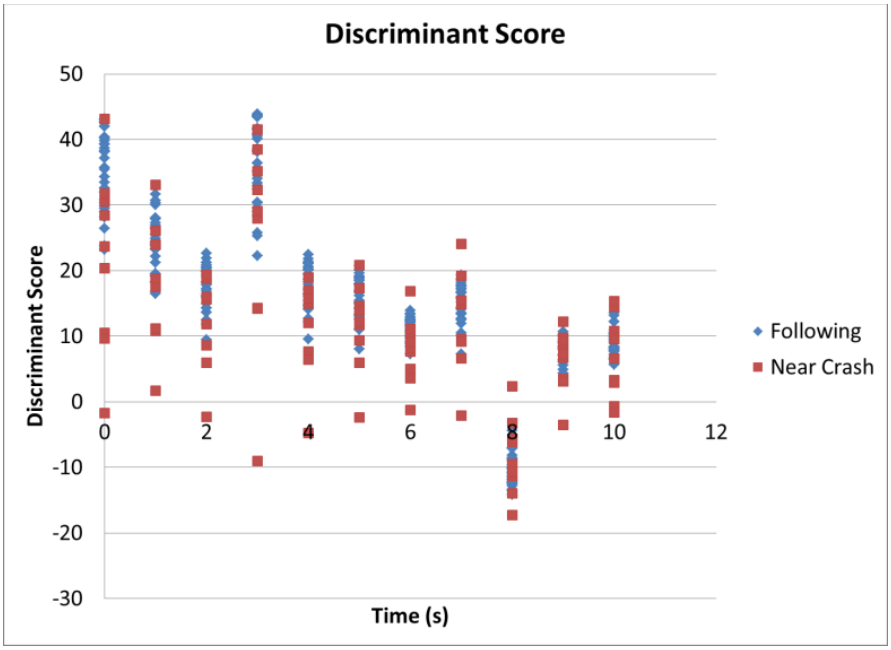

Figure 119. Graph. Discriminant scores for using Driver 1's coefficients for Driver m.

Table 30 presents the results of applying the procedure described for the two example drivers to eight more car drivers for a total of ten drivers. The results show that each driver has a specific combination of the six variables and a specific threshold value that can be used to distinguish between normal driving behavior and behavior in safety critical events. Also, the drivers' times vary, which means that the distinction between normal driving and safety critical behavior can be made at different times before the events for each driver. The largest time used in the analysis for some drivers was 6 s.

Table 30. Driver-specific discriminant coefficients and thresholds.

Driver	Time	Speed	Range	Range Rate	Acceleration	Yaw	Offset	Threshold
a	5.5	0.003	0.004	0.117	-13.221	74.676	-0.002	-4.525
b	6	0.013	-0.032	0.032	12.760	9.265	0.006	-7.458
c	6	0.153	-0.002	0.141	-8.460	-40.239	-0.051	0.538
d	1	0.027	0.013	0.190	-0.097	46.726	0.004	-2.728
e	6	0.087	0.005	0.243	5.184	-7.990	0.003	0.406
f	6	0.014	0.012	-0.093	-9.140	149.098	0.024	-4.317
g	6	0.056	0.003	-0.206	9.727	-73.285	0.035	-1.685
h	4.5	0.033	0.017	0.473	24.979	13.641	-0.008	0.719
i	6	0.170	-0.006	0.092	-32.342	0.000	0.227	-27.000
j	1.5	0.126	0.017	0.360	-1.852	75.317	-0.018	2.763
k	6	0.106	-0.001	-0.016	17.577	108.475	0.020	-0.504

Conclusions and Recommendations

The results support the hypothesis that using the seven selected variables along with the corresponding coefficients creates a discriminant score that can accurately distinguish between

safe and unsafe driving conditions. For Driver l, this distinction can be made 6 s before an event occurs.

Also, in the case of Driver m, this distinction can be made 7 s before an event occurs. The results of the discriminant analysis for Driver m arise from the specific combination of the six variables selected for analysis: longitudinal acceleration, vehicle speed, lane offset, yaw angle, range, and range rate. This means that the warning signs of an impending safety critical event are captured and represented by these six selected variables. These warning signs are subtle changes in the conditions that go unnoticed when looking at individual variables. But using the specific combination of the six variables, as defined by the discriminant analysis, results in a noticeable warning sign.

Truck Drivers

Table 31 presents the coefficient values resulting from a discriminant analysis at different time steps before the occurrence of a safety critical event for truck drivers. The table shows consistency in some of the coefficients over time, while the rest of the coefficients appear to be time dependent. Figure 120 shows the percentage of misclassification error at each time step. Figure 121 shows the results of applying the coefficients of the different time steps to both the near-crash and normal car-following data points. The results are clear: the near-crash data points are high negative values, and the car-following (safe) data points are positive values.

Table 31: Discriminant coefficients for Driver E.

Time	Longitudinal Acceleration	Lateral Acceleration	Speed	Lane Offset	Yaw Angle	Range	Range Rate
0	-6.8	13.9	0.001	0.10	-11.6	0.0003	0.12
0.5	-2.4	16.0	0.006	0.10	-13.8	0.0000	0.12
1	-0.5	15.1	0.010	0.09	-13.1	-0.0009	0.13
1.5	-3.6	10.8	0.003	0.07	40.3	-0.0016	0.13
2	0.3	9.7	0.010	0.05	36.4	-0.0028	0.14
2.5	-4.2	12.2	0.011	0.05	0.0	-0.0030	0.14
3	0.4	11.4	0.017	0.05	-4.0	-0.0038	0.14
3.5	1.2	10.6	0.018	0.04	9.3	-0.0043	0.14
4	9.6	22.0	0.008	0.11	16.1	0.0082	0.00

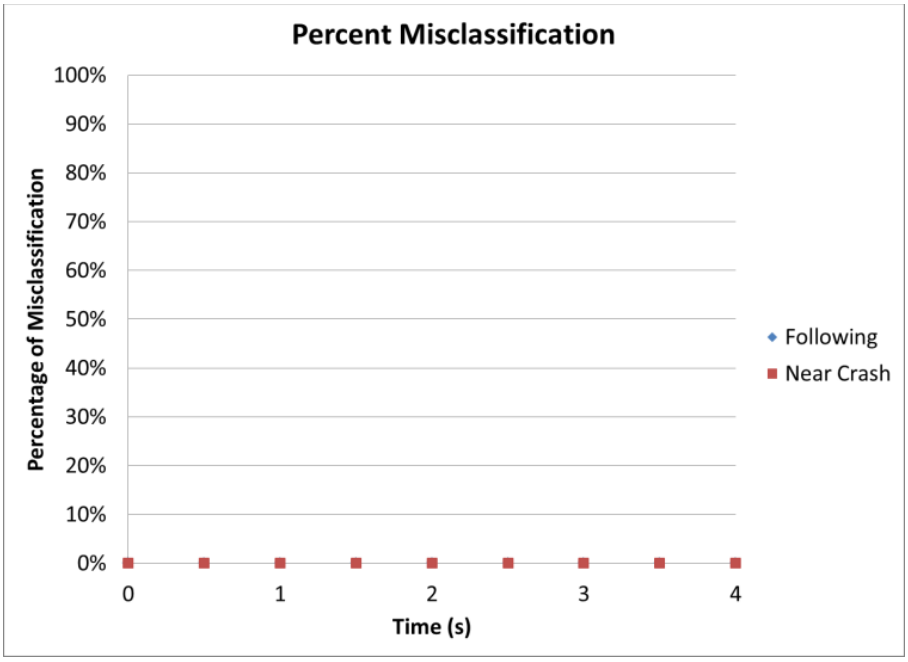

Figure 120. Graph. Percentage of misclassified data points for Driver E.

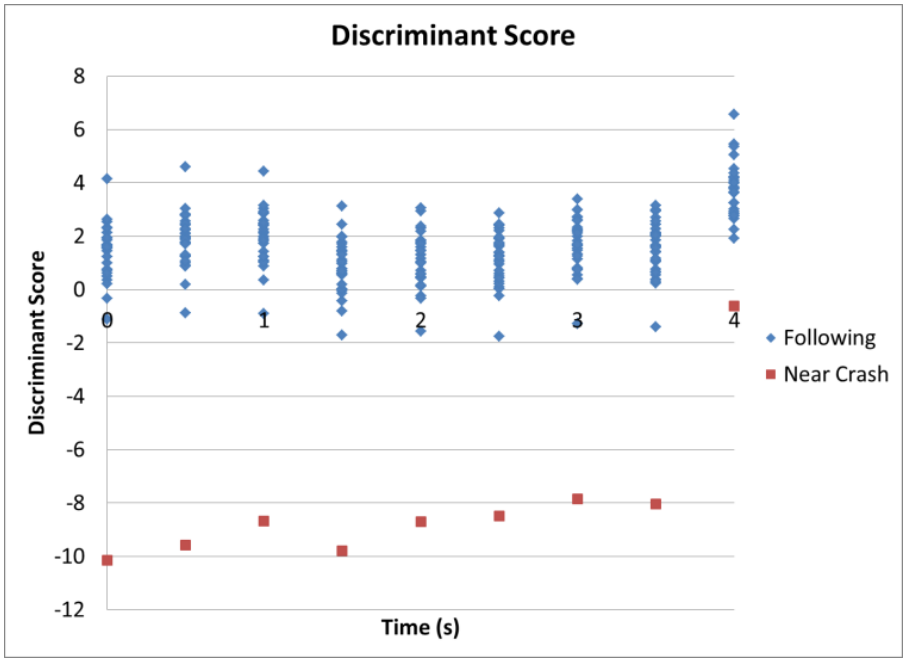

Figure 121. Graph. Discriminant scores for Driver E.

Figure 122, Figure 123, and Figure 124 show the graphs of some of the terms that constitute the discriminant score for Driver E. These graphs are the result of multiplying the input variables by the corresponding discriminant coefficient.

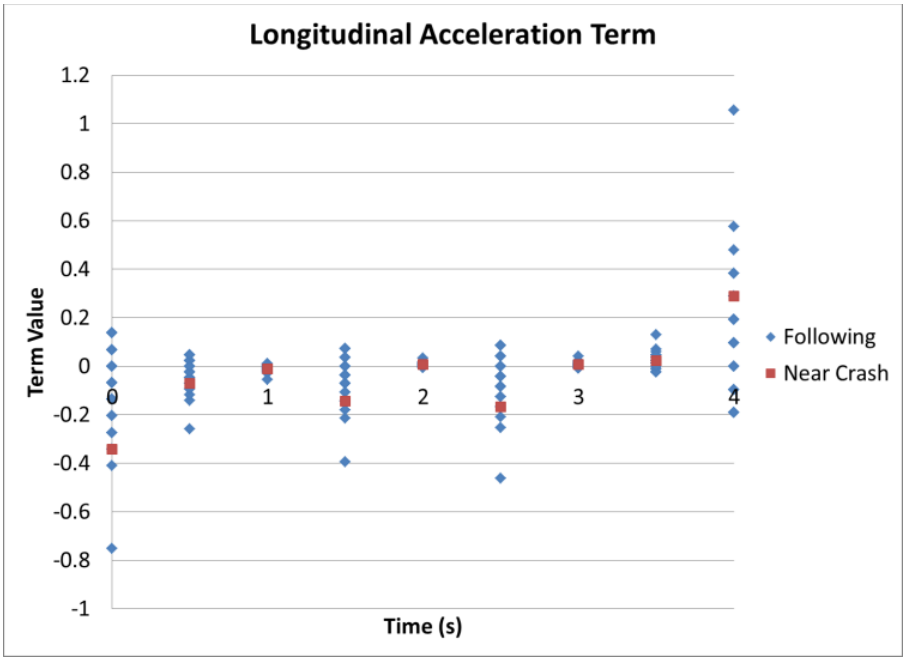

Figure 122. Graph. Longitudinal acceleration term for Driver E.

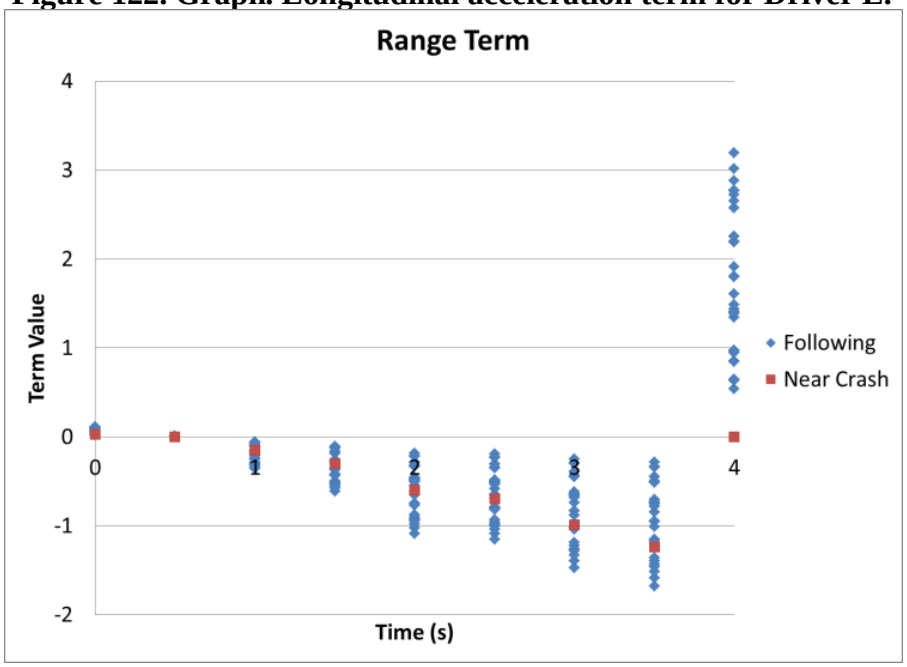

Figure 123. Graph. Range term for Driver E.

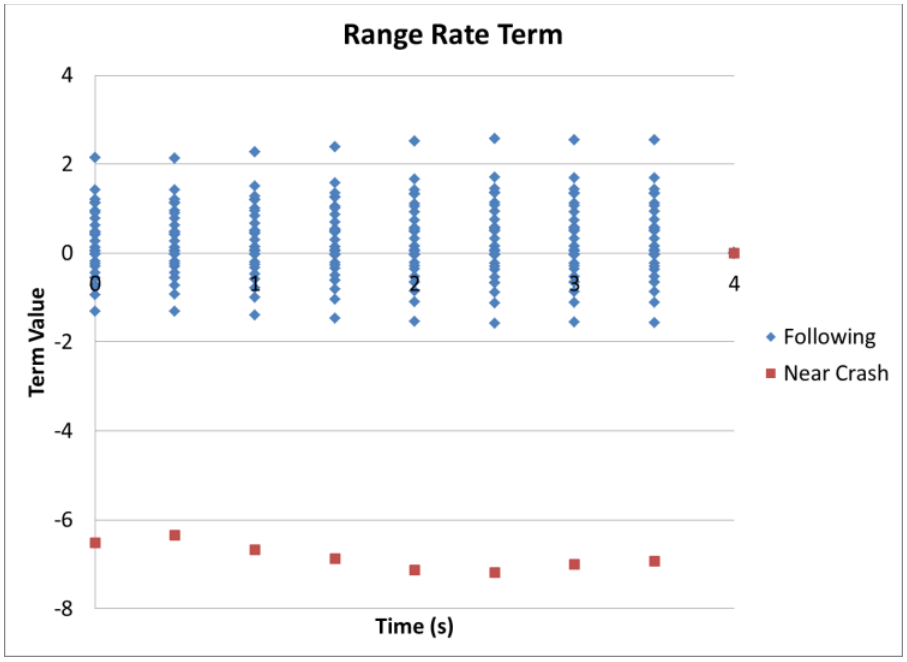

Figure 124. Graph. Range rate term for Driver E.

Table 32 shows the discriminant coefficients for Driver G. Figure 125 shows the percentage of misclassification associated with each set of discriminant coefficients.

Table 32. Discriminant coefficients of Driver G.

Time	Longitudinal Acceleration	Lateral Acceleration	Speed	Lane Offset	Yaw Angle	Range	Range Rate
0	19.7	-1.3	-0.012	0.007	-79.1	0.03	-0.05
0.5	35.4	14.8	-0.007	0.024	-52.9	0.02	0.05
1	35.7	11.5	-0.009	0.025	-59.2	0.02	0.04
1.5	37.2	9.8	-0.009	0.028	-55.2	0.01	0.05
2	33.9	4.2	-0.014	0.023	-65.9	0.02	0.01
2.5	1.1	-14.5	-0.027	-0.004	-76.8	0.04	-0.11
3	0.8	21.5	0.026	0.006	69.4	-0.03	0.13
3.5	5.2	21.3	0.026	0.011	48.6	-0.03	0.15
4	12.1	11.6	0.025	0.017	27.5	-0.03	0.14
4.5	4.1	27.3	0.023	0.010	60.4	-0.03	0.15
5	-3.2	-3.2	-0.026	-0.007	-74.2	0.04	-0.08
5.5	5.9	-15.5	-0.026	-0.002	-75.2	0.03	-0.11

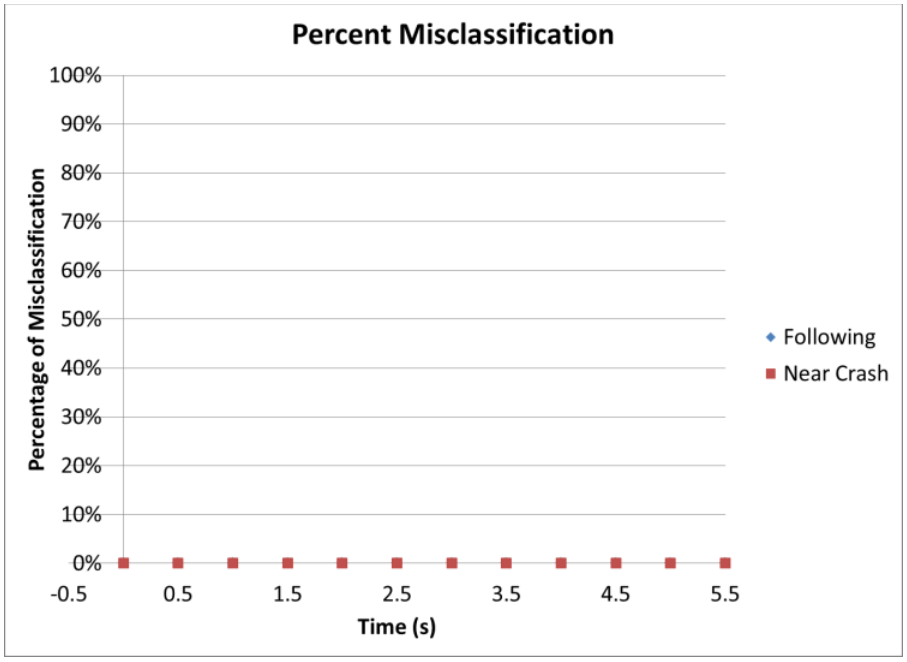

Figure 125. Graph. Percentage of misclassification for Driver G.

Figure 126 shows the discriminant scores for the data points from both a near-crash event and normal car-following periods. These scores are calculated by multiplying the discriminant coefficients by the appropriate data values and then taking the summation.

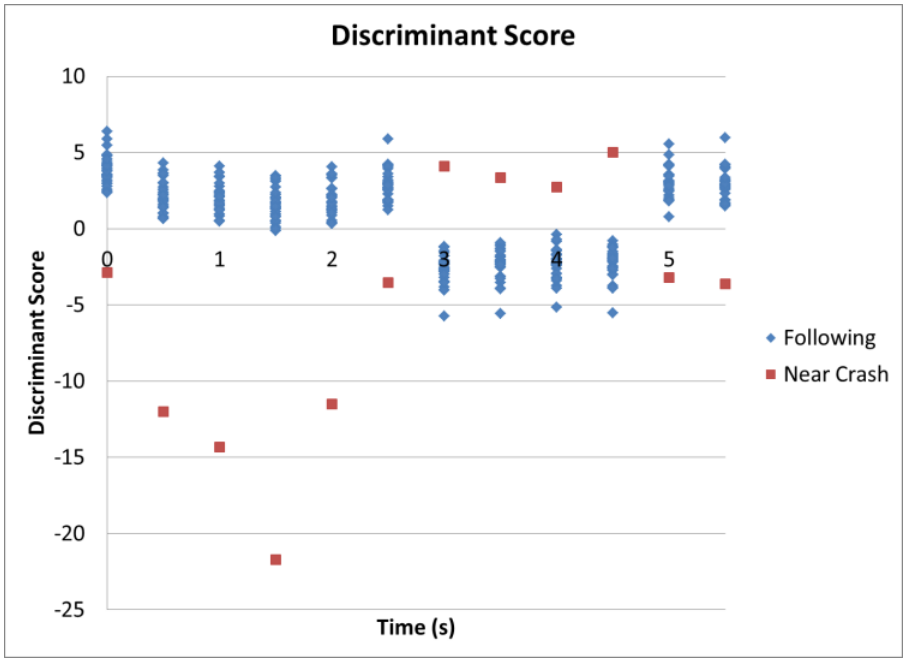

Figure 126. Graph. Discriminant scores for Driver G.

Figure 127, Figure 128, and Figure 129 show a further breakdown of the discriminant score for Driver G into three of the six terms: longitudinal acceleration, range, and range rate.

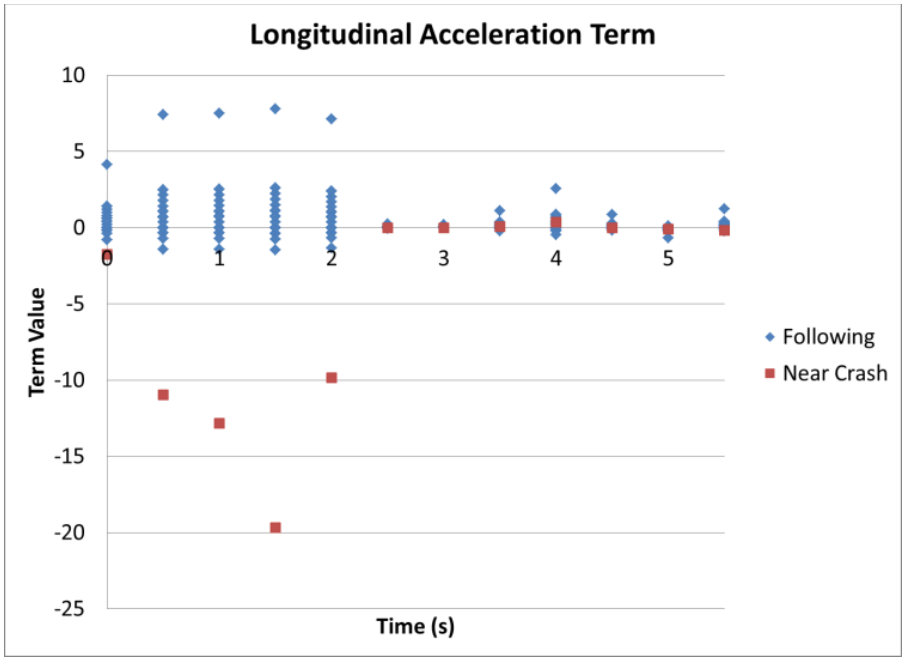

Figure 127. Graph. Longitudinal acceleration term for Driver G.

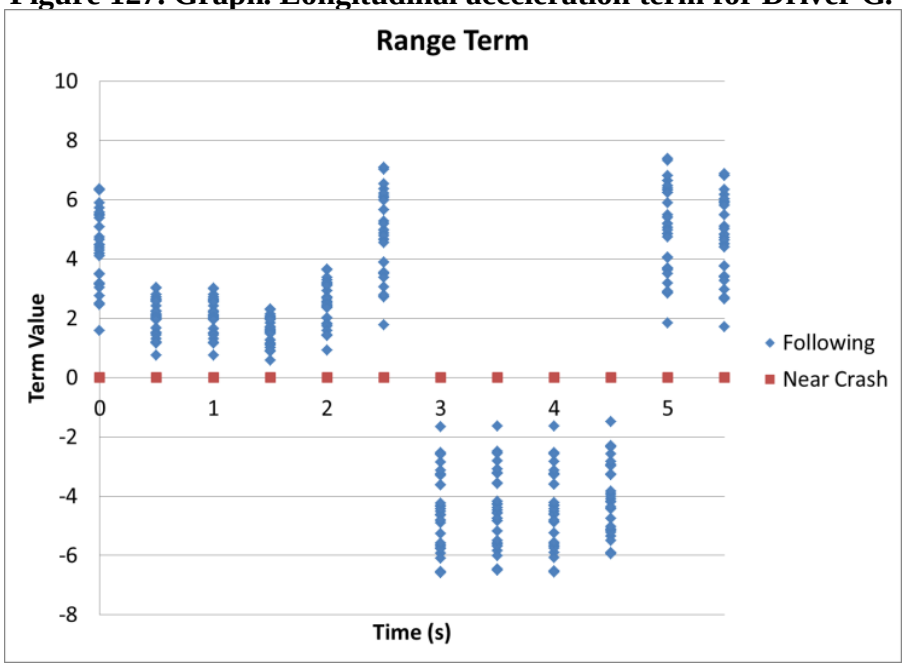

Figure 128. Graph. Range term for Driver G.

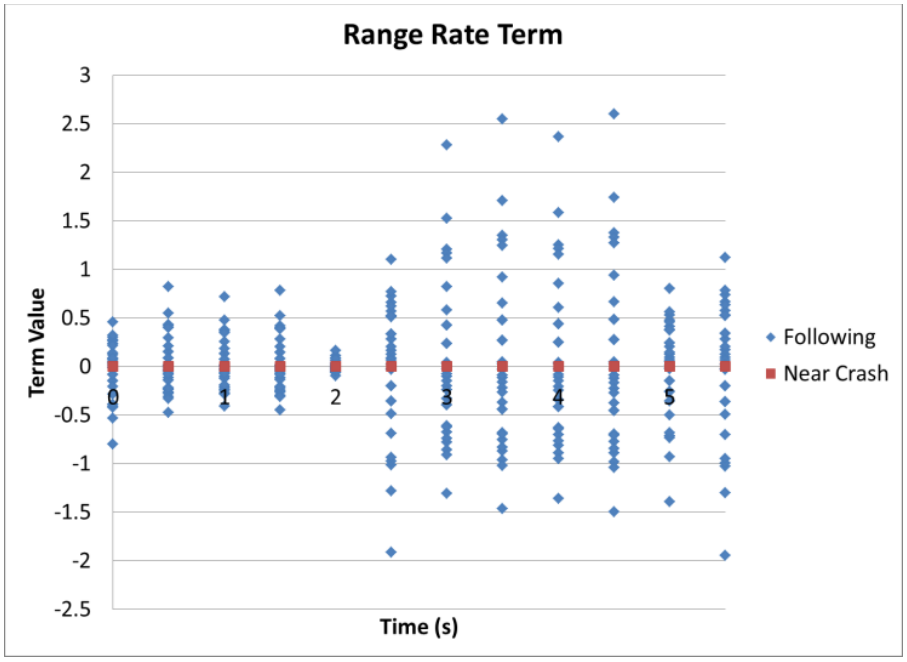

Figure 129. Graph. Range rate term for Driver G.

Table 33 shows the results of a discriminant analysis using the data from both drivers. The reason for this analysis is that truck drivers have more similarities than car drivers and may thus show common behaviors that would not exist between car drivers. The resulting coefficients from the analysis are then applied to both drivers in Figure 130 and Figure 131. The figures show that the coefficients are very accurate for both drivers until after 2 s. At that point, misclassification begins to occur for Driver G. This misclassification error is classifying a safety critical event as normal behavior, which is not acceptable.

Table 33. Discriminant coefficients from joint analysis.

Time	Longitudinal Acceleration	Lateral Acceleration	Speed	Lane Offset	Yaw Angle	Range	Range Rate
0	19.8	9.6	0.016	0.012	-28.6	0.001	0.11
0.5	25.8	15.3	0.018	0.018	-17.4	-0.002	0.12
1	25.4	13.9	0.019	0.018	-20.7	-0.003	0.13
1.5	21.5	12.9	0.020	0.017	0.6	-0.004	0.14
2	28.4	11.8	0.019	0.019	-7.5	-0.003	0.12
2.5	0.4	6.7	-0.002	0.005	-21.0	0.002	0.13
3	2.1	0.8	0.001	0.006	-15.8	0.001	0.13
3.5	-1.2	2.8	0.001	0.003	18.8	0.000	0.13
4	-7.3	6.8	-0.014	-0.012	38.4	0.012	-0.04

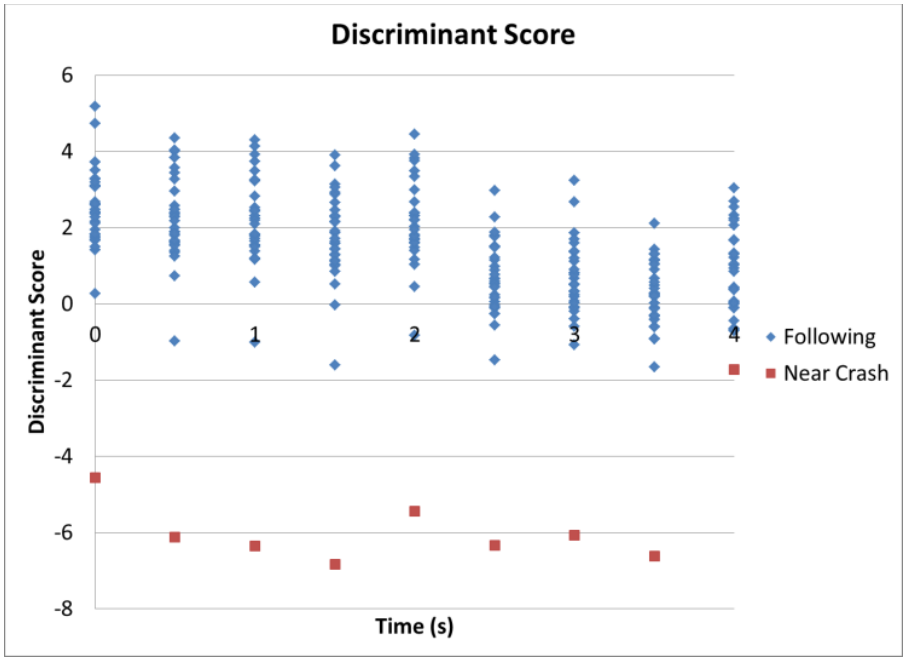

Figure 130. Graph. Discriminant score for using the joint coefficients for Driver E.

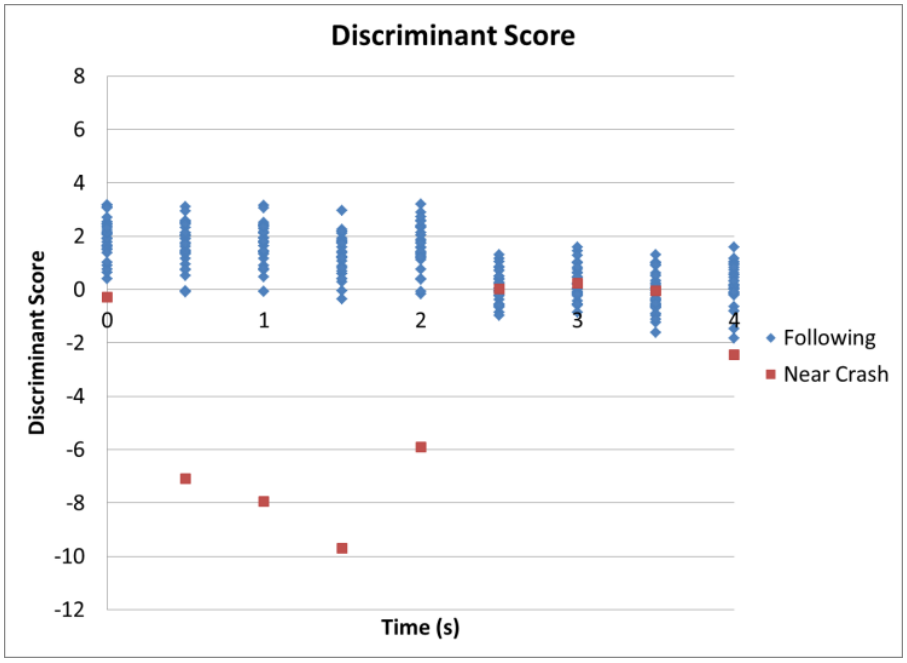

Figure 131. Graph. Discriminant scores for using the joint coefficients for Driver G.

Table 34 and Table 35 presents the results of applying the described procedure for the two example drivers to eight more truck drivers from the two naturalistic truck driving studies. The results show that each driver has a specific combination of the six variables and a specific threshold value that can be used to distinguish between normal driving behavior and safety critical event behavior. Also, the drivers's times vary, which means that the distinction between normal driving and safety critical behavior can be made at different times before the events for each driver. The largest time used in the analysis for some drivers was 6 s.

Table 34. 34-Truck driver-specific coefficients and thresholds.

Driver	Time	Speed	Range	Range Rate	Acceleration	Yaw	Offset	Threshold
B	6	0.584	-0.006	0.022	22.477	0.000	-0.003	-4.048
I	6	0.050	0.000	0.291	18.283	-272.421	-0.105	-15.492

Table 35. 8-Truck driver-specific coefficients and thresholds.

Driver	Time	Speed	Range	Range Rate	Acceleration	Yaw	Offset	Threshold
D	5.5	0.142	-0.005	0.186	8.525	10.129	-0.025	6.176
E	4	0.012	-0.002	0.203	-3.060	2.745	-0.051	-5.666
F	5.5	0.001	-0.013	0.123	16.541	-53.312	-0.020	-5.666
G	5.5	0.022	-0.043	0.149	-5.273	-56.211	0.004	-10.683
H	1	0.036	0.000	0.045	-11.439	-96.941	-0.050	-1.646
M	1	0.036	0.000	0.045	-11.439	-96.941	-0.050	-1.646
N	8.5	0.017	-0.012	-0.010	-21.512	-20.721	0.008	-3.299
O	1.5	0.735	-0.005	0.011	11.958	14.281	-0.049	33.157
P	3.5	0.112	-0.009	0.134	8.647	12.024	0.028	-1.956

Conclusions and Recommendations

The results show that a discriminant analysis of safety critical event data, as compared to normal car-following data, yields a method of distinguishing between the two sets of data. This distinction uses six variables: longitudinal acceleration, vehicle speed, lane offset, yaw angle, range, and range rate. The results support the hypothesis that using the six selected variables, along with the corresponding coefficients, creates a discriminant score that can accurately distinguish between safe and unsafe driving conditions. For Driver E, this distinction can be made 4 s before an event occurs.

Also, in the case of Driver G, this distinction can be made 2 s before an event occurs. This means that the warning signs of an impending safety critical event are captured and represented by the six variables and a specific combination of these variables as defined in the results of the discriminant analysis.

The joint discriminant analysis results show that a set of coefficients can be created that will work accurately for both drivers. This suggests that multiple drivers can be grouped together and analyzed, which simplifies situations with numerous drivers.

CHAPTER CONCLUSION

Finding the difference between normal driving behavior and safety critical event behavior can be difficult. The statistical process of a discriminant analysis facilitates finding this difference. In this case, a discriminant analysis is used to find the specific combination of six variables that result in a visual difference between two datasets. Also, it was found that the difference between normal behavior and safety critical behavior extended beyond the actual event into the behavior immediately before the event. In the examples shown, there was a visible transition from safe or

normal behavior to safety critical behavior. This resulted in the ability to turn the agent behavior on and off. The trained agents offer benefits to behaviors in safety critical events. Further, because simulation packages are designed to follow specific car-following models, the ability to turn the agent behavior on and off can be beneficial. This also means that the agents could be trained only for the safety critical event behavior, and the car-following models can be used otherwise.

Chapter 7. AGENT IMPLEMENTATION IN VISSIM

INTRODUCTION AND PURPOSE

This chapter describes the agent implementation in VISSIM. The purpose of the demo prototype included in the deliverables of this project is to show the complete simulation chain, from generating parameter files to actually selecting the agent in VISSIM and observing its driving behavior. The following points are demonstrated:

- How to produce an agent parameter file with the APC GUI.
- How to define agent vehicles in VISSIM, including the reference to the DLL file and the parameter file.
- What settings to adjust in VISSIM to have "pure" agent behavior that is not manipulated by internal VISSIM plausibility checks.
- Behavior of the agent model in case of an event that is close to an event observed in reality.

OVERALL WORKFLOW

Figure 132 shows the overall flow of the implementation framework. A parameter file is used to get the model parameters into VISSIM using the driver model DLL. The parameters used are those generated by the techniques described in the previous chapters. The implementation framework utilizes one parameter file for each agent; in other words, for 20 agents there will be: parameter files *ParameterFile_Agent_1.APC, ParameterFile_Agent_2.APC, ... ParameterFile_Agent_20.APC.*

In VISSIM, vehicle types can be defined that use a specific DriverModel.DLL as an external driver model. (See the description in the next section and the VISSIM manual section 5.3.1. Note that the parameter file referred to in the manual and in the VISSIM Vehicle Type dialog does not conform to the parameter files used in this project.)

The DriverModel.DLL writes a specific output file with information about the driver state, and so on. To allow the files to be written properly, there must be a specific DLL for each agent in the model. The name of the DLL is part of the output file name (which will be saved in the current working directory).

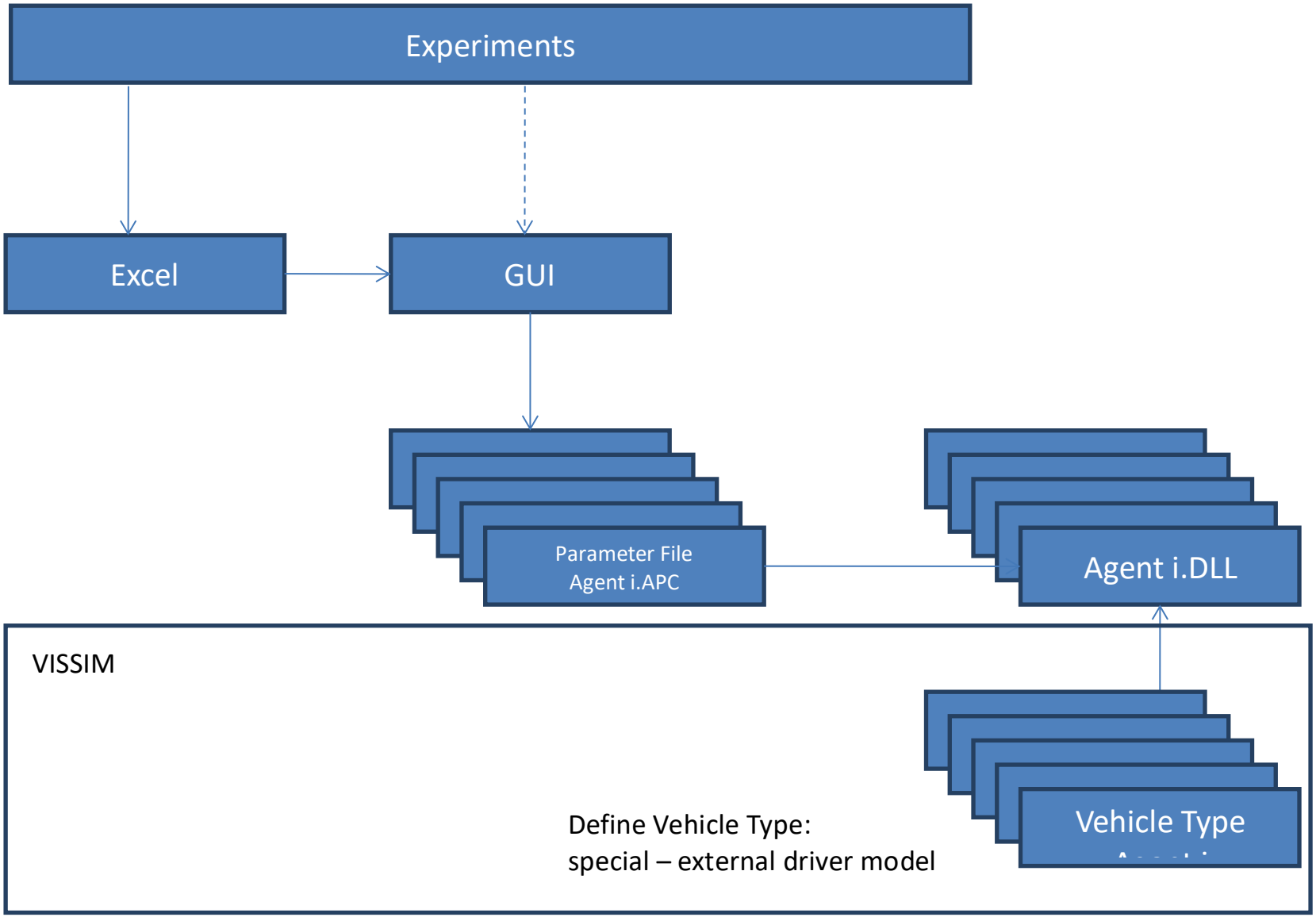

Figure 132. Chart. Overall workflow.

VISSIM SETTINGS

Vehicle Types

Two new vehicle types are defined: agent and leader. The respective driver model DLL and parameter files should be selected as shown in Figure 133 (file locations have to be adjusted to the local machine).

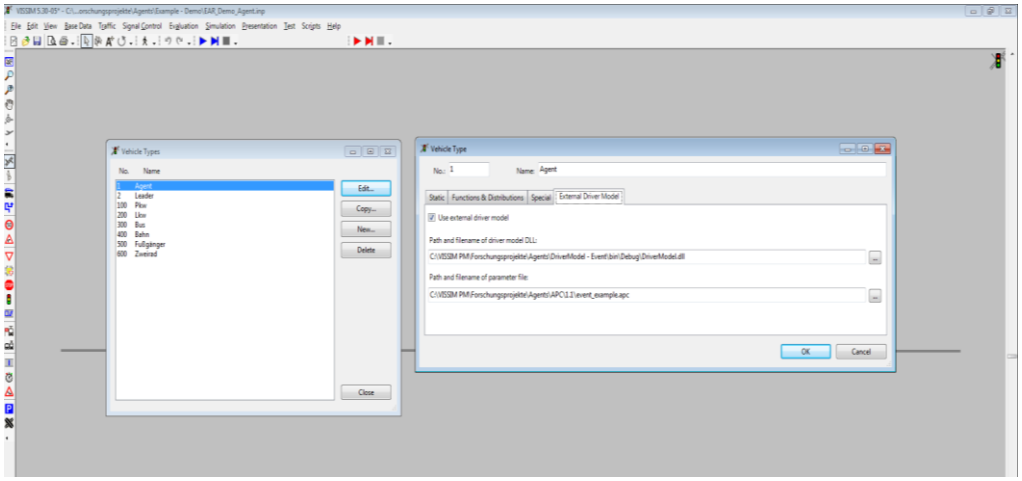

Figure 133. Screen shot. Specification of vehicle types to use the agent model.

2D/3D Model Distribution

For the exact interpretation of the lane angle set by the Agent Model DLL, the geometry of the 2D model used for the agent vehicle needs to be adjusted. To make sure everything works properly, perform the following steps:

1. Check the Model Distribution that is used for the Agent vehicle type (e.g. 10 PKW) (Figure 134)

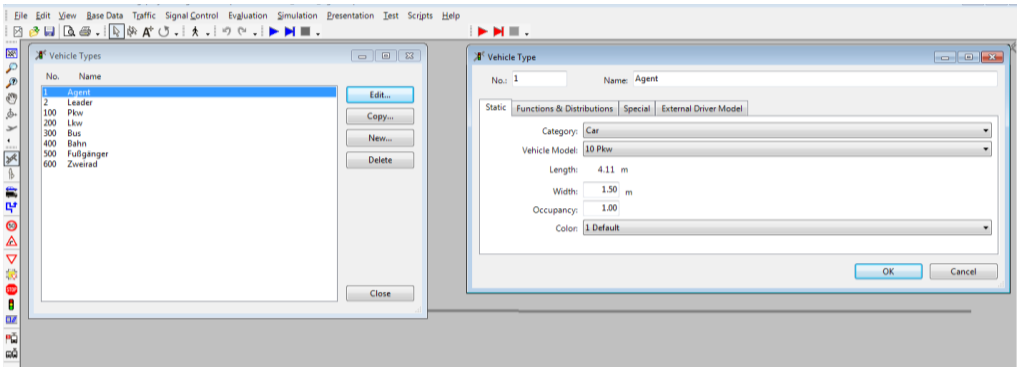

Figure 134. Screen shot. Model distribution for agent vehicle type.

2. Open the Model Distribution dialog and edit the Model Distribution (Figure 135):

a. Use only one model in the distribution (i.e., set the share for all models except one to zero).
b. For that model, move the rear axle to the rear end of the vehicle (i.e., set Rear Axle and Rear Joint to the same value as the vehicle length, e.g., 4.110 m).

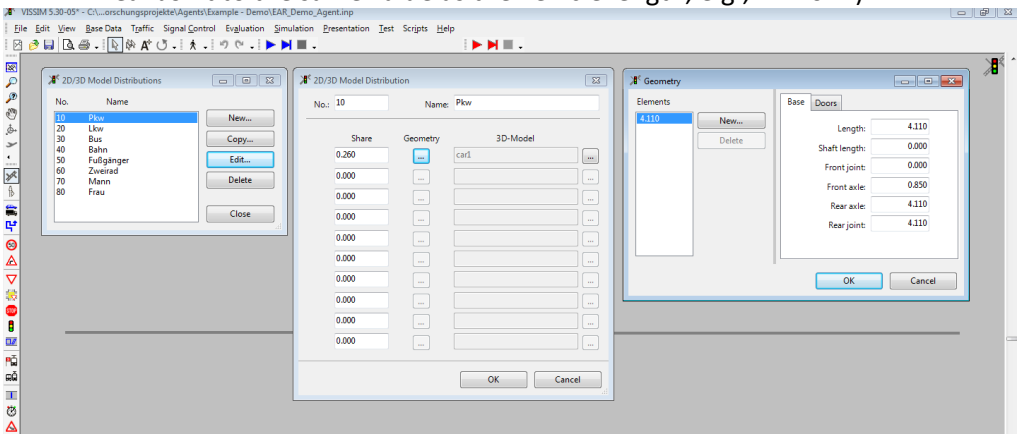

Figure 135. Screen shot. Model Distribution Dialog.

DRIVING BEHAVIOR

In order for the lane angle to be exchanged between VISSIM and the Agent Model DLL, the "Observe vehicles on next lane(s)" option in the Driving Behavior Parameter Sets dialog (Figure 136) need to be activated (even if there is only one lane).

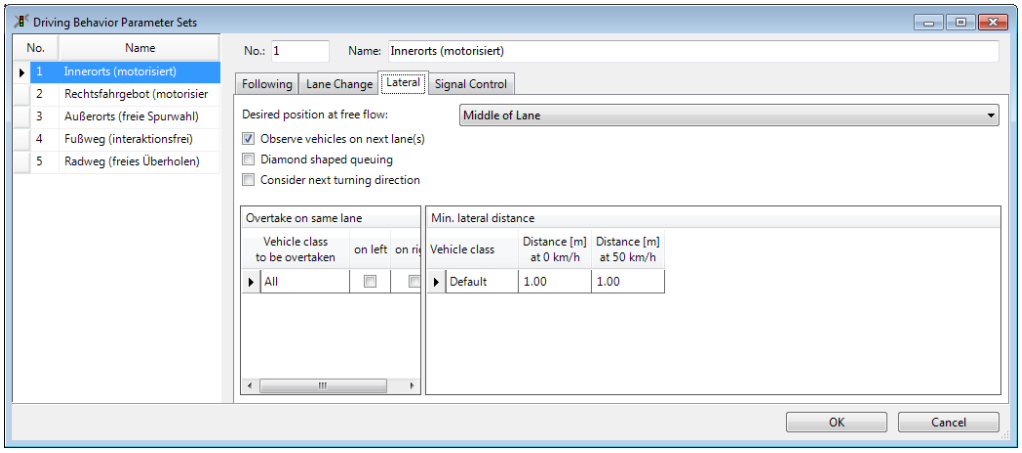

Figure 136. Screen shot. Lateral driving behavior.

ACCELERATION AND DECELERATION FUNCTIONS

In VISSIM, acceleration and maximum deceleration functions for the vehicles controlled by the DLLs have to be introduced. This is because VISSIM checks if the acceleration and deceleration proposed by the DLL lies within the limits defined in VISSIM by these functions. Namely, VISSIM sets the applied acceleration to the minimum and maximum desired acceleration and the applied deceleration to the minimum and maximum desired deceleration. The maximum acceleration and deceleration functions are named in the "DLL." The maximum acceleration function is defined as $+7 m/s^2$ at any speed, and the maximum deceleration function is defined as

-100m/s^2 at any speed. These limits are clearly out of the feasible range and should not be used for normal simulations. In the demo, they are used to produce accident-like decelerations (leader) and to show the pure agent behavior for the follower.

PARAMETER FILE *.APC

GUI

There are two ways to get the data into the parameter file:

1. The model parameters are entered in an Excel file (with the same structure of the file: *NFACRL structure thresholds and weights_example-March 2011.XLSX*). As the model is implemented in the spreadsheet, the response of the model (outputs Acceleration and Lane Angle) can be checked using different state variables. In this way, the Excel file can act as kind of a plausibility check for the model parameters. The parameters can be imported to the *Agent Parameter GUI.* From the GUI, the parameter file can be written.
2. The second option is to enter the parameters directly into the GUI and export them into the parameter file. This is not recommended for tutorial purposes because it is more cumbersome and there is no way to check if the model will produce meaningful output (of course, it can be checked once the parameter file is loaded to the driver model DLL and the parameters are used in VISSIM).

In addition to the model parameters, there are threshold values to be defined. The thresholds define when the driving behavior switches from standard VISSIM (i.e., Wiedemann model) to agent-based behavior and vice versa.

To parameterize the agents, a GUI is provided (as illustrated in Figure 137 and Figure 138). The GUI gives access to all user definable parameters of the agent, as well as to a global threshold value that determines in what situations the agent model should be used instead of the standard VISSIM driving behavior.
To parameterize the agent, the GUI provides tabs for the different model weights and thresholds, as well as a text field for the global threshold.

There is a button to read the values directly from an Excel file with the same structure of the file: *NFACRL structure thresholds and weights _example-March 2011.XLSX*.

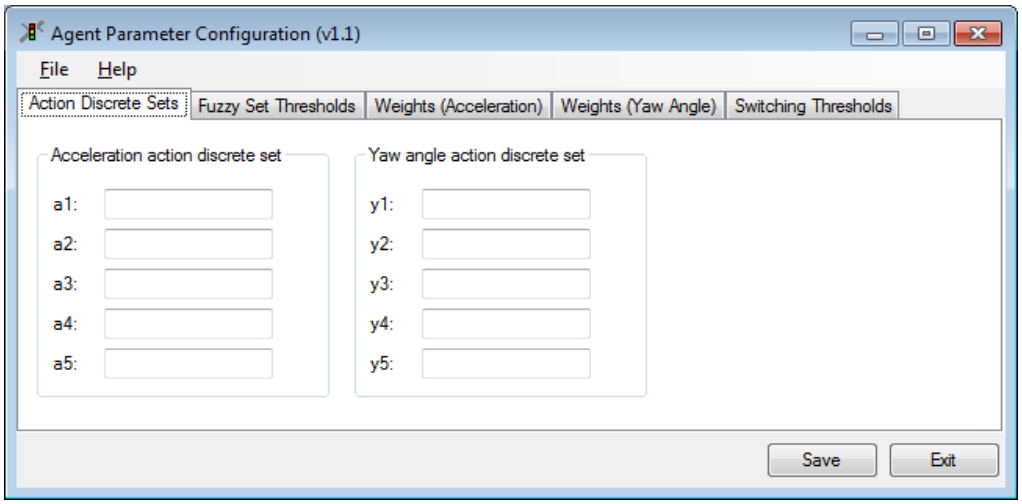

Figure 137. Screen shot. Agent parameter GUI—action discrete sets.

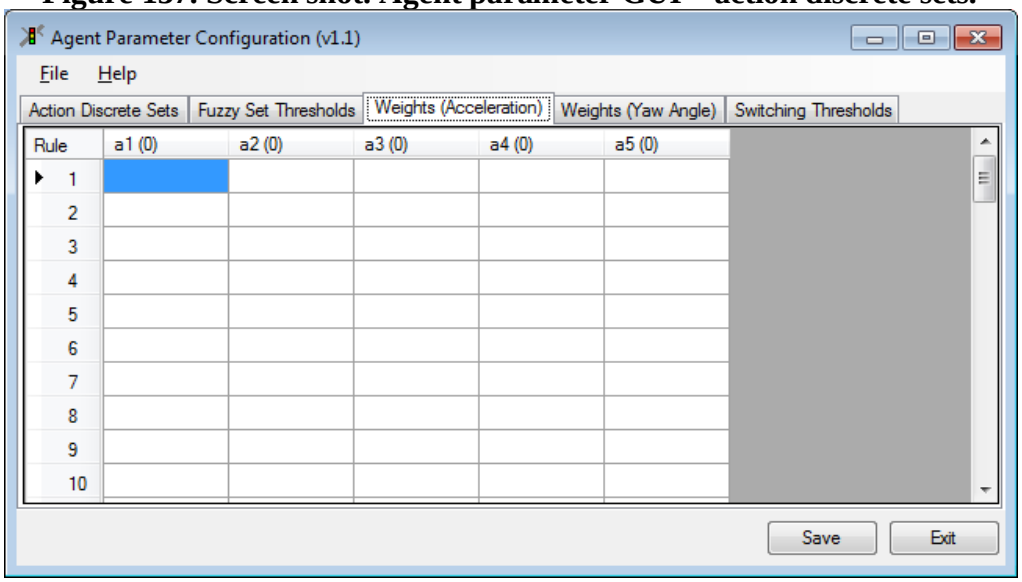

Figure 138. Screen shot. Agent parameter GUI—acceleration weights.

For this project, there is one type of agent model—in other words, the number of neurons, the rules, the layers, and the calculation logic is set in the DLL. Any changes in the model structure (such as adding a new layer) will require an updated driver Model DLL and a changed GUI to set the parameters.

Parameter Contents and Format

Table 36 shows a description of the parameter file format. The parameter file is a comma-separated text file, and all numbers are double format. There should be one parameter file per agent (i.e., for 20 agents there will be files *parameterFile_[n].APC*, with n = 1...20).

Table 36. Parameter file description.

Description	Values (in blocks, separated by commas); See APC

	Documentation
Fuzzy Set Layer Thresholds	Fuzzy Set Layer Thresholds
thresholdSpeed:	LowerthresholdSpeed, upperthresholdSpeed
thresholdRange:	LowerthresholdRange, upperthresholdRange
thresholdRangeRate:	LowerthresholdRangeRate, upperthresholdRangeRate
tresholdAcceleration:	LowerthresholdAcceleration, upperthresholdAcceleration
thresholdLaneAngle:	Lower thresholdLaneAngle, upper thresholdLaneAngle
thresholdOffset:	Lower thresholdLaneAngle, upper thresholdLaneAngle
AccelerationDiscreteSet:	a1,a2,a3,a4,a5
LaneAngleDiscreteSet:	la2,la2,la3,la4,la5
Weights W1(i,j):	W1(1,1), ..., W1(1,5), W1(2,1), ..., W1(i,j),....,W1(32,5)
Weights W2(1,j):	W2(1,1), ..., W2(1,5), W2(2,1), ..., W2(i,j),....,W2(32,5)
Switch Conditions	There will be one (or more) threshold value that determines if the agent model is switched on or off. At the moment, this is Time to Collision. It can, of course, be changed to other values.

SCENARIO DESCRIPTION "EVENT"

Description

The example presented here is based on an observed real-world scenario. It should show the behavior of the VISSIM implementation of the agent in comparison to the observed data and the behavior of the Excel implementation of the agent. The following vehicle is controlled by an agent according to the parameters provided in the example Excel file. Because external driver model *DriverMode.DLL* has to be selected, the parameter file is *Event_Example.APC*.
It should be noted that directly comparing VISSIM output to naturalistic data output can be problematic for two reasons:

1. All naturalistic data variables are observed variables, whereas the same variables are a mix of observed and calculated state variables in VISSIM. In other words, the range and range rate are VISSIM observed state variables; the others are calculated state variables. Substituting the range and range rate into the VISSIM model will not always produce the corresponding calculated variables (in VISSIM, knowing the acceleration of the follower and the speed of the leader will result in exact calculation of the range and range rate for the next time step; in naturalistic data, the leader might perform any action that will result in a different range and range rate). This might lead to discontinuity in the state-space surface. For example, if the data showed that the vehicle stopped, and the model predicts that that there is acceleration > 0, the next modeled acceleration will have to either use the observed data as the previous state, or accumulate a shift in the modeled state from the observed state. In the provided Excel sheet, the model uses a mixture of observation and previous model output as state variables for the next time step. In the VISSIM model, range and range rate are naturally simulation outputs, and they respond to the model output (i.e., if an acceleration > 0 is calculated, the vehicle accelerates regardless of any observation that is saved in the spreadsheet).

2. Another issue that should be noted is that the example results start with an initial state in terms of speed, range, range rate, acceleration, lane angle, and offset. It is not possible to create exactly that initial condition in VISSIM such that, starting from there, the results of the Excel implementation and VISSIM can be compared. That is because of limitations of the DLL interface and the discrete time steps. In other words, it is possible to generate a state that is very close to the initial state described in the observed data, but it will not be the very same initial state.

These two facts (the former more than the latter) suggest that the Excel model and the VISSIM model should be compared by other means:

1. Generate a test case in VISSIM (close to the observed event) and run the simulation with the agent model switched on if the follower is as close to the leader as in the observed event.
2. Use the state variables from VISSIM (instead of reality) to calculate the model's output in Excel.
3. If the two outputs are identical (or close to identical, due to numerical reasons), it shows that the VISSIM implementation behaves like the Excel implementation.

How to run

Run COM Script

For the leading vehicle to perform the observed trajectory, the COM interface is used to set the speed of the leading vehicle. To run the simulation, the button "Run VISSIM Example" (see Figure 139 below) in the sheet "VISSIM-COM" must be clicked.

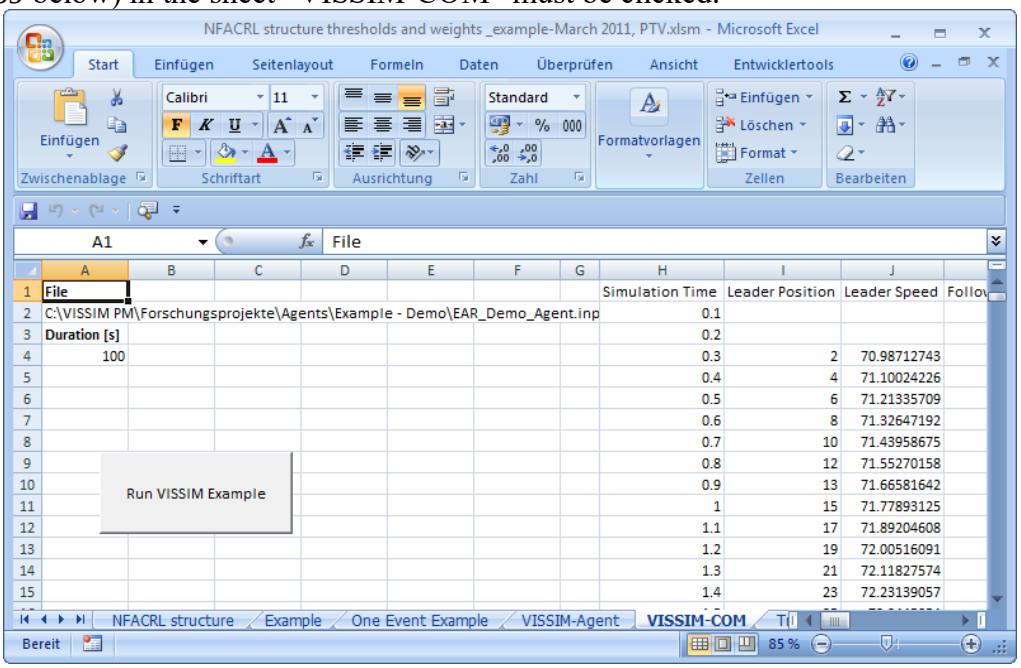

Figure 139. Screen shot. "Run VISSIM example" command.

The leading vehicle drives at relatively low speed; the following vehicle enters the network 10 s later with a higher speed (approximately constant speed is assumed). The DLL and the COM script constantly check to see if the (steadily decreasing) range between the vehicles falls below a threshold (120.9 ft) that is derived from the observation data provided in the Excel spreadsheet. If this condition is fulfilled, the leading vehicle's speed is set according to the observed values

(gathered from the Excel sheet by the COM script) and the action of the following vehicle calculated based on the Agent Model DLL.

Copy Output Data from Text File to Excel

The simulation output is written to a text file "Data_Test_[Name of the used parameter File].txt" and saved in the working directory. The content of this file needs to be copied to the Excel file (mark the whole content in the text file and paste in the Excel file, sheet "VISSIM-Agent," at cell A1.)

Run the Excel Model

To run the Excel model to compare the output derived from Excel with the model output in VISSIM, click the button "Run Excel Agent based on VISSIM data" in sheet "VISSIM-Agent" as shown in Figure 140.

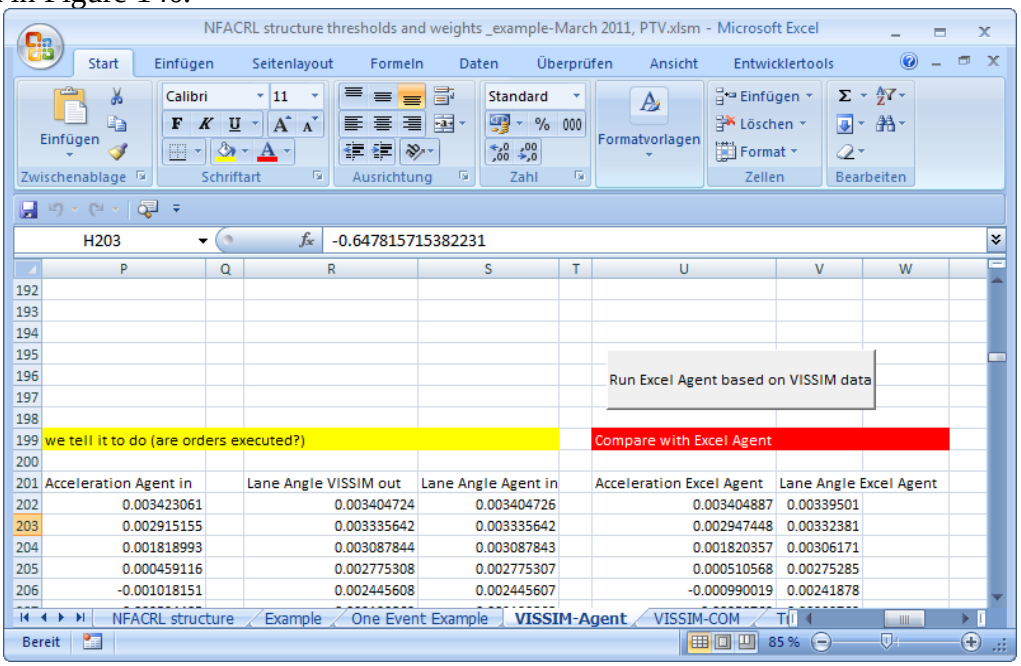

Figure 140. Screen shot. "Run Excel Agent based on VISSIM data" command.

Figure 141 through Figure 145 shown in the next section are generated automatically.

Output

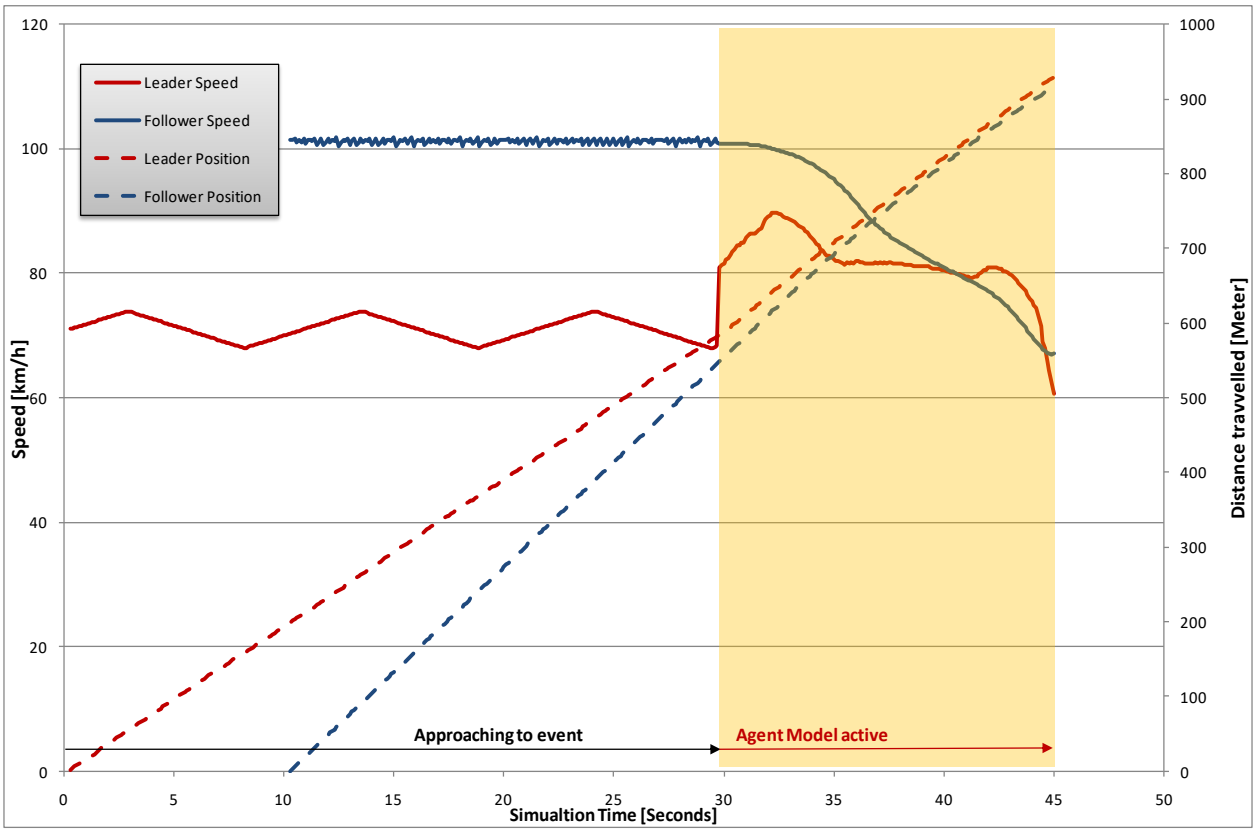

Figure 141. Graph. Comparison of speeds and trajectories of leading vehicle and agent over time.

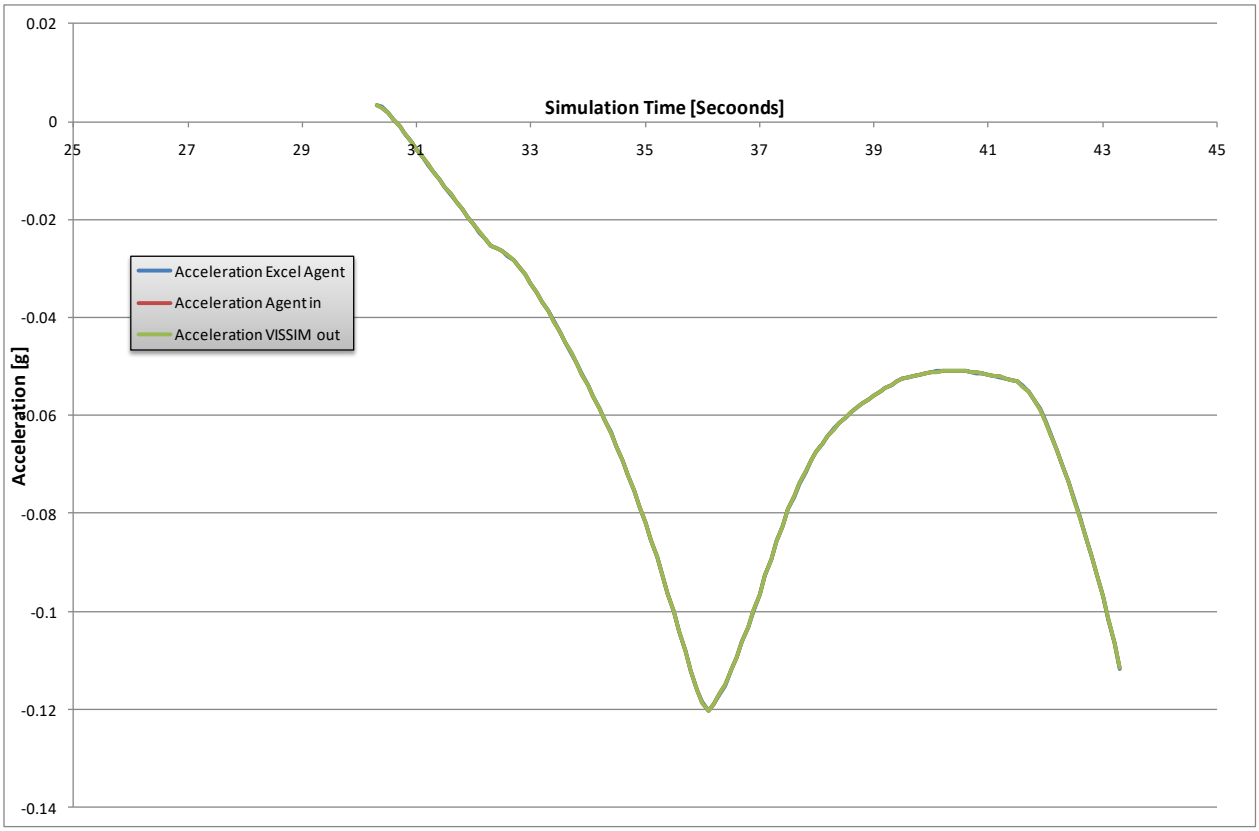

Figure 142. Graph. Comparison of acceleration over time (acceleration calculated in Excel as reference, acceleration calculated in VISSIM and performed in VISSIM to be compared).

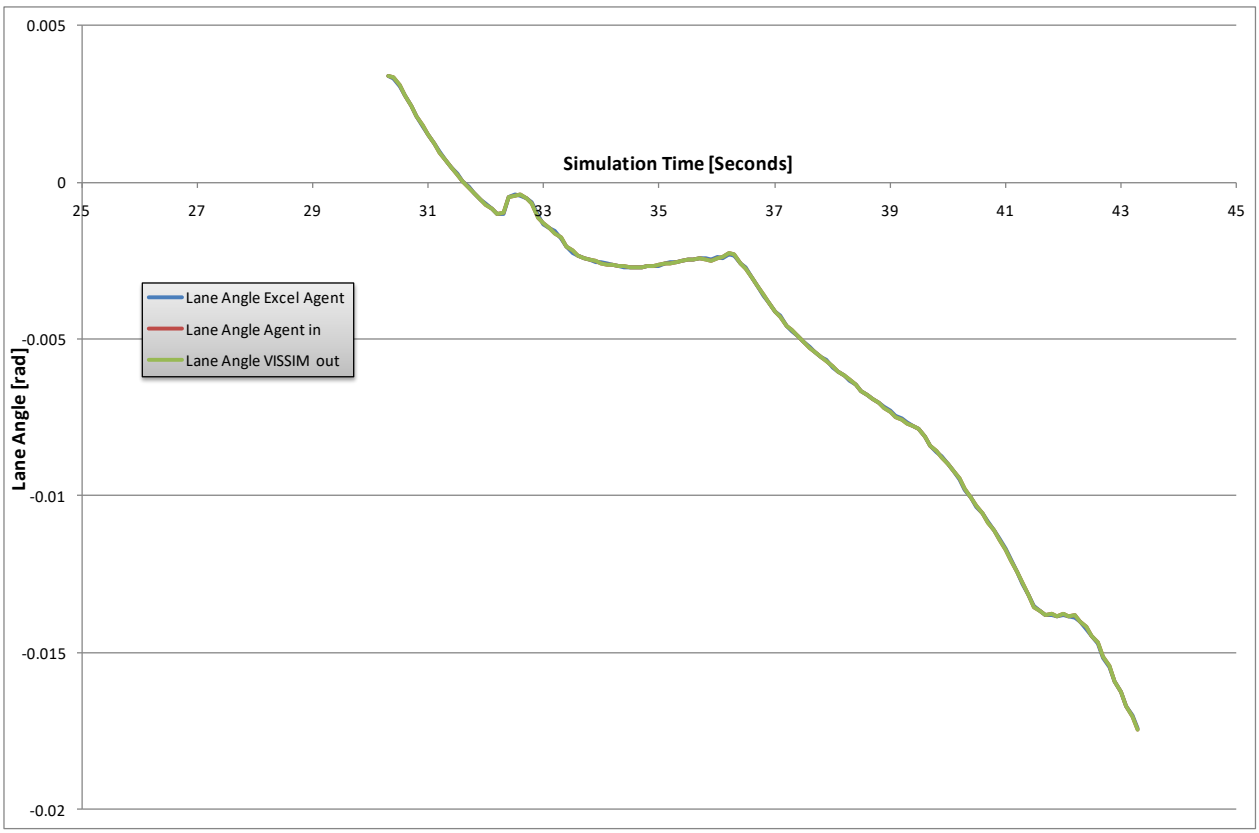

Figure 143. Graph. Comparison of lane angle over time (lane angle calculated in Excel as reference, lane angle calculated in VISSIM and performed in VISSIM to be compared)

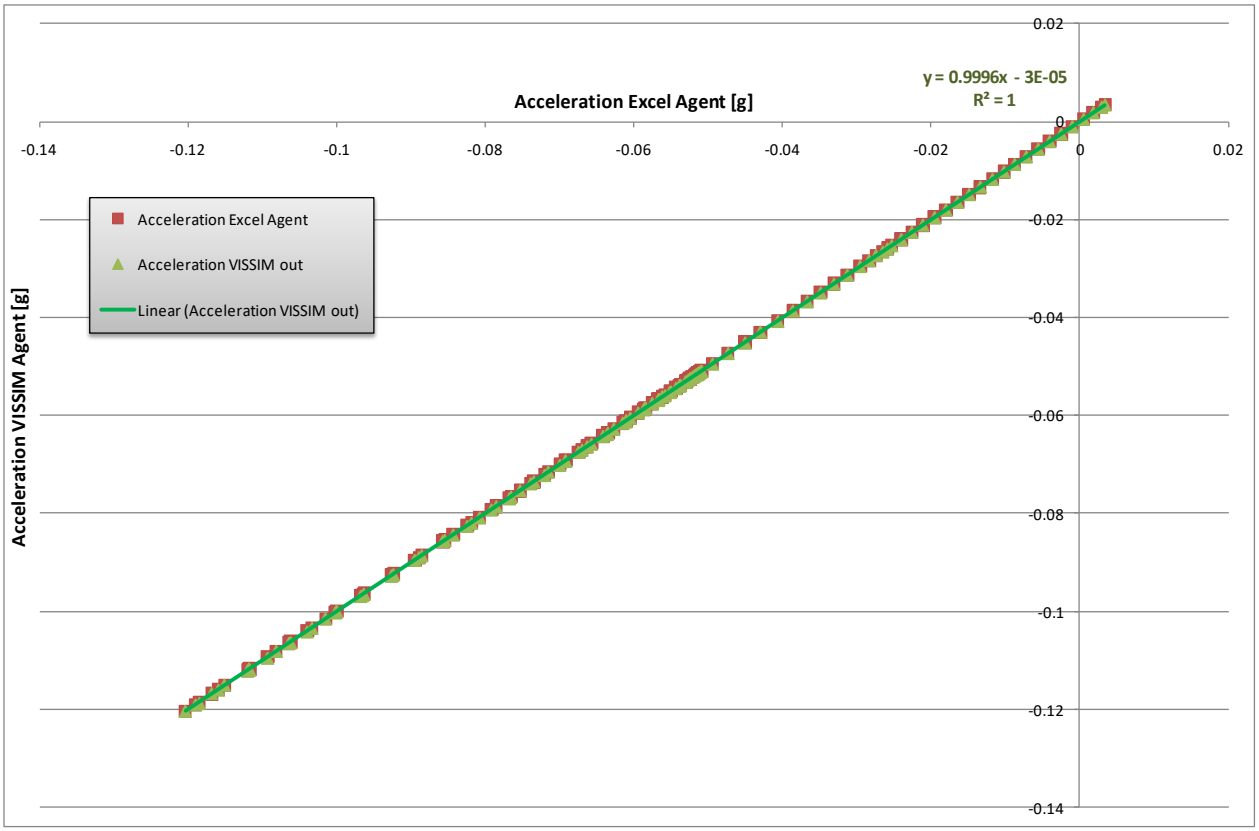

Figure 144. Graph. Acceleration scatter plot with linear regression analysis, Excel agent versus VISSIM agent.

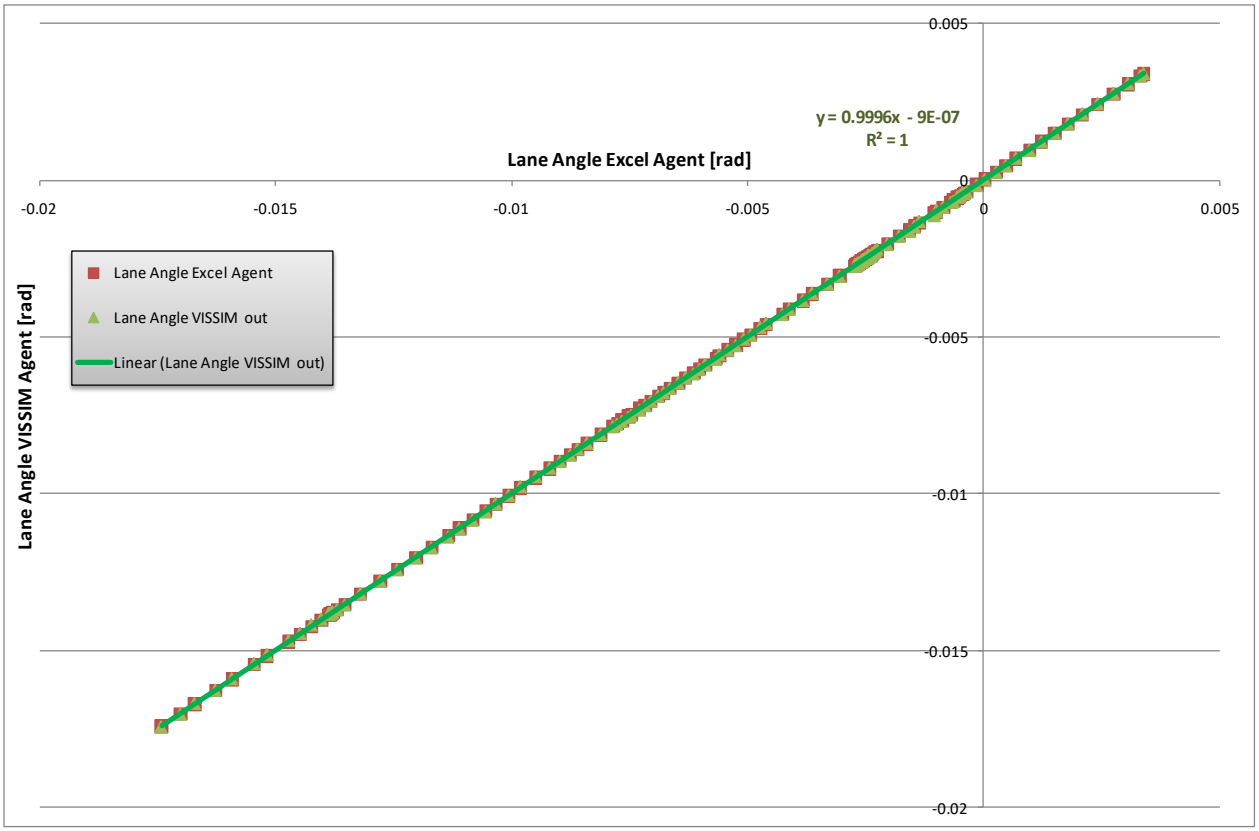

Figure 145. Graph. Lane angle scatter plot with linear regression analysis, Excel agent versus VISSIM agent.

For testing, there is another text file written to the working directory called "Test_[Name of the used parameter File].txt". It contains all the parameters from the parameter file, several pieces of intermediate data, and the results of the model for each time step. This file can be used to compare the agent behavior in VISSIM with the agent behavior in the Excel representation of the model as well.

Chapter 8. SIMULATION AND ANALYSIS

INTRODUCTION

Previous tasks have illustrated the feasibility of developing and training agents to respond to safety critical events in a way that mimics the behavior of actual drivers observed in the naturalistic dataset. This chapter documents the results of the task with the intent to insert those agents into VISSIM and compare their performance to default drivers using the Wiedemann model for driver behavior.

Simulation Environment

The agents were tested within VISSIM version 5.30, build 8. Very simple networks were created for each of the 20 developed agents based on scenarios derived from the naturalistic data. The networks consisted of one-lane roadways with a leader and following vehicle. The leader is designed to behave in a way that creates a safety critical event and causes the follower (the agent) to react. The duration of each simulation is 100 s. Simulation runs were conducted with and without the agents activated so that the impacts of the agent behavior could be evaluated relative to the Wiedemann model. Each of the 20 scenarios (both with and without the agent) was replicated 20 times to account for the variability in conditions.

Evaluation

The agents were evaluated from both an operational and a safety perspective. The operational analysis was conducted by collecting and analyzing the vehicle trajectory files from each run with and without the agent active. Safety was analyzed using the Surrogate Safety Assessment Model (SSAM) developed by the Federal Highway Administration. SSAM identifies conflicts in the traffic stream as modeled by VISSIM and reports a number of measures from which safety can be evaluated. The results of the operational and safety analyses are discussed in detail in the following sections. The chapter concludes with a discussion of agent scalability. A slightly more complex network was developed to evaluate the performance of the agents in situations beyond the specific conditions for which they had been trained.

OPERATIONS ASSESSMENT

The impact of the agents on vehicle dynamics and operational measures was assessed first. This section first presents a qualitative description of the observed agent behavior in VISSIM. Next it presents an analysis of objective vehicle dynamics metrics to illustrate differences in driver behavior between the agents and the Wiedemann model using the simple test network.

Agent Behavior in VISSIM

The models for the car and truck agents were tested in VISSIM and evaluated for their effects on traffic operations and mobility. An overview of the observed agent behavior on this simple one-lane VISSIM test network appears below.

Car Agents

- **Agent 101A**: The agent enters the network at a higher speed than the leader. Upon closing with the leader, the agent decelerates, swerves slightly, and passes (overruns) the leader.

- **Agent 106A**: The agent enters the network at a higher speed than the leader. Upon closing with the leader, the agent overruns the leader.

- **Agent 113A**: The agent enters the network at a higher speed than the leader. Upon closing with the leader, the agent decelerates significantly. Depending on the individual agent's reaction, the agent either comes to a complete stop prior to colliding with the leader or overruns the leader.

- **Agent 117A:** The agent enters the network at a higher speed than the leader. Upon closing with the leader, the agent decelerates significantly. Depending on the individual agent's reaction, the agent either comes to a complete stop prior to colliding with the leader or overruns the leader.

- **Agent 201A:** The agent enters the network at a higher speed than the leader. Upon closing with leader, the agent swerves slightly and decelerates sharply behind the leader. The agent usually comes to complete stop behind leader but occasionally overruns the leader if it cannot brake in time.

- **Agent 307A**: The leader travels faster than the agent, and the gap between vehicles expands during simulation. No significant interaction is observed between the vehicles.

- **Agent 311A**: The agent enters the network at a higher speed than the leader. Several behaviors were observed when the agent and the leader interact. If the agent cannot stop in time, it sometimes overruns the leader. Otherwise, the agent decelerates quickly and swerves slightly as it brakes behind the leader. After dwelling for some time, the agent accelerates sharply until it catches up with the leader again, at which point it again decelerates and swerves off the road, sometimes by a significant margin.

- **Agent 317A**: The agent enters the network at a higher speed than the leader. When it catches up to the leader, it consistently overruns the leader.

- **Agent 415A**: Several behaviors were observed for this agent. The agent enters the network at a higher speed than the leader. When it catches up to the leader, it usually decelerates and swerves, but ultimately it overruns the leader. Sometimes the agent will catch up to the leader and maintain a safe following distance, and other times it decelerates quickly and leaves the roadway.

- **Agent 460A:** The agent enters the network at a higher speed than the leader. When it catches up to the leader, it consistently overruns the leader.

- **Agent 462A:** The agent enters the network at a higher speed than the leader. When it catches up to the leader, it consistently overruns the leader.

Truck Agents

- **Agent 6:** The agent enters the network at a higher speed than the leader. When it catches up to the leader, it swerves off the right side of the road to avoid the leader, passing it in the process.

- **Agent 23:** The agent enters the network at a higher speed than the leader. The agent swerves off the right side of the road and returns back to the road, coming to a complete stop perpendicular to the roadway. The distance of the roadway departure changes based on the simulation random number seed.

- **Agent 38:** Several behaviors were observed for this agent. The agent enters the network at a higher speed than the leader. When it catches up to the leader, it usually decelerates and swerves, passes the agent, and returns to the roadway. However, sometimes the agent swerves and does not pass the leader, and instead continues to swerve farther and farther off the roadway, maintaining a distance of approximately 50 feet from the roadway.

- **Agent 51:** The agent enters the network at a higher speed than the leader. The agent swerves slightly to the left but ultimately overruns the leader.

- **Agent 63:** The agent enters the network at a higher speed than the leader. The agent swerves slightly to the left but ultimately overruns the leader.

- **Agent 64:** Several behaviors were observed for this agent. The agent sometimes enters the network at a higher speed than the leader, decelerates quickly, and stops for a short period, then accelerates, catches the leader, and overruns it. Other times, the agent stops suddenly upon entering the network and after a few seconds accelerates. Occasionally it catches and overruns the leader, but other times the simulation ends before the agent catches the lead vehicle.

- **Agent 79:** The agent enters the network at a higher speed than the leader. When it catches up to the leader, the agent decelerates quickly and consistently overruns the leader, eventually accelerating to its original speed.

- **Agent 85:** The agent enters the network at a higher speed than the leader. The agent swerves slightly to the right and ultimately overruns the leader.

- **Agent 249:** The agent enters the network at a higher speed than the leader. When it catches up to the leader, it consistently overruns the leader.

A visual inspection of the simulations demonstrates some unexpected behavior in certain agents. Some agents appear to overrun the lead vehicle in some scenarios (Agents 101A, 106A, 113A, 201A, 311A, 317A, 415A, 460A, 462A, 51, 63, 64, 79, 85, and 249), and some agents show multiple behaviors based on the simulation seed (Agents 113A, 117A, 201A, 311A, 415A, 38, and 64).

To show the agents' behavior visually, two plots were made for each simulation run of each agent. The set of figures shows the time-space diagrams of a lead vehicle, an agent, and a following vehicle using the Wiedemann model as implemented in VISSIM. The figures can be found in Appendix [A]. These figures show that most agents enter the network at a higher rate of speed than the non-agent following vehicles. With the exception of agents 117A, 307A, 51, 64, and 85, agents reached the lead vehicle before the non-agent, often displaying little change in speed when closing in. The most significant difference between the agents and the non-agents was the inability of the agents to consistently follow a lead vehicle. While some agents noticeably swerved around the lead vehicle, no agents were able to stop without appearing to overrun the lead vehicle in simulation in at least a few of the 20 simulation runs. In contrast, the non-agents began following the lead vehicles and never overtook them in any simulation run. As an example of this characteristic, Figure 146 demonstrates the differences in passing and following behavior of Agent 311A and its Wiedemann model counterpart across 20 simulation runs. In the figure, although Agent 311A can occasionally react and decelerate in time to avoid collision, on some simulation runs the lead vehicle is overrun. This is a manifestation of how the agents have now been explicitly trained to behave in a manner that allows safety critical events to occur.

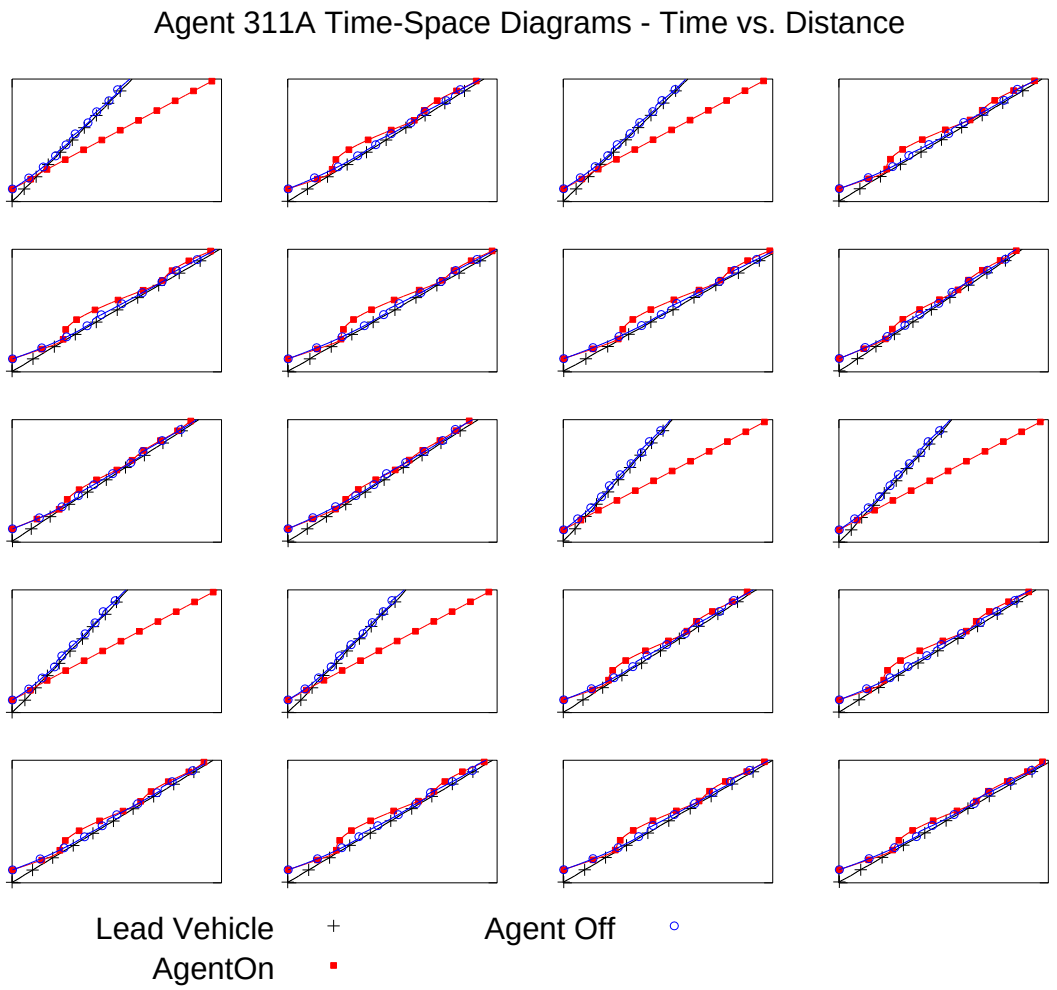

Figure 146. Graphs. Time-space graphs of each simulation run of Agent 311A, both with the agent on and the agent off.

Figure 147 and Figure 148 show plots of headway versus difference in speed between the agent and the lead vehicle. The figures show a scenario with the agent driver model and one without. In these figures, when an agent's headway is at the highest point in the chart, it indicates that the agent vehicle has passed the lead vehicle. Plots for each of the agents and their corresponding Wiedemann cases are found in Appendix [B]. The figures show most of the agents initially in what appears to be the "closing in" zone. As the agent approaches the lead vehicle, the agent may decelerate at the last minute, as seen in Figure 147 with Agent 311A, or may continue into the emergency regime with little or no reaction, as seen in Figure 148 with Agent 249. Other agents appear to follow the lead vehicle, creating the circular plot line similar to Wiedemann model vehicles. This generally reflects the agent behavior described earlier, where some agents do not appear to react to the lead vehicle by slowing, but instead by swerving or driving as before until they have passed the lead vehicle, while other agents attempt to decelerate at varying rates.

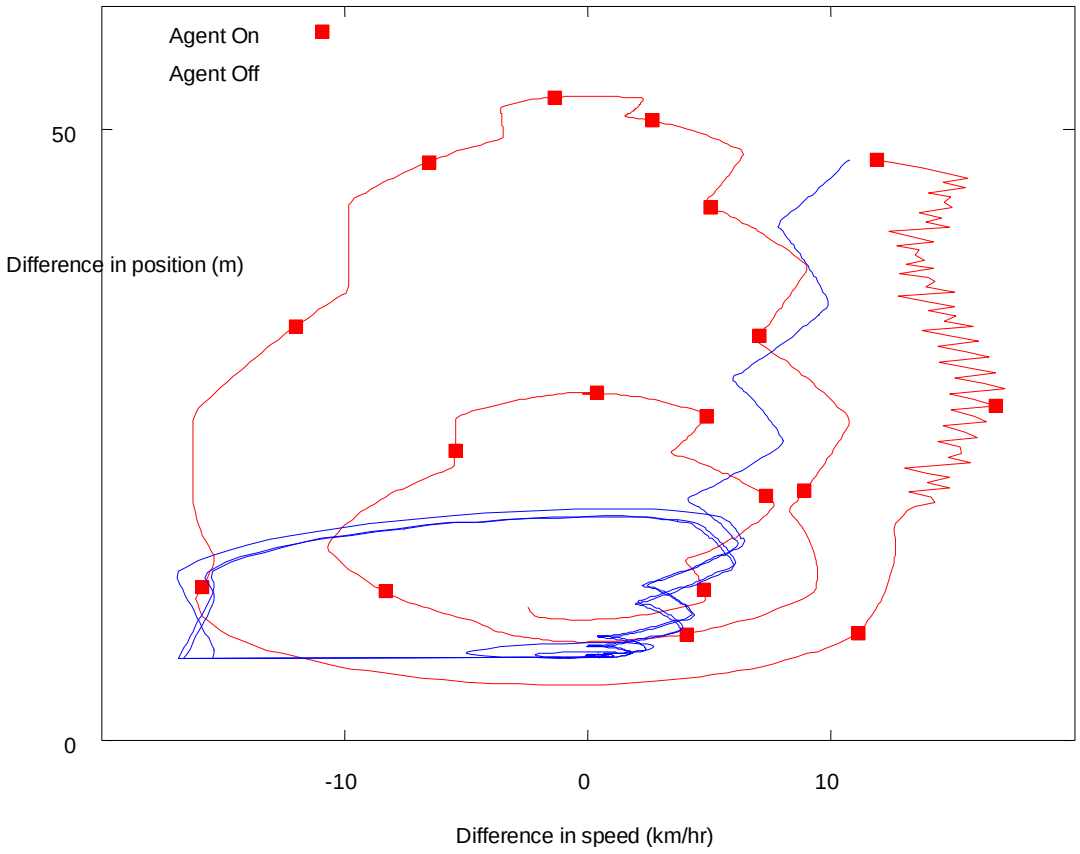

Figure 147. Chart. Wiedemann diagram of a single simulation run of Agent 311A.

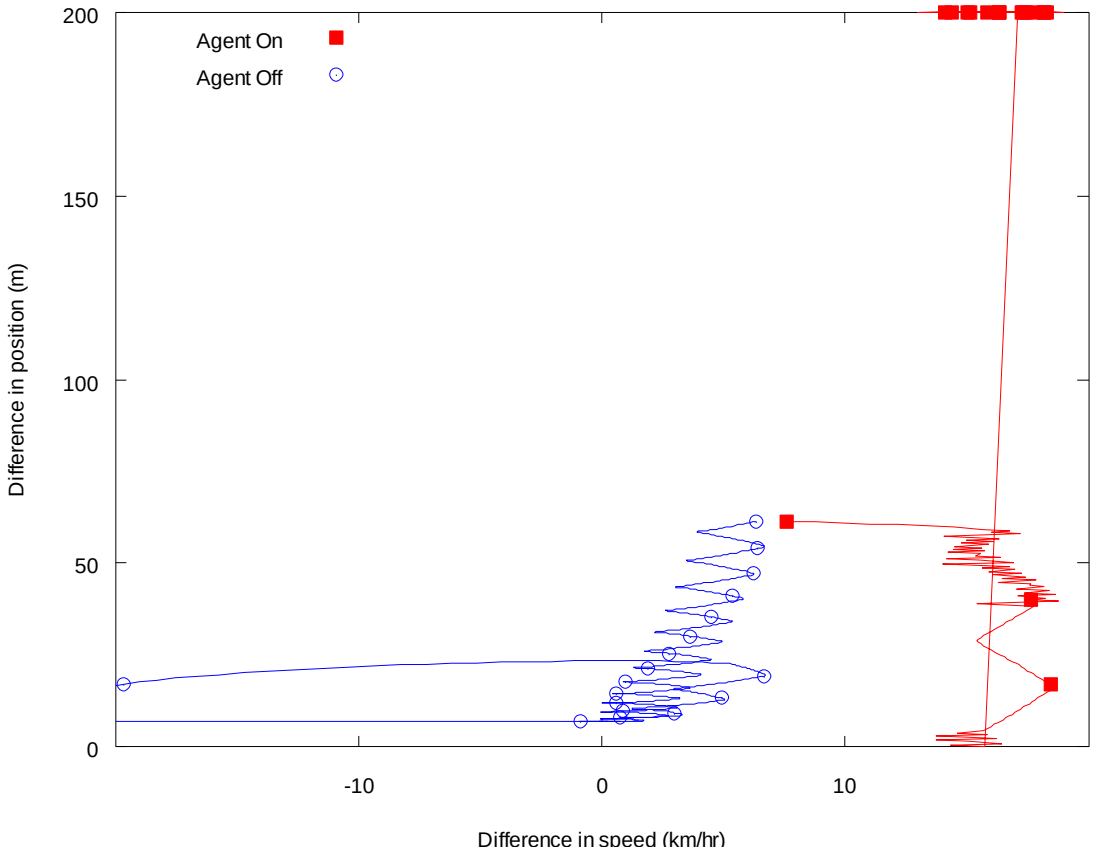

Figure 148. Chart. Wiedemann diagram of a single simulation run of Agent 249.

Quantitative Assessment of Vehicle Behavior

Because the model was restricted to a single-lane, one-way network by design, and because the agents themselves could not transition from the agent-specific driving model to the general Wiedemann model, the agent could not be tested in a traditional multi-lane lane network with interchanges and intersections. Therefore more traditional macroscopic metrics such as travel time and delay could not be used, simply because the test networks consisted of two vehicles in a single interaction.

Several metrics were used to better understand the behavior of the agent:

- **Minimum following distance (m):** The minimum distance between the lead vehicle and the agent. A value of 0.0 indicates that the agent collides with the lead vehicle or passes it while traveling off of the roadway link.

- **Minimum acceleration (m/s^2):** The minimum acceleration rate experienced by the agent in a single run. Several agents had unusually high deceleration rates of greater

than 50 m/s^2 when first inserted. These values were considered to be arbitrary simulation error and were ignored in the analysis.

- **Average acceleration (m/s^2)**: The average acceleration rate of an agent over an entire simulation run.

- **Average of absolute acceleration (m/s^2):** The average of the absolute value of an agent's acceleration over an entire simulation run. High values indicate frequent and/or drastic changes in speed.

- **Average speed (km/hr):** The average speed of an agent.

Each metric was evaluated with the agent (i.e., "Agent On") and in a base case using the Wiedemann model as implemented in VISSIM (i.e., "Agent Off"). Table 37 shows the average value for each metric over 20 simulation runs of the car agents. Table 38 shows the same information for the truck agents.

In general, most agents eventually passed the lead vehicle, as demonstrated by the minimum following distance of 0 m, while the lead vehicle was never passed in the base cases. Because a one-lane network was simulated, the agent vehicle either departed the roadway to pass the lead vehicle or drove through the lead vehicle and passed through them (a collision event). Most agents (with the exceptions of agents 51, 63, and 64) were able to eliminate the unrealistically high decelerations seen in the base cases when collision was imminent. However, because many of the agents appeared to overrun the lead vehicles, these more realistic decelerations may not in fact demonstrate more realistic behavior overall. Average acceleration was much lower in the base cases than for the agents, which is likely due to the very high decelerations used by the following vehicle to avoid collisions. Average speed was generally much lower for the base cases, which is likely a result of the base case following vehicles reducing their speeds to stay behind the lead vehicles for the bulk of the simulation. In comparison, the agents often passed the lead vehicle and continued at their desired speeds.

Table 37. Mobility performance metrics for car agents, average values over 20 simulation runs.

Agent	Agent Status	Minimum Following Distance (m)	Minimum Acceleration (m/s^2)	Average Acceleration (m/s^2)	Average Absolute Acceleration (m/s^2)	Average Speed (km/hr)
101A	On	0.00	-6.46	0.00	2.62	36.68
	Off	6.27	-100.00	-0.92	1.32	24.38
	Δ	**-6.27**	**93.54**	**0.92**	**1.30**	**12.31**
106A	On	0.00	-8.18	0.01	3.29	13.74
	Off	6.26	-100.00	-3.28	3.81	5.99
	Δ	**-6.26**	**91.82**	**3.29**	**-0.52**	**7.75**
113A	On	2.08	-7.59	-0.44	2.51	8.56
	Off	6.26	-100.00	-3.28	3.81	5.99
	Δ	**-4.18**	**92.41**	**2.84**	**-1.29**	2.57
117A	On	2.29	-9.36	0.01	2.55	5.61
	Off	6.26	-100.00	-3.28	3.81	5.99
	Δ	**-3.98**	**90.64**	**3.29**	**-1.26**	-0.38
201A	On	4.10	-6.78	-0.01	0.81	6.67
	Off	6.26	-100.00	-3.31	3.84	5.98
	Δ	**-2.17**	**93.22**	**3.30**	**-3.02**	0.69
307A	On	14.52	-8.53	0.01	3.59	3.62
	Off	6.26	-100.00	-3.02	3.57	5.93
	Δ	**8.26**	**91.48**	**3.03**	0.02	**-2.31**
311A	On	4.02	-6.59	-0.03	1.20	23.06
	Off	6.26	-100.00	-1.15	1.56	19.01
	Δ	**-2.25**	**93.41**	**1.12**	-0.36	**4.05**
317A	On	0.00	-7.20	0.01	2.74	32.95
	Off	6.26	-100.00	-1.15	1.56	19.01
	Δ	**-6.26**	**92.80**	**1.16**	**1.18**	**13.94**
415A	On	2.38	-6.78	0.00	2.58	29.92
	Off	7.50	-91.82	-0.46	0.72	28.07
	Δ	**-5.12**	**85.04**	**0.45**	**1.85**	**1.86**
460A	On	0.00	-6.46	0.00	2.62	36.68
	Off	6.27	-100.00	-0.92	1.32	24.38
	Δ	**-6.27**	**93.54**	**0.92**	**1.30**	**12.31**
462A	On	0.00	-7.32	-0.09	2.91	19.10
	Off	6.26	-100.00	-1.27	1.64	14.15
	Δ	**-6.26**	**92.68**	**1.17**	**1.27**	**4.94**

Values shown in bold are significant at $\alpha = 0.05$

Table 38. Mobility performance metrics for truck agents, average values over 20 simulation runs.

Agent	Agent Status	Minimum Following Distance (m)	Minimum Acceleration (m/s^2)	Average Acceleration (m/s^2)	Average Absolute Acceleration (m/s^2)	Average Speed (km/hr)
6	On	0.00	-6.54	0.02	2.31	71.67
	Off	6.27	-99.49	-0.49	0.85	48.28
	Δ	**-6.27**	**92.95**	**0.51**	**1.46**	**23.38**
23	On	0.00	-7.08	-2.00	2.46	13.11
	Off	13.08	-90.66	-0.33	0.69	56.58
	Δ	**-13.08**	**83.58**	**-1.66**	**1.78**	**-43.48**
38	On	0.82	-4.11	0.01	1.00	91.35
	Off	6.27	-98.26	-0.24	0.88	86.15
	Δ	**-5.45**	**94.15**	**0.25**	0.12	5.20
51	On	0.00	-21.49	0.02	3.71	18.97
	Off	6.27	-100.00	-0.97	1.24	16.40
	Δ	**-6.27**	**78.51**	**0.99**	**2.46**	**2.56**
63	On	0.00	-81.87	0.00	2.94	53.06
	Off	6.27	-100.00	-0.44	0.79	43.27
	Δ	**-6.27**	**18.13**	**0.44**	**2.14**	**9.79**
64	On	0.00	-100.00	-10.81	12.30	83.36
	Off	6.26	-100.00	-0.34	0.92	79.84
	Δ	**-6.26**	**0.00**	**-10.47**	**11.38**	**3.52**
79	On	0.00	-8.16	0.06	2.62	76.18
	Off	6.27	-100.00	-0.68	1.23	42.65
	Δ	**-6.27**	**91.84**	**0.74**	**1.39**	**33.53**
85	On	0.00	-6.73	0.05	2.44	68.52
	Off	6.26	-100.00	-0.55	1.04	57.87
	Δ	**-6.26**	**93.27**	**0.60**	**1.40**	**10.65**
249	On	0.00	-9.09	0.03	3.50	38.17
	Off	6.27	-96.48	-0.62	0.88	24.50
	Δ	**-6.27**	**87.39**	**0.64**	**2.63**	**13.67**

Values shown in bold are significant at $\alpha = 0.05$

For a visual comparison of every simulation run of each agent and its corresponding base case, refer to Appendices [A and B].

Summary of Operations Analysis

The qualitative and quantitative analysis on the simple linear VISSIM network revealed the following findings:

- Most agents were able to enter the emergency regime, where an agent's speed is greater than the lead vehicle's, yet headway is small. Vehicles following the Wiedemann model as implemented in VISSIM are unable to enter the emergency regime, and instead vehicles will be forced to decelerate at unrealistic rates to avoid collisions.

- Almost all agents were able to eliminate the improbably high deceleration rates used by vehicles following the Wiedemann model as implemented in VISSIM. The agents instead ran over, swerved, or collided with leading vehicles. The vast majority of the agents' deceleration rates to avoid collisions were realistic.

- Although agents experienced realistic deceleration rates, in many cases they did not behave realistically across all simulation runs. Some agents stopped in the roadway even when a crash was not imminent. Other agents reacted to a leading vehicle upon the initial encounter, but after accelerating and encountering the vehicle a second time in a single simulation run, ran over the vehicle without noticeable reaction. The agents' behavior indicates that although each agent is programmed to respond a certain way to a certain scenario, the definition of that scenario may be some what narrow that changes in the random seed fail to re-create it. The definition of the scenario may need to be broadened or better defined so that vehicles can behave in a more consistent way.

- The agents seemed unable to react to the lead vehicles unless it recognized the scenario for which it had been trained. This led to some unusual behavior of the agents, particularly after an encounter with the lead vehicle. If the agent appeared to run over the lead vehicle, and therefore crash, the agent then continued past the lead vehicle while maintaining its desired speed. Ideally, the agent would be able to transfer into and out of a traditional car-following model such as the Wiedemann model. If the agent could change models when it no longer recognized the scenario for which it has been trained, it would be much simpler to integrate these agents into a typical multi-vehicle network and measure the macroscopic properties of the agents.

SAFETY ASSESSMENT

A series of safety surrogate measures were analyzed to compare the behaviors of the agents and the VISSIM Wiedemann car-following model. This provided an indication as to whether the agents were creating significantly more safety critical events than the base VISSIM car-following model. The Surrogate Safety Assessment Model (SSAM), version 2.1.6, was used to examine the influence of the agents and the Wiedemann model on safety surrogate measures.

The SSAM can post-process vehicle trajectory files generated by VISSIM and provides a mechanism to compare estimated safety surrogate measures with and without the various agents.

The SSAM identifies conflict events based on time to collision (TTC) thresholds. TTC is defined as the elapsed time before two vehicles collide if they maintain their current speed and heading. In this analysis, conflicts were identified using two TTC thresholds: 1.5 s and 2.5 s. A TTC of 2.5 s is a relatively high threshold for triggering conflicts, but it was set high to ensure that some conflict events would be triggered when the Wiedemann car-following model was active. The 2.5-s value was based on the assumed perception-reaction time for stopping sight distance in the American Association of State Highway and Transportation Officials' *Policy on the Geometric Design of Highways and Streets*.

When the conflict TTC threshold is exceeded, the SSAM generates a series of safety surrogate measures for each conflict event. Specific measures examined in this research include:

- **TTC** is the minimum time to collision between the agent and leading vehicle during the conflict. If a conflict occurs and has a TTC of 0, it means that a collision will occur.

- **PET** is the minimum post encroachment time observed during the conflict. This represents the elapsed time between when the leading vehicle occupies a point and when the agent arrives at the same position. A PET of zero indicates a collision has occurred.

- **MaxS** is the maximum speed of either vehicle throughout the conflict.

- **DeltaS** is the difference in speeds observed at the time step when the minimum TTC occurs. If the minimum TTC equals zero, this would be the relative difference in speeds at collision.

- **DR** is the initial deceleration rate of the following vehicle (agent) during the conflict. If the following vehicle brakes, this is the first negative acceleration value observed during the conflict. If the follower does not brake, then this is the lowest acceleration value observed during the conflict.

- **MaxD** is the maximum deceleration of the following vehicle (agent). Negative values indicate deceleration, and positive values indicate acceleration.

- **MaxDeltaV** is the maximum change in velocity between when a conflict first begins and immediately post-crash, assuming a hypothetical inelastic crash between vehicles. This is a surrogate for conflict severity.

All of these measures are tabulated when a conflict has been identified. In other words, if a conflict does not occur, then zeros are not included in the averages.

Each agent scenario was examined with both the agent active and with the agent inactive (Wiedemann car-following active). Twenty replications were conducted for each scenario and each agent status. This resulted in a total of 800 simulation trajectories [20 agents × 20

replications × 2 levels of agent status (agent active or Wiedemann active)] that were evaluated using SSAM. These were evaluated using the 1.5-s and 2.5-s TTC values for conflict identification.

General Observations on Agent Behavior and Impact on Safety Surrogates

The reactions of agents were observed during the course of each simulation to identify responses that could impact the safety surrogate measures being evaluated. First, even though the simulation duration was only 100 s, several agents would produce multiple safety critical events over the course of the simulation. For example, an agent sometimes approached the leader and braked sharply, coming to a complete stop. After dwelling at a stop for a few seconds, the agent accelerated sharply until it caught up to the leader again. At that point, it braked sharply a second time. This sort of behavior could produce multiple conflict events in one simulation replication, but the behavior of the agent was not necessarily realistic because it just repeated the same unsafe behavior multiple times.

A second behavior that was commonly seen in the agent simulations was the agent overrunning (traveling completely over) the leading vehicle. In this case, the safety surrogates would show a value of 0 seconds for the TTC and PET since a collision would occur. After passing over the leader, however, both the leader and agent proceeded as if no collision had occurred. The behavior after the collision was not necessarily realistic in many cases.

Finally, some agents would swerve around the lead vehicle, sometimes actually leaving the roadway link. While this behavior is certainly a conflict, it may not show up strongly in some of the safety surrogate measures. When a vehicle swerves around the leader, it may not have a short enough TTC to trigger a conflict identification because it has moved onto a path where it will not conflict with the lead vehicle. This would mean that critical safety surrogate measures for these agents may be underreported.

The safety surrogates provided by SSAM provide some general indications of performance of the agents, but there are definitely cases where they may not fully represent the agent behavior. As a result, many of the safety surrogates should be viewed for relative changes between the agent and the Wiedemann model, as opposed to absolute values of each measure.

Evaluation of Car Agents

Table 39 summarizes the safety surrogate measures for the car agents, using a 1.5-s TTC to define a conflict. In addition to the surrogate measures discussed earlier, the average number of conflicts per run is also listed. All conflicts evaluated were rear-end conflicts because the agents were developed based on rear-end conflict cases. No other conflict types were produced for any agent. The Wiedemann car-following model (noted as Agent Off in Table 39), did not produce any conflicts when the TTC threshold was set to 1.5 s. This shows that the Wiedemann model does not permit close following behaviors by default, and thus uses the assumption of safe driving behavior.

With the exception of agent 307A, all agents produced conflicts at least occasionally. Agent 307A entered the network at a lower speed than the leading vehicle, meaning the agent never caught up to the leader to cause a conflict. The remaining agents produced a mean number of conflicts of between 0.55 and 1.75 (11 and 35 conflicts per 20 runs, respectively). Mean TTCs and PETs were 0 s for 5 of 11 car agents: 101A, 117A, 317A, 415A, and 460A. This means that all of these agents produced collisions between the agent and the leading vehicle.

Table 39. Safety surrogate measure comparison using 1.5-s TTC – car agents.

Agent	Agent Status	Mean # of Conflicts	Mean TTC (s)	Mean PET (s)	Mean MaxS (m/s)	Mean DeltaS (m/s)	Mean DR (m/s^2)	Mean Max D (m/s^2)	Mean MaxDeltaV (m/s)
101A	On	1.0	0	0	10.44	3.95	-0.13	-0.13	1.98
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
106A	On	1.75	0.62	0.43	4.11	2.42	-2.58	-3.01	1.21
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.75	n/a	n/a	n/a	n/a	n/a	n/a	n/a
113A	On	0.80	0.23	0.12	3.93	2.31	-0.24	-1.24	1.16
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.80	n/a	n/a	n/a	n/a	n/a	n/a	n/a
117A	On	0.65	0	0	2.36	0.72	-0.05	-0.11	0.36
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.65	n/a	n/a	n/a	n/a	n/a	n/a	n/a
201A	On	0.55	0.46	0.11	2.55	0.77	0.16	-0.45	0.39
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.55	n/a	n/a	n/a	n/a	n/a	n/a	n/a
307A	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
311A	On	0.60	0.46	0.09	8.11	3.21	-0.72	-1.72	1.61
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.60	n/a	n/a	n/a	n/a	n/a	n/a	n/a
317A	On	1.0	0	0	9.44	4.13	-0.15	-0.16	2.03
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
415A	On	0.85	0	0	8.67	1.15	-0.38	-2.83	0.57
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.85	n/a	n/a	n/a	n/a	n/a	n/a	n/a
460A	On	1.0	0	0	10.44	3.95	-0.13	-0.13	1.98
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
462A	On	1.60	0.60	0.24	6.04	2.25	-2.50	-3.55	1.12
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.60	n/a	n/a	n/a	n/a	n/a	n/a	n/a

n/a: Measure or comparison not generated because no conflicts were detected.

Table 40 summarizes the safety surrogate measures for the car agents using a TTC threshold of 2.5 s to define a conflict. When the TTC threshold was raised to 2.5 s, the Wiedemann model sometimes produced conflicts for some agent scenarios. The scenarios for agents 106A, 113A, 117A, 201A, 307A, and 462A (6 of 11 scenarios) then had some conflicts detected when the Wiedemann car-following model was used. Once again, all agents except for agent 307A produced some conflicts, with the mean number of conflicts ranging from 0.7 to 1.95 (14 to 39 conflicts per 20 runs). Mean TTC and PET values of the agents increased because conflicts at higher TTCs were then included in the summary.

For the five cases where both the agent and the Wiedemann model produced conflicts, it was possible to examine differences between the surrogate measures using t-tests. Values shown in bold in Table 40 were significant at $\alpha = 0.05$. As expected, the agents produced more safety

critical conflicts, and the surrogate measures generally showed that the conflicts were more severe than when the Wiedemann model was used.

Table 40. Safety surrogate measure comparison using 2.5-s TTC – car agents.

Agent	Agent Status	Mean # of Conflicts	Mean TTC (s)	Mean PET (s)	Mean MaxS (m/s)	Mean DeltaS (m/s)	Mean DR (m/s^2)	Mean Max D (m/s^2)	Mean MaxDeltaV (m/s)
101A	On	1.0	0	0	10.45	4.06	-0.08	-0.15	2.03
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
106A	On	1.95	1.21	0.82	4.10	2.41	-2.39	-3.57	1.20
	Off	0.20	2.20	1.47	2.73	1.27	-0.12	-4.49	0.63
	Δ	**1.75**	**-0.995**	**-0.657**	**1.373**	**1.141**	**-2.277**	0.923	0.570
113A	On	1.45	1.37	0.98	4.10	2.46	-3.20	-4.29	1.23
	Off	0.20	2.20	1.47	2.73	1.27	-0.12	-4.49	0.63
	Δ	**1.25**	**-0.828**	**-0.499**	**1.370**	**1.188**	**-3.088**	0.205	0.594
117A	On	0.70	0.18	0.04	2.39	0.72	-0.03	-0.11	0.36
	Off	0.20	2.20	1.47	2.73	1.27	-0.12	-4.49	0.63
	Δ	**0.50**	**-2.021**	**-1.432**	-0.335	-0.552	0.091	**4.380**	-0.276
201A	On	0.80	1.12	0.31	2.53	1.02	0.13	-0.67	0.51
	Off	0.20	2.20	1.47	2.73	1.27	-0.12	-4.49	0.63
	Δ	**0.60**	**-1.081**	**-1.169**	-0.196	-0.251	0.249	**3.819**	-0.125
307A	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0.30	2.25	1.62	2.43	1.08	-0.21	-18.70	0.54
	Δ	-0.30	n/a	n/a	n/a	n/a	n/a	n/a	n/a
311A	On	0.80	0.88	0.21	8.41	3.68	-0.48	-1.70	1.84
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.80	n/a	n/a	n/a	n/a	n/a	n/a	n/a
317A	On	1.05	0.22	0.13	9.44	4.38	-1.35	-1.80	2.19
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a
415A	On	0.95	0.22	0.03	8.70	1.22	-0.38	-2.51	0.61
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.95	n/a	n/a	n/a	n/a	n/a	n/a	n/a
460A	On	1.0	0	0	10.45	4.06	-0.08	-0.15	2.03
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
462A	On	1.80	1.11	0.45	6.06	2.30	-4.16	-5.86	1.15
	Off	0.25	2.20	0.66	4.11	1.22	-0.33	-0.80	0.61
	Δ	**1.55**	**-1.092**	**-0.207**	**1.943**	**1.087**	**-3.828**	**-5.066**	0.543

n/a: MOE or comparison not generated since no conflicts were detected
Values shown in bold are significant at $\alpha = 0.05$.

Evaluation of Truck Agents

Table 41 summarizes the evaluation of the truck agent scenarios when the TTC threshold for conflicts was set to 1.5 s. Like the car agent scenarios, the Wiedemann car-following model never produced any conflicts when the TTC threshold was set to 1.5 s. When the truck agents were active, seven of nine agents produced conflicts at least occasionally. For those agents that produced conflicts, the mean number of conflicts ranged from 0.8 to 1.60 conflicts per run (16 to

32 conflicts per 20 runs). Agents 51, 63, 64, 79, 85, and 239 all had a mean TTC and PET of 0, meaning that they produced collisions in each conflict generated.

In the case of Agents 6 and 23, no conflicts were produced. Agent 6 consistently swerved around the lead vehicle by a significant margin, leaving the network and re-entering the link in front of the leader. While this was definitely a significant safety critical event, the agent never triggered the 1.5-s TTC threshold to define a conflict, so no safety surrogate measures were generated by SSAM. A similar situation happened with Agent 23. In this case, the vehicle decelerated and swerved off of the roadway link before it could get into a conflict with the lead vehicle. Thus, even though a conflict event was not triggered for either agent, the agent behavior was definitely safety critical.

Table 41. Safety surrogate measure comparison using 1.5-s TTC – truck agents.

Agent	Agent Status	Mean # of Conflicts	Mean TTC (s)	Mean PET (s)	Mean MaxS (m/s)	Mean DeltaS (m/s)	Mean DR (m/s^2)	Mean Max D (m/s^2)	Mean MaxDeltaV (m/s)
6	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
23	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
38	On	0.80	0.55	0.06	30.07	6.46	-0.62	-3.18	3.23
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.80	n/a	n/a	n/a	n/a	n/a	n/a	n/a
51	On	1.10	0	0	5.95	1.82	0.48	0.48	0.91
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
63	On	1.60	0	0	16.31	3.79	1.45	1.45	1.90
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.60	n/a	n/a	n/a	n/a	n/a	n/a	n/a
64	On	1.05	0	0	27.91	8.13	-1.16	-2.55	4.07
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a
79	On	1.0	0	0	21.52	9.41	-2.77	-5.09	4.71
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
85	On	1.05	0.03	0	19.51	1.93	-0.03	-0.28	0.96
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.05	n/a	n/a	n/a	n/a	n/a	n/a	n/a
249	On	1.0	0	0	10.96	4.52	-0.01	-0.01	2.26
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a

n/a: MOE or comparison not generated because no conflicts were detected.

Table 42 summarizes the safety surrogate measures for the truck agents when the TTC threshold for a conflict was set to 2.5 s. Once again, Agents 6 and 23 did not produce any conflicts because they left the roadway and did not interact with the leader. The remaining agents produced an average number of rear-end conflicts ranging from 0.95 to 1.60 (19 to 32 conflicts per 20 runs). Using the 2.5-s TTC threshold to define a conflict also created conflicts

when the Wiedemann car-following model was used in scenarios for Agents 6, 23, 51, 64, and 85 (5 of 9 cases).

Agents 51, 64, and 85 were the only cases where both the agent and the Wiedemann model produced conflicts. The t-tests in those cases showed that the agents tended to produce significantly more conflicts than when the Wiedemann model was used, and the conflicts produced were more severe. This was as expected.

Table 42. Safety surrogate measure comparison using 2.5-s TTC – truck agents.

Agent	Agent Status	Mean # of Conflicts	Mean TTC (s)	Mean PET (s)	Mean MaxS (m/s)	Mean DeltaS (m/s)	Mean DR (m/s^2)	Mean Max D (m/s^2)	Mean MaxDeltaV (m/s)
6	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0.10	2.25	0.20	12.18	1.16	-0.49	-1.05	0.58
	Δ	-0.10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
23	On	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Off	0.10	2.50	0.20	14.34	0.90	-0.58	-0.78	0.45
	Δ	-0.10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
38	On	0.95	0.78	0.11	30.19	6.27	-1.63	-3.57	3.14
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	0.95	n/a	n/a	n/a	n/a	n/a	n/a	n/a
51	On	1.30	0.38	0.18	5.89	1.78	-0.39	-0.54	0.89
	Off	0.10	2.40	0.60	4.29	0.99	-0.12	-2.76	0.49
	Δ	**1.20**	**-2.023**	-0.423	1.594	**0.793**	-0.277	**2.220**	**0.396**
63	On	1.60	0	0	16.31	3.51	1.44	1.44	1.75
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.60	n/a	n/a	n/a	n/a	n/a	n/a	n/a
64	On	1.05	0	0	28.18	8.35	-1.08	-2.64	4.18
	Off	0.15	2.47	0.20	20.45	1.55	-0.53	-0.56	0.77
	Δ	**0.90**	**-2.467**	**-0.20**	**7.730**	**6.808**	**-0.554**	**-2.077**	**3.404**
79	On	1.0	0	0	21.52	8.62	-2.45	-5.20	4.31
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
85	On	1.05	0.03	0	19.56	1.97	0.03	-0.30	0.98
	Off	0.10	2.35	0.30	16.34	2.09	-0.64	-0.72	1.05
	Δ	**0.95**	**-2.317**	**-0.295**	**3.213**	-0.122	**0.673**	0.418	-0.061
249	On	1.0	0	0	10.96	4.53	0	-0.01	2.26
	Off	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Δ	1.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a

n/a: MOE or comparison not generated because no conflicts were detected.
Values shown in bold are significant at $\alpha = 0.05$.

Summary of Safety Surrogate Analysis

The safety surrogate measure analysis on the simple linear VISSIM network revealed the following findings:

- As expected, the agents produced many more safety critical events than when VISSIM used the Wiedemann-car-following model to control the following vehicle. The

Wiedemann car-following model never produced any conflicts when the TTC threshold for a conflict was set at 1.5 s, regardless of the scenario being evaluated.

- Several of the agents were not well suited to analysis using SSAM. Agents that departed the roadway by a significant margin were not properly classified as having caused safety critical events.

- When the TTC threshold for conflicts was increased to 2.5 s, the Wiedemann model sometimes produced conflicts. However, the number and severity of conflicts with the Wiedemann model were not as severe as when the agent was active.

- Reactions of the agents following the safety critical event were sometimes not realistic. Agents would repeat behaviors that created safety critical events multiple times, or would proceed following a collision as if nothing had happened. Before the agents can be implemented in a more complex model, behaviors following critical events will need to be re-examined for the agents. In effect, non-collision events would need to reset to the traditional Wiedemann model, and collision events would need to be modeled by having both vehicles come to a stop after the event.

EVALUATION OF AGENTS ON MORE COMPLEX NETWORKS

A simple VISSIM test network was developed to evaluate agent performance on a network more complex than the single-lane, unidirectional link evaluated earlier. The intent of this analysis was to determine the potential impact of agent behaviors on macroscopic traffic characteristics, for example by examining the impact of safety critical events on traffic flow and travel time reliability. The test network consisted of a 4-mile stretch of freeway with two interchanges. One of the interchanges in the example network is shown in Figure 149 below. Traffic volumes were set so that transient congestion formed at freeway on-ramps and there was turbulence due to lane changing at the off-ramps. The intent was to evaluate the impact of different agent types and penetrations on operational measures on this network. Since the ramp termini were near capacity using the Wiedemann car-following model, the expectation was that the agents would create more significant congestion on the network.

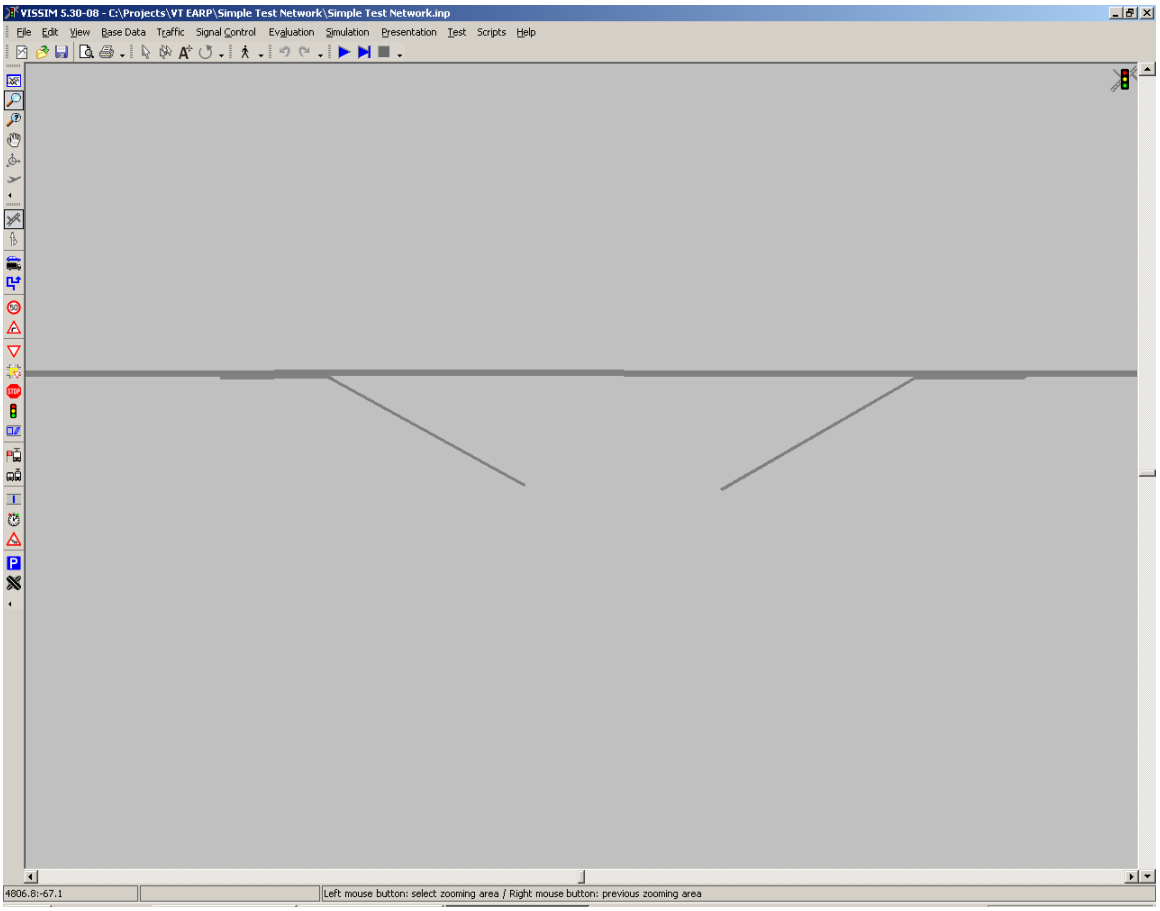

Figure 149. Screenshot. Example of interchange on test network.

Initial evaluations on this network revealed several problems with directly implementing the agents. First, many of the problems that were manifested on the simple linear network continued to occur on this network. Agents would overrun leading vehicles, and once an agent was activated it seemed not to revert back to normal driving behavior. The net result of these issues was that traffic operations on this network did not appear to be realistic because vehicle interactions with the agents were not reasonable. As a result, a detailed evaluation of the agents on this network could not be conducted.

The root of the problem appears to be twofold. First, agents cannot adapt to situations for which they were not trained, which limited their ability to be applied in a more complex network. Secondly, the agents never reset to normal car-following behavior after a safety critical event. Once the agent was activated, it remained activated. The effect was that many more safety critical events were occurring in the simulation than could be reasonably expected.

Before the agents are ready to be implemented in a production environment in VISSIM, several issues need to be resolved:

- The areas of application (speeds, geometric conditions) where specific agents should be used needs to be defined. Before being used in a production environment, the selection of appropriate agents for a specific condition should be automated.

- Agent reactions following a safety critical event need to be defined. If a collision occurs, the leader and follower should stop for a certain dwell time. If a vehicle must decelerate sharply, then it should revert to typical car-following behavior rather than replicating the same safety critical actions that led to the conflict in the first place.

- As a future research item, there is a need to estimate the frequency with which certain agent behaviors occur in the driving population to ensure that they are modeled at appropriate levels.

Once these changes are made, the inclusion of agents in VISSIM would offer a powerful opportunity to examine safety implications of different designs, as well as to better assess the impacts of incidents and other unpredictable driver behavior on travel time reliability. The initial testing of the agents in VISSIM shows that the use of agents is viable conceptually, but additional refinement is needed before they can be used outside of a research environment focused on specific events.

Chapter 9. CONCLUSIONS AND RECOMMENDATIONS

CONCLUSIONS

This report has documented the research performed to to model driver behavior in traffic under naturalistic driving data. VTTI naturalistic driving data was used in this research due to its unique capabilities to capture the behavior of drivers when they are not in a test environment. Both safety-critical and car-following periods were used for training agents to mimic the range of behaviors of drivers.

The naturalistic data analysis allowed for the discovery of new thresholds for the Wiedemann model: a passing threshold and a hook following threshold. These two new thresholds increase the usability and accuracy of the Wiedemann model when natural or naturalistic driver behaviors are concerned because these thresholds were developed based on observations made in the naturalistic data. A hybrid model was developed, combining the strengths of both the GHR model and the Wiedemann model to yield a more accurate and highly flexible model. The comparison of different car-following models resulted in finding that different models have the least error for different drivers, but the velocity difference model and the Wiedemann model both perform adequately across all the drivers tested.

In order to model safety-critical events in addition to normal events, NFACRL, an agent-based artificial intelligence machine-learning technique, was used to model driving behavior. The naturalistic driving database was used to train and validate driver agents. The advantages of NFACRL lie mainly in its ability to simulate heterogeneous vehicle actions in complicated traffic environments. It is worth mentioning that this research was an attempt to apply reinforcement learning techniques to solving high-dimensional state problems and continuous-action simulation problems simultaneously in transportation research, especially in traffic flow theory. From the perspective of microscopic traffic behavior modeling, the proposed methodology was able to simulate lateral action, which brings new insights to the modeling of driver maneuvering behavior during safety critical events. The proposed methodology also simulated events from different drivers and proved behavior heterogeneities. Robust agent activation techniques were also developed using discriminant analysis.

The developed agents were implemented in VISSIM simulation platform and were evaluated by comparing the behavior of vehicles with and without agent activation. The results showed very close resemblance of the behavior of agents and driver data. Prototype agents spreadsheets and codes were developed and submitted with this research report.

FUTURE RECOMMENDATIONS

The next step of this research is the extension of the NFACRL framework to simulate other traffic behavior, such as lane-changing behavior, merging behavior in the upstream and downstream of ramps, and evacuation behavior. It would be interesting to model individual driver behavior and the decision-making process under these traffic conditions.

Training parameters, such as speed, memory, and scaling factor were fixed for all the events in this research, but it would be interesting to see how NFACRL performs under different

combination of these factor sets. As with most artificial intelligence methods, NFACRL is a heuristic approach, so there is no guarantee that training results would be optimal. Therefore, training parameters play an important role in guiding NFACRL to find a near optimal result. We also found that agent performance is very sensitive to the driver-dependent training parameters, discrete action sets, and state bounds. Theoretically, it would be better if these parameters were preoptimized before training. However, because the only way to test agent performance is through NFACRL training, parameter optimization and training forms a cycle, and computation time would exponentially increase. Our approach set the parameters based on statistical quartiles, and this yielded good results. Future research should focus on the optimization methodology for training parameters and will improve agent performance.

Initial evaluations on events beyond what the agents were trained for revealed that agents can behave erratically in those cases. It it recommended to train agents for future naturalistic data sets (e.g., SHRP II naturalistic datasets). In addition, it would also be interesting to examine the behavior of agents if they were first trained to mimic the Wiedemann model behavior, for example, then overlayed with naturalistic data training.

Finally, the research conducted in this report has also showed the importance of two key future research issues:

1- Adaptability of agents in real time: agents could be programmed to adapt during the simulation itself.
2- Human factor issues related to warning individual drivers about a change in their driving behavior that might lead to a safety-critical event.

APPENDIX A.

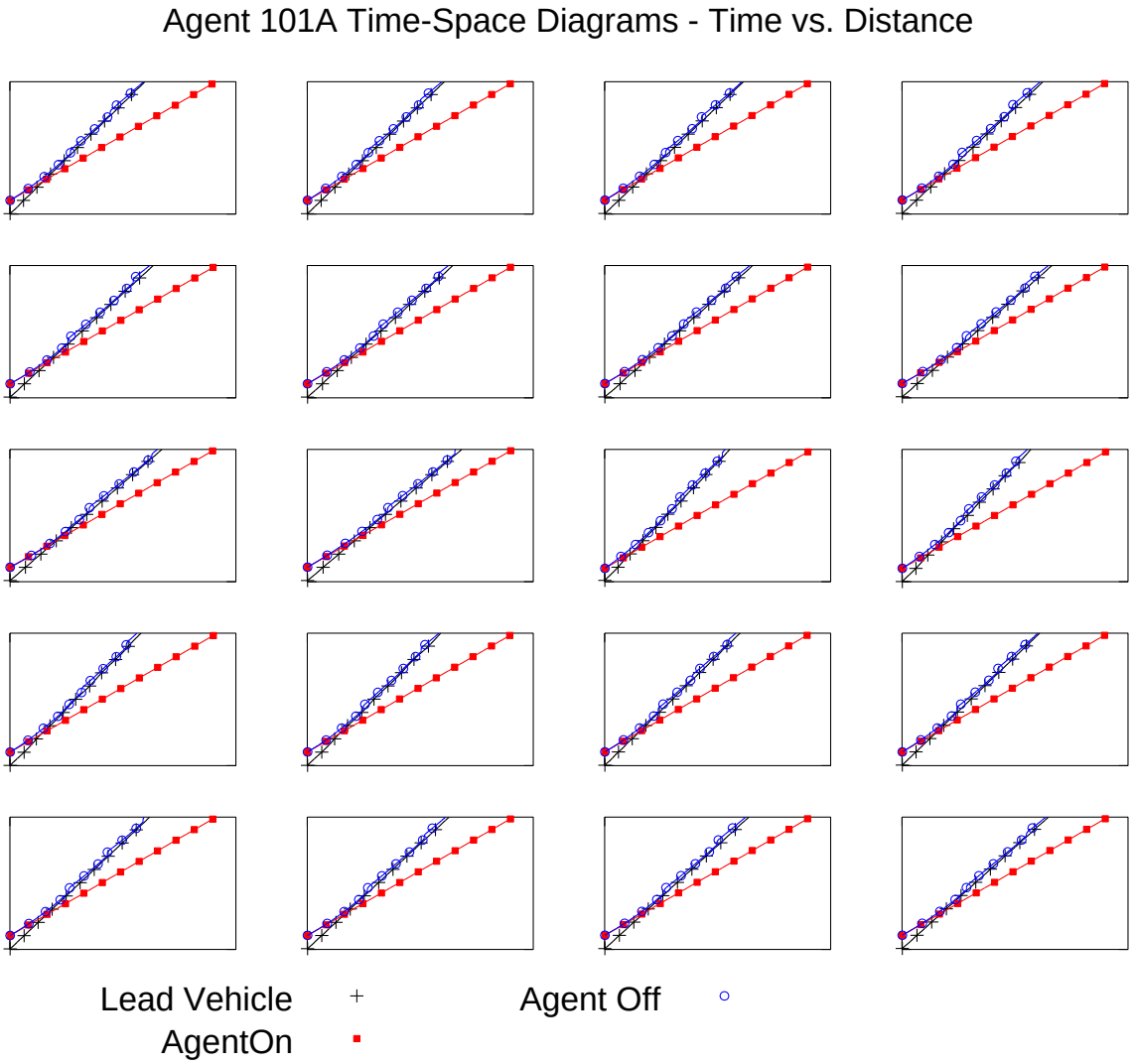

Figure 150. Agent 101A time-space diagrams (time on x-axis, distance on y-axis).

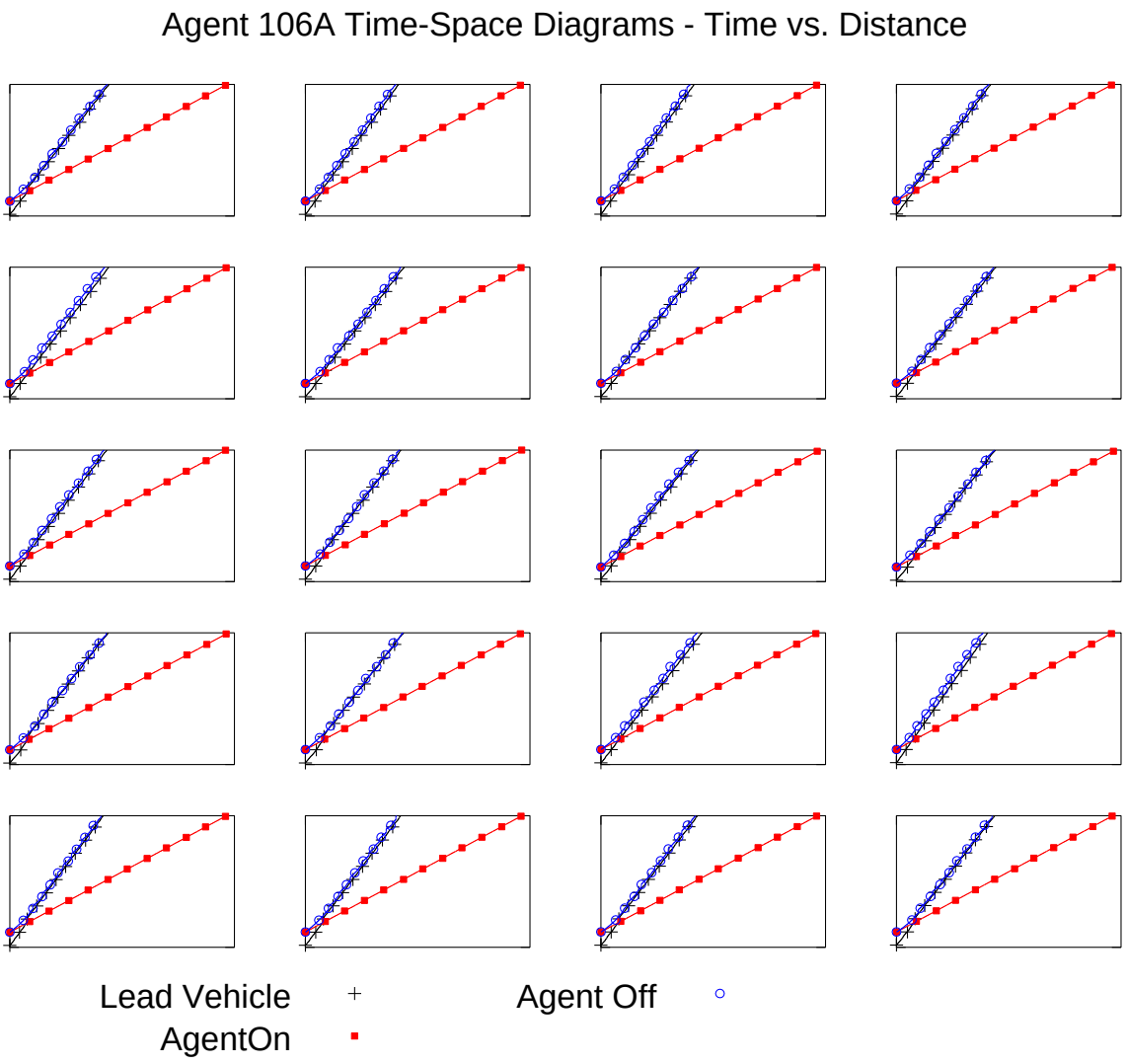

Figure 151. Agent 106A time-space diagrams (time on x-axis, distance on y-axis).

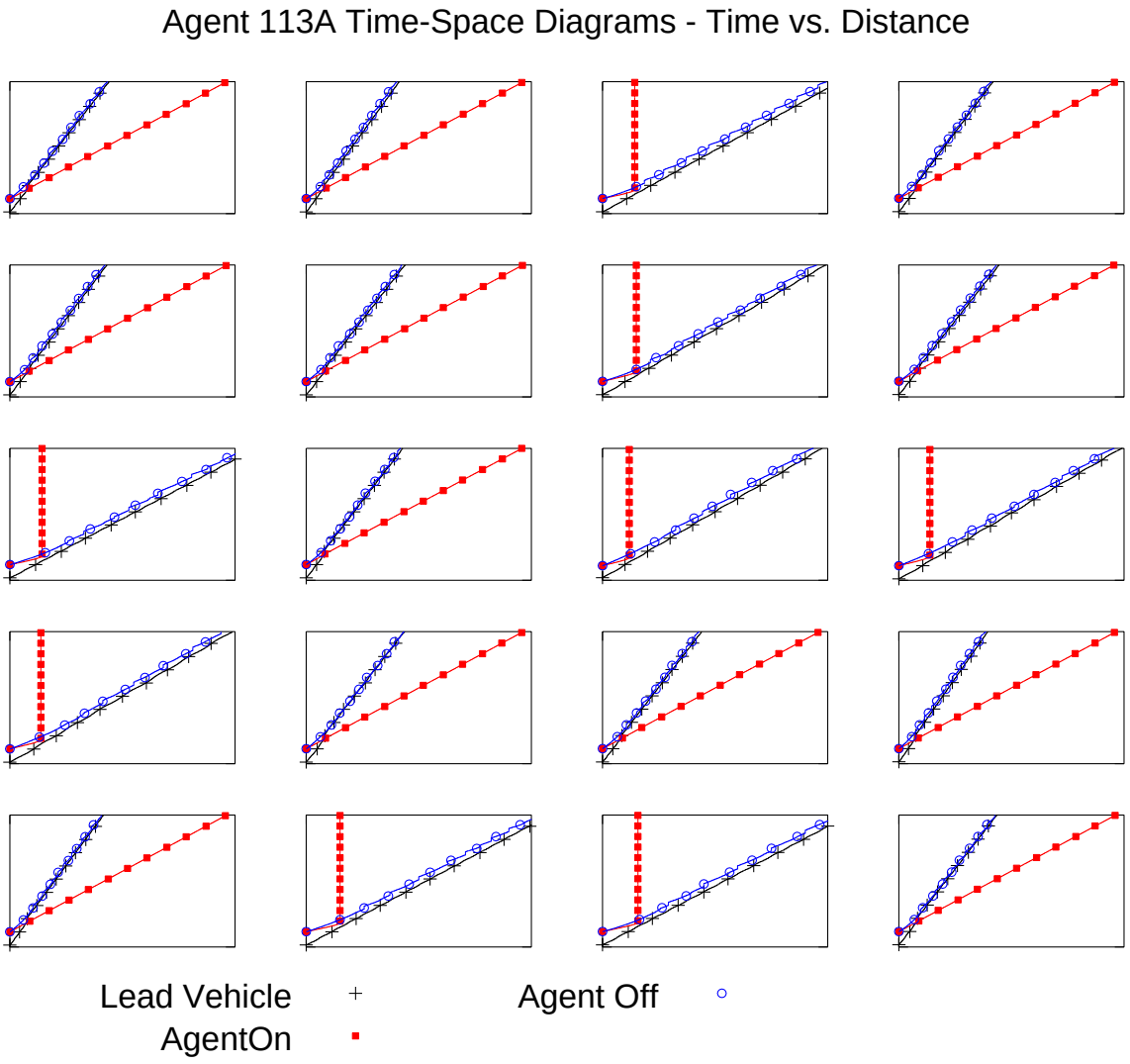

Figure 152. Agent 113A time-space diagrams (time on x-axis, distance on y-axis).

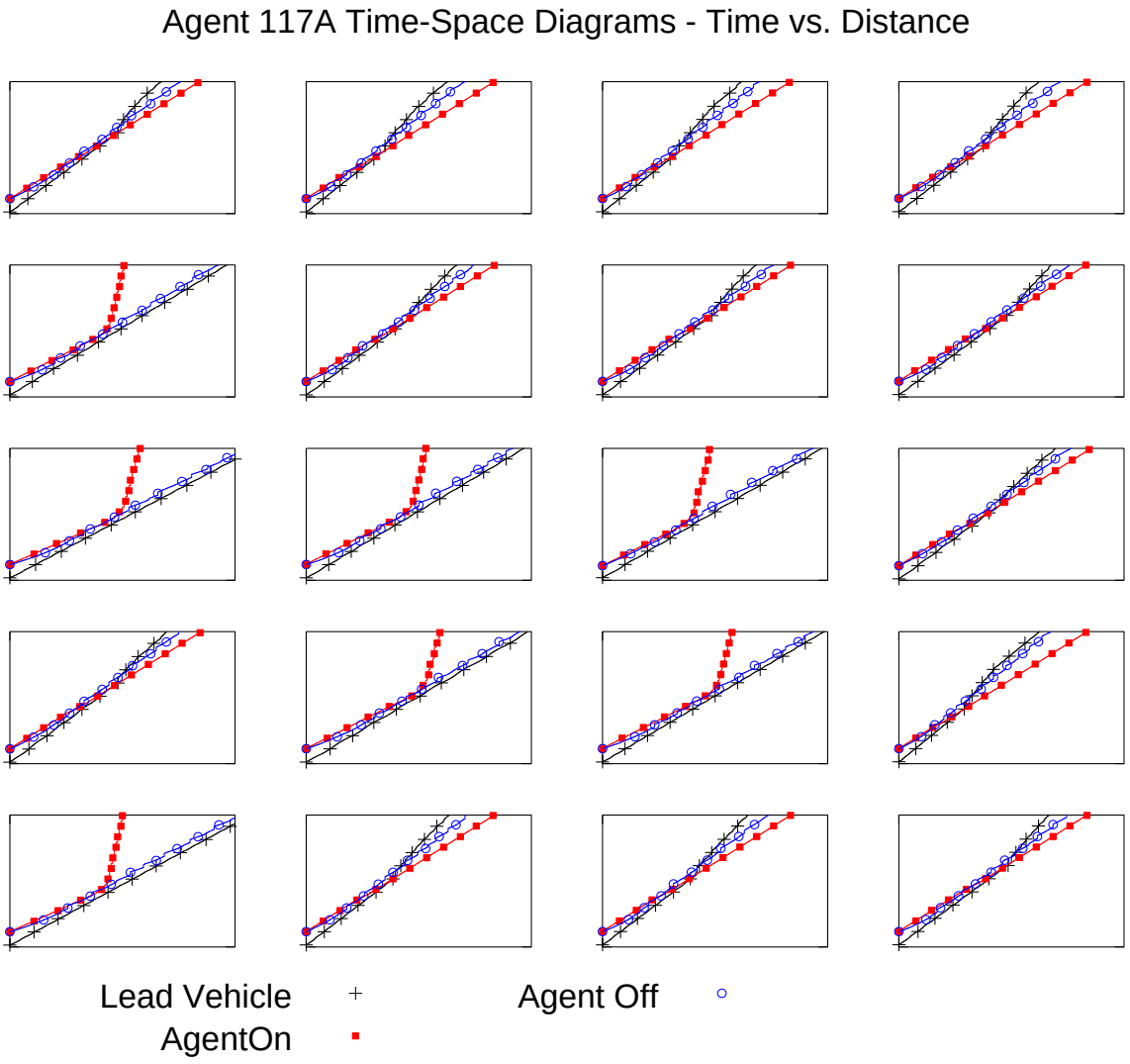

Figure 153. Agent 117A time-space diagrams (time on x-axis, distance on y-axis).

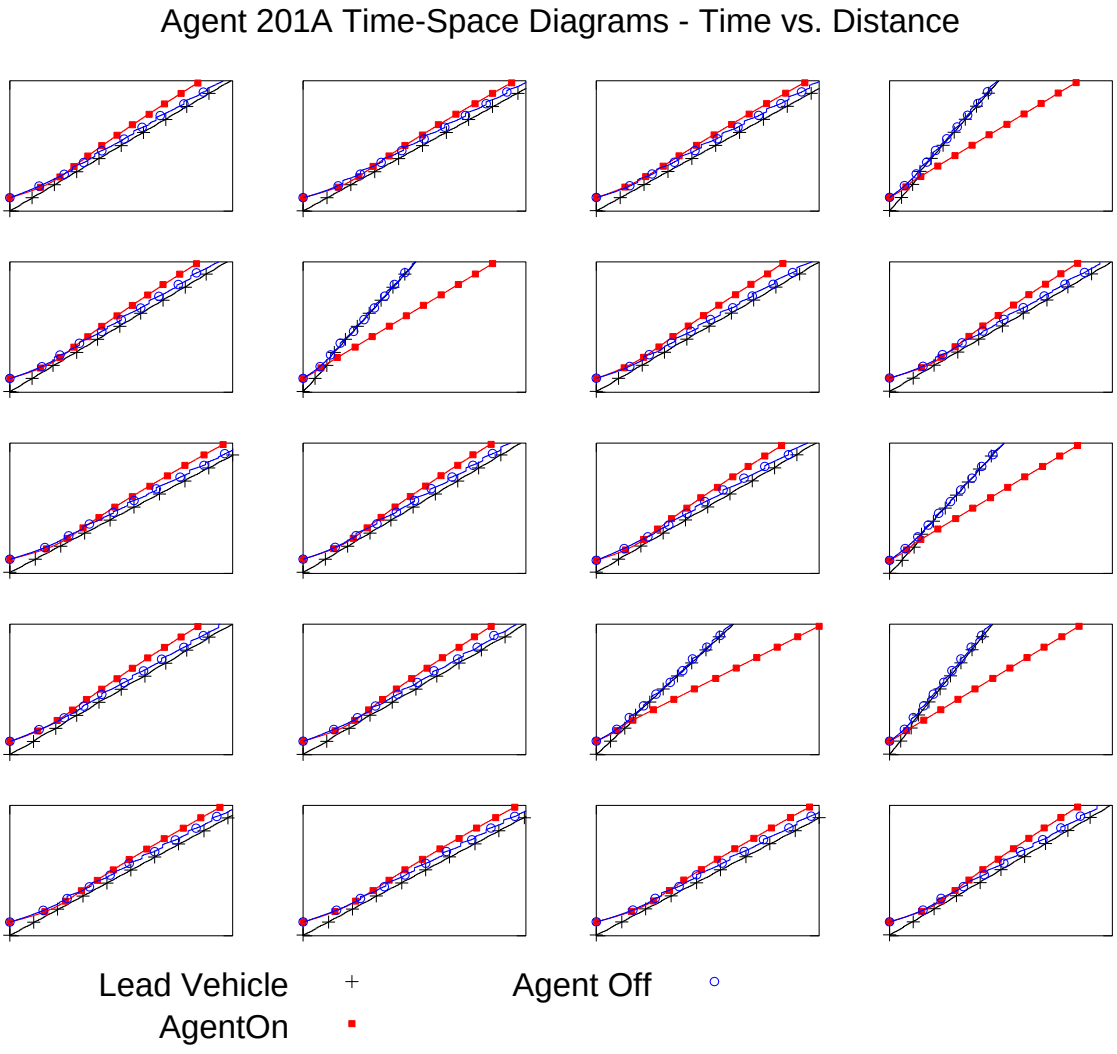

Figure 154. Agent 201A time-space diagrams (time on x-axis, distance on y-axis).

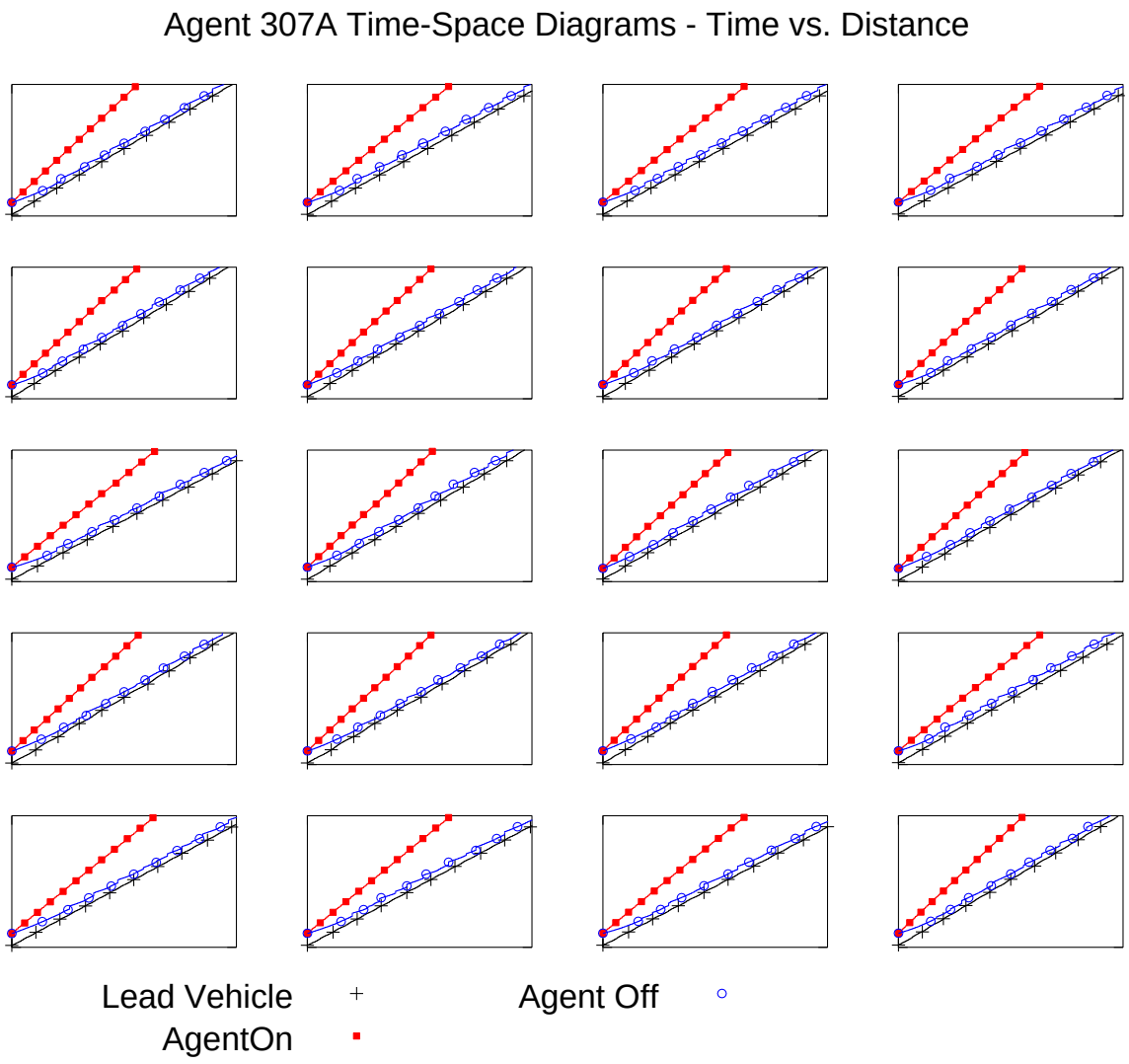

Figure 155. Agent 307A time-space diagrams (time on x-axis, distance on y-axis).

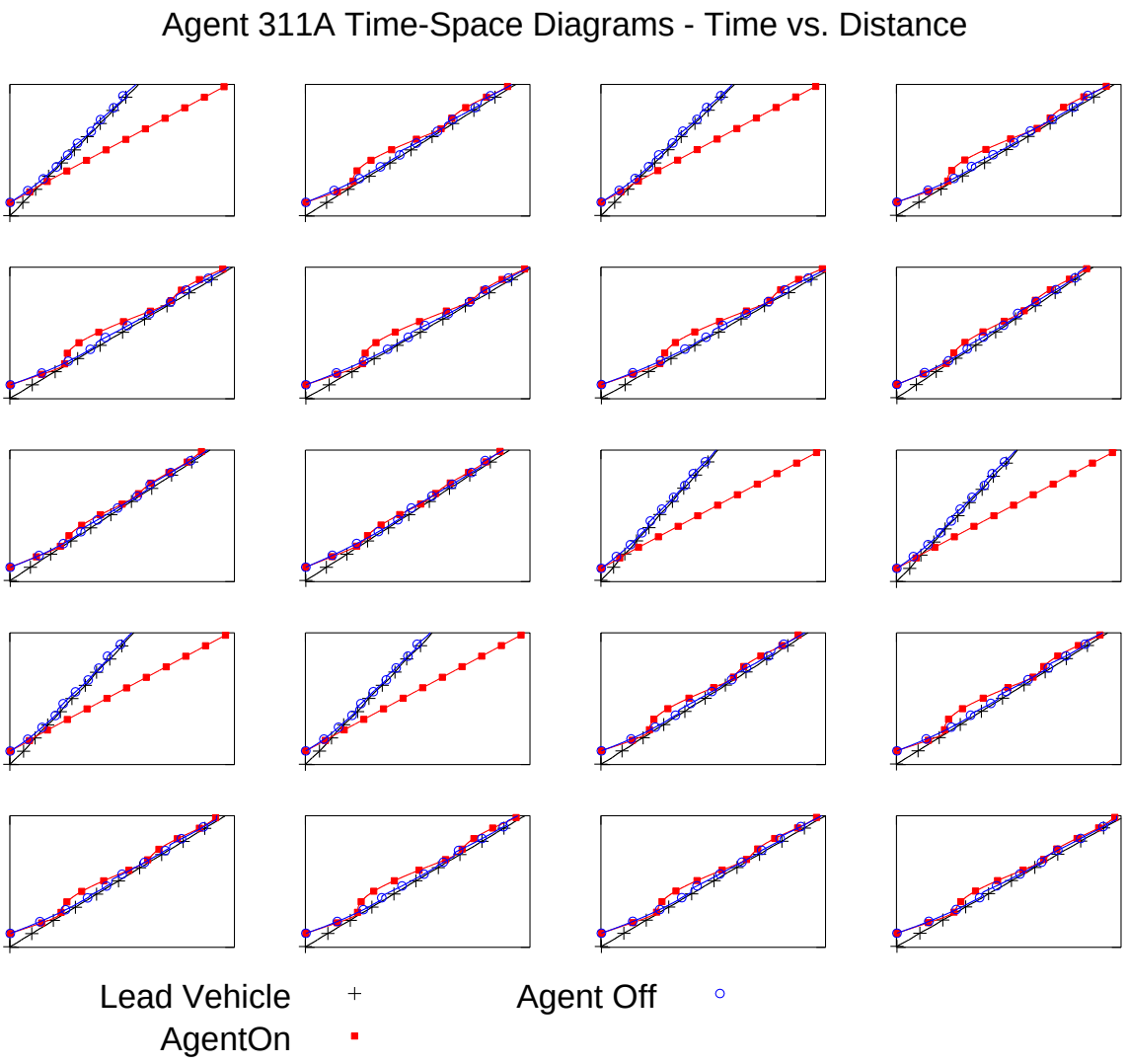

Figure 156. Agent 311A time-space diagrams (time on x-axis, distance on y-axis).

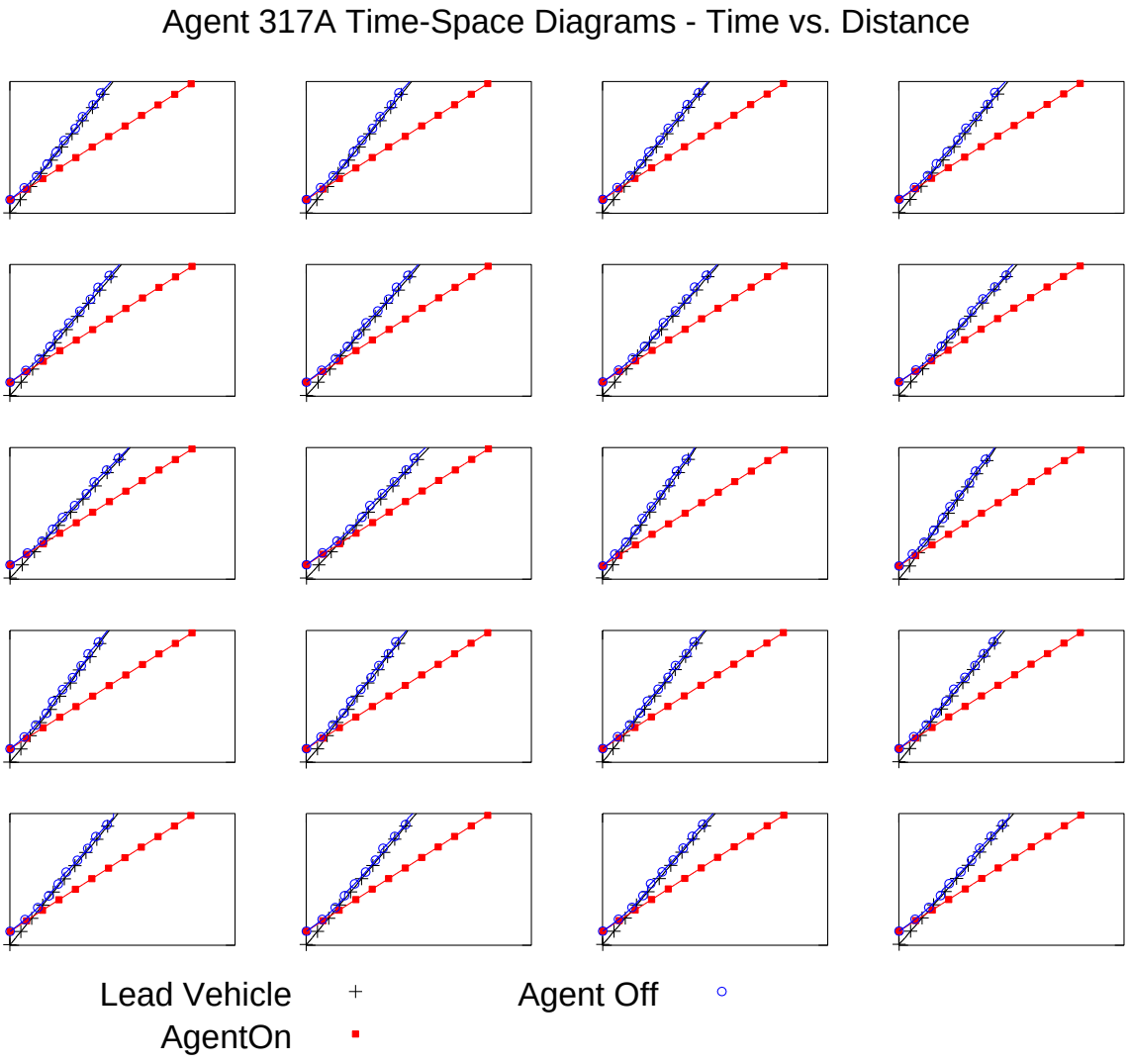

Figure 157. Agent 317A time-space diagrams (time on x-axis, distance on y-axis).

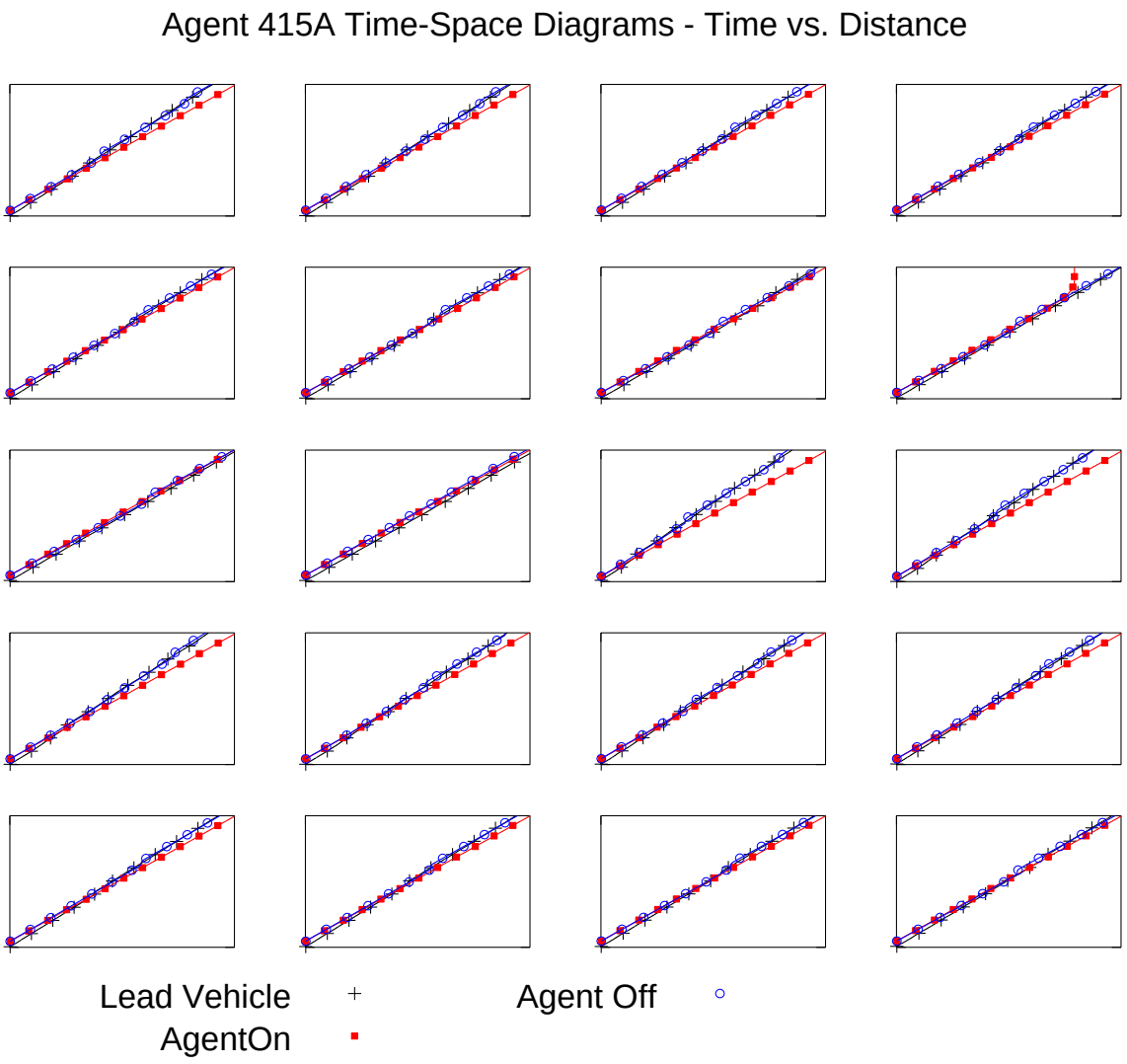

Figure 158. Agent 415A time-space diagrams (time on x-axis, distance on y-axis).

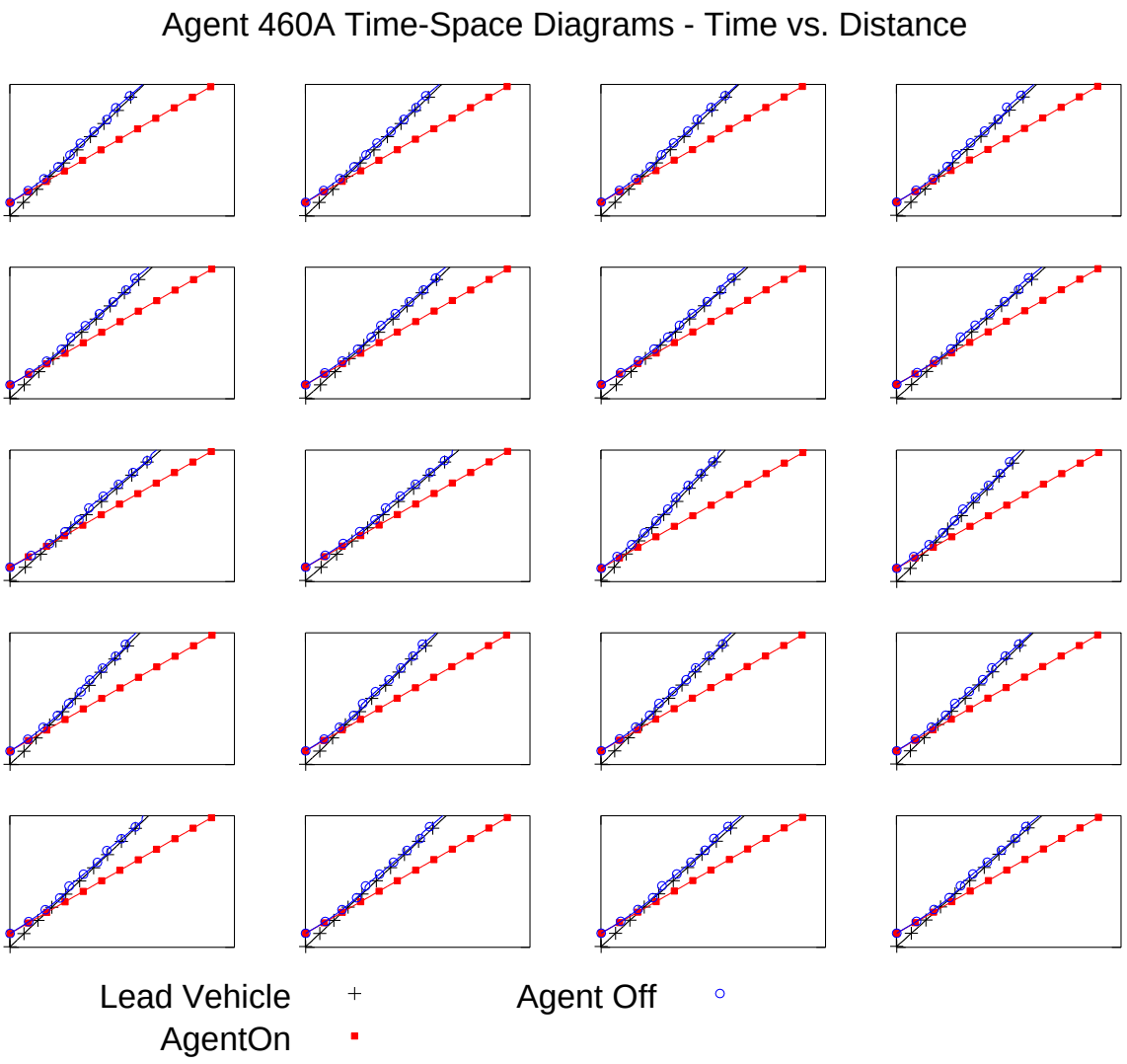

Figure 159. Agent 460A time-space diagrams (time on x-axis, distance on y-axis).

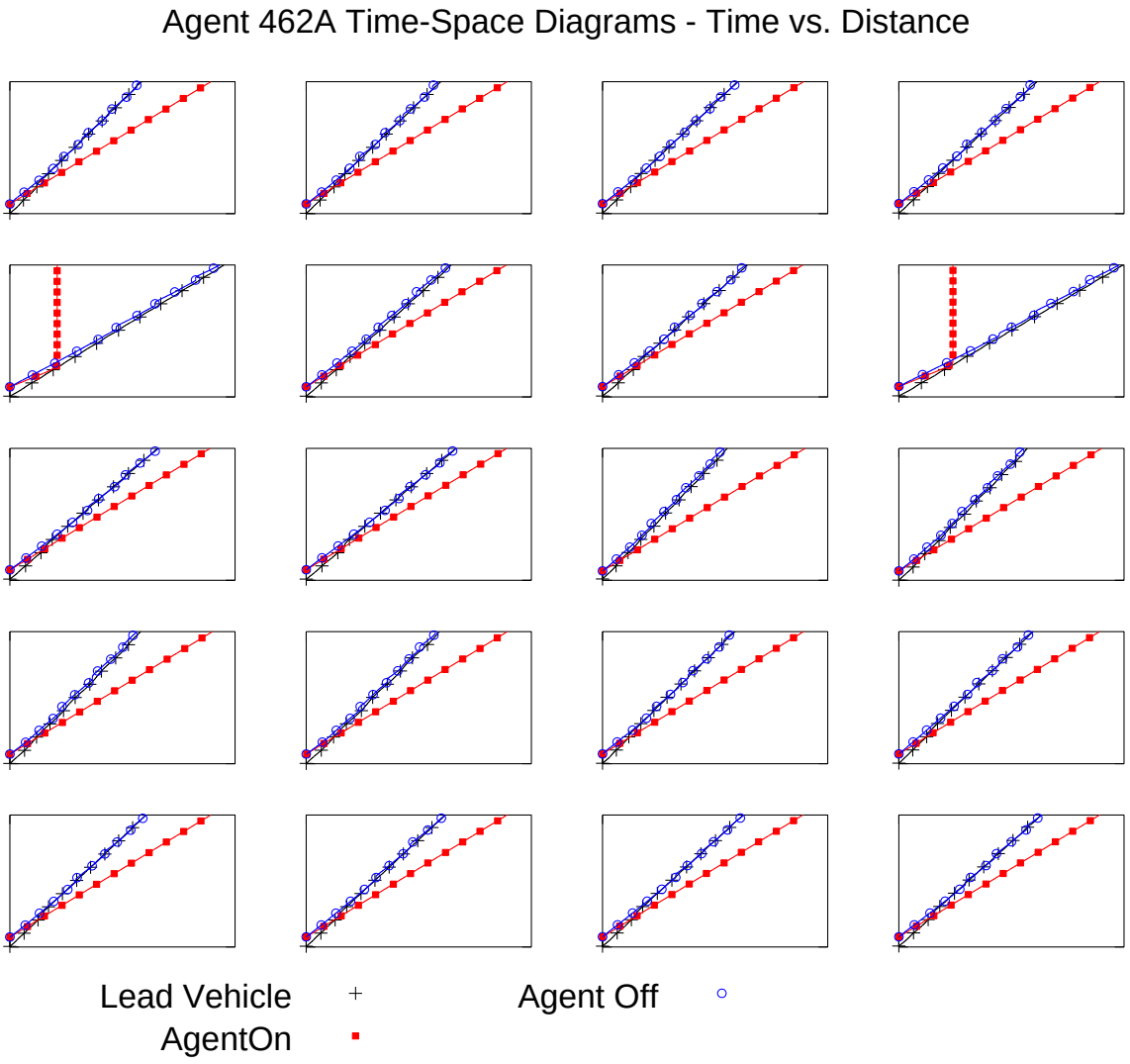

Figure 160. Agent 462A time-space diagrams (time on x-axis, distance on y-axis).

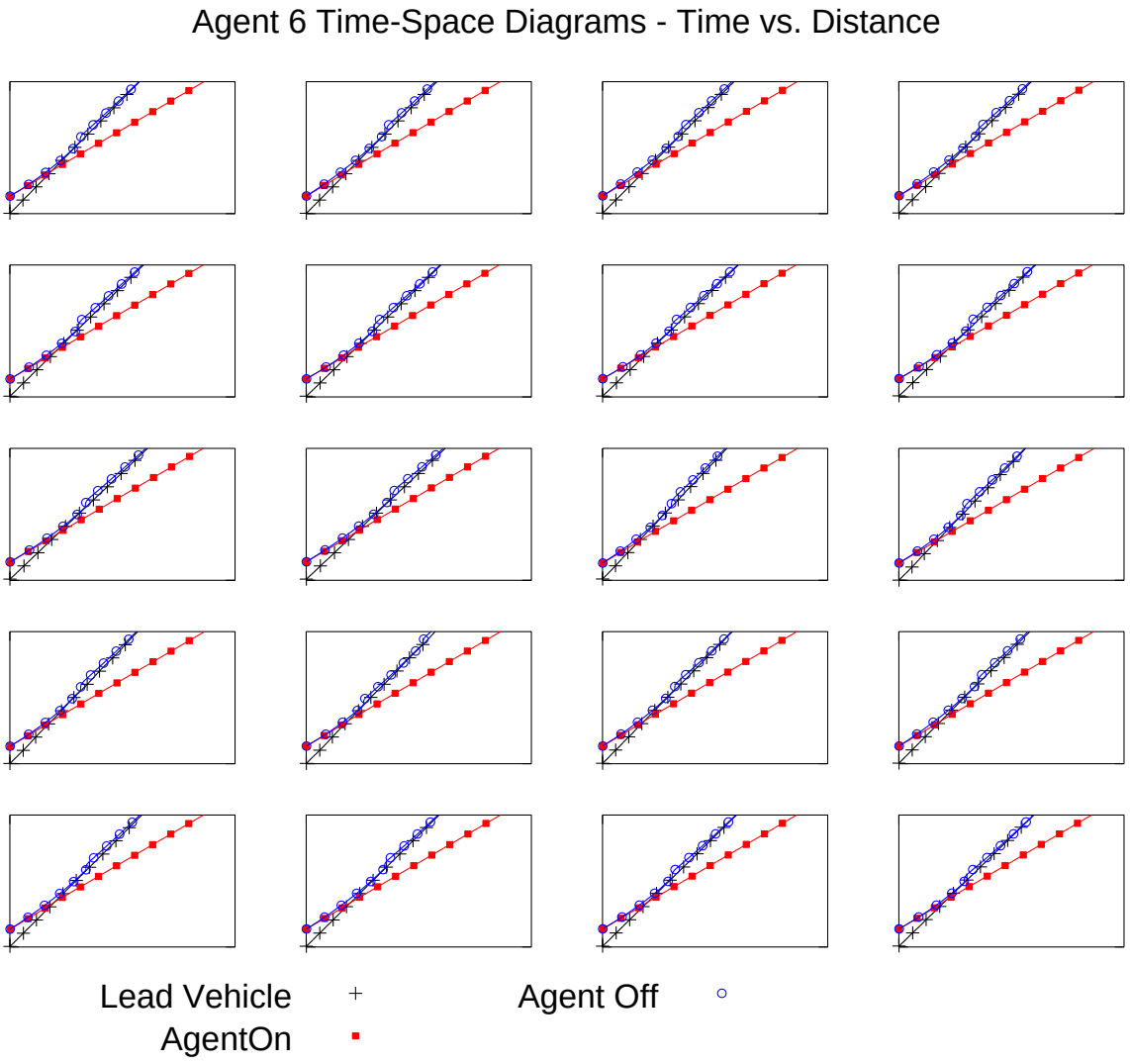

Figure 161. Agent 6 time-space diagrams (time on x-axis, distance on y-axis).

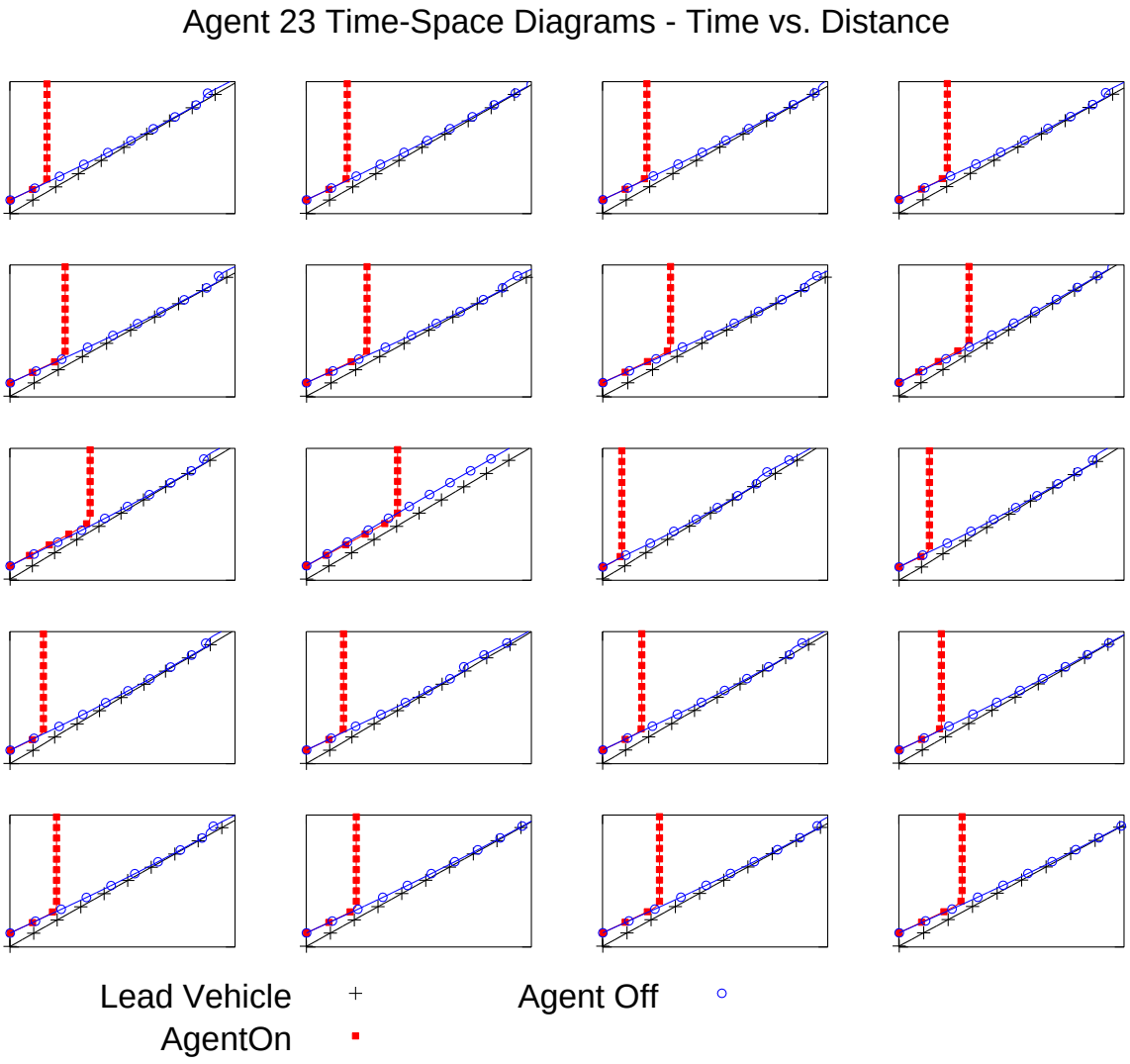

Figure 162. Agent 23 time-space diagrams (time on x-axis, distance on y-axis).

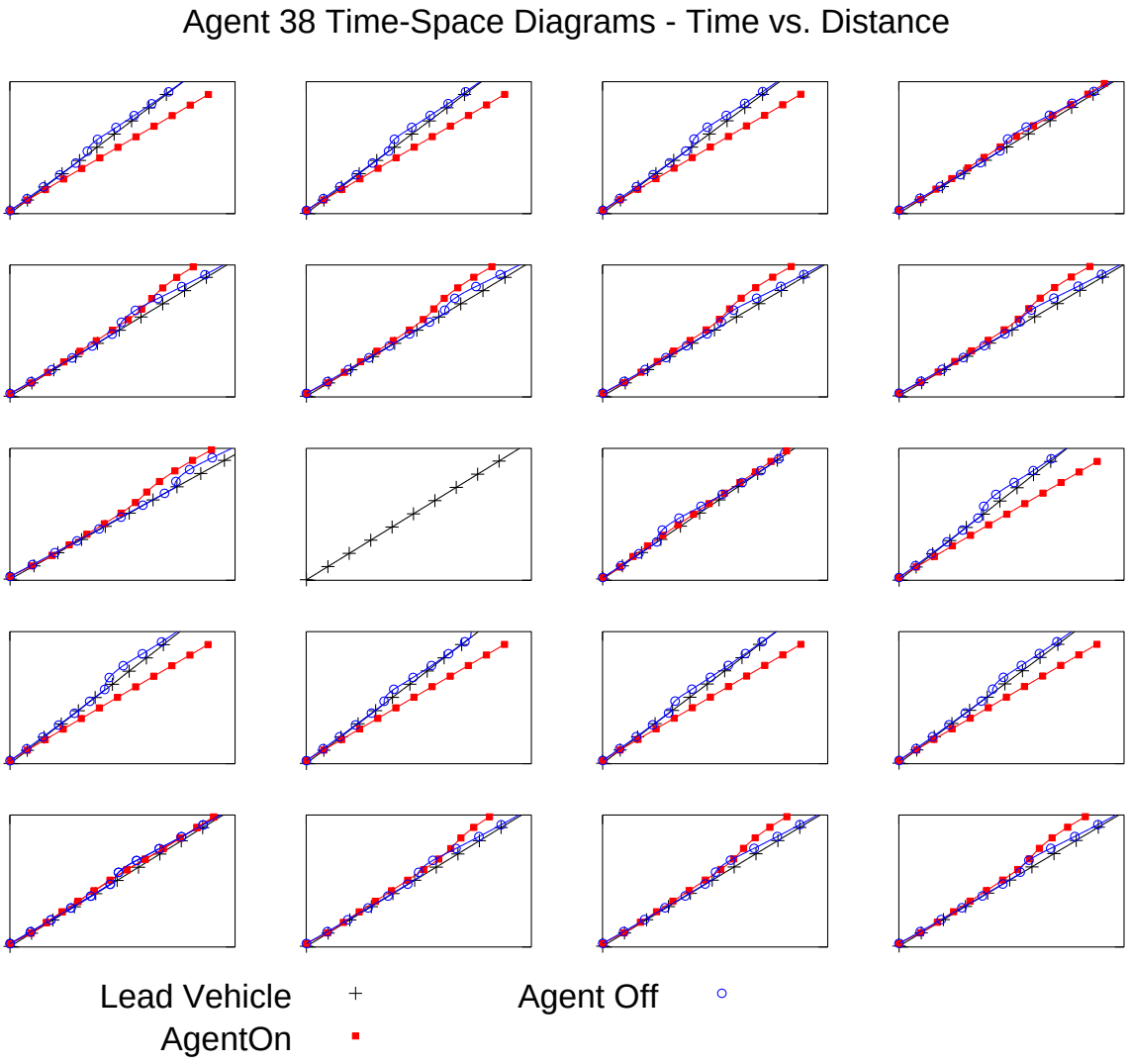

Figure 163. Agent 38 time-space diagrams (time on x-axis, distance on y-axis).

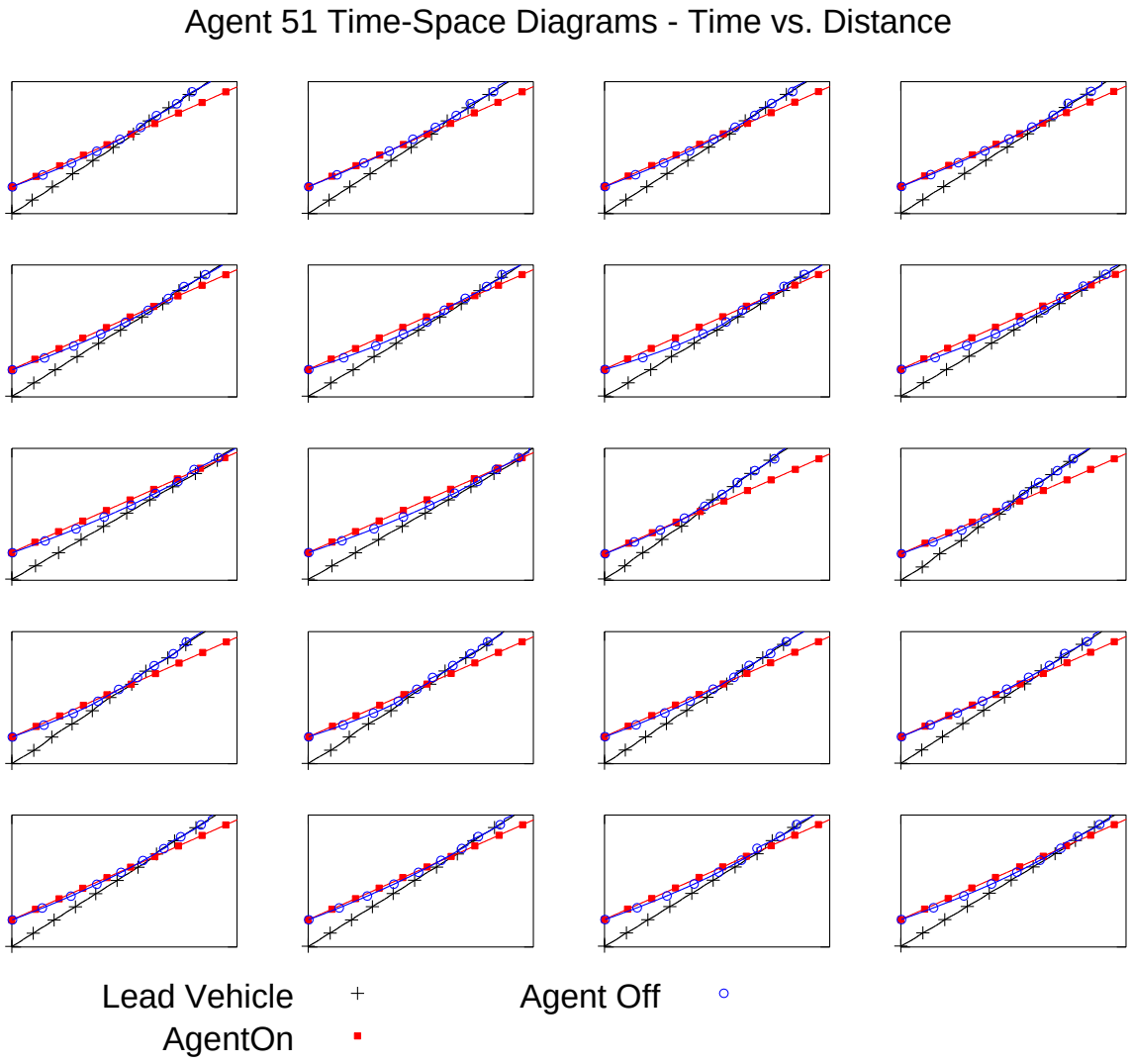

Figure 164. Agent 51 time-space diagrams (time on x-axis, distance on y-axis).

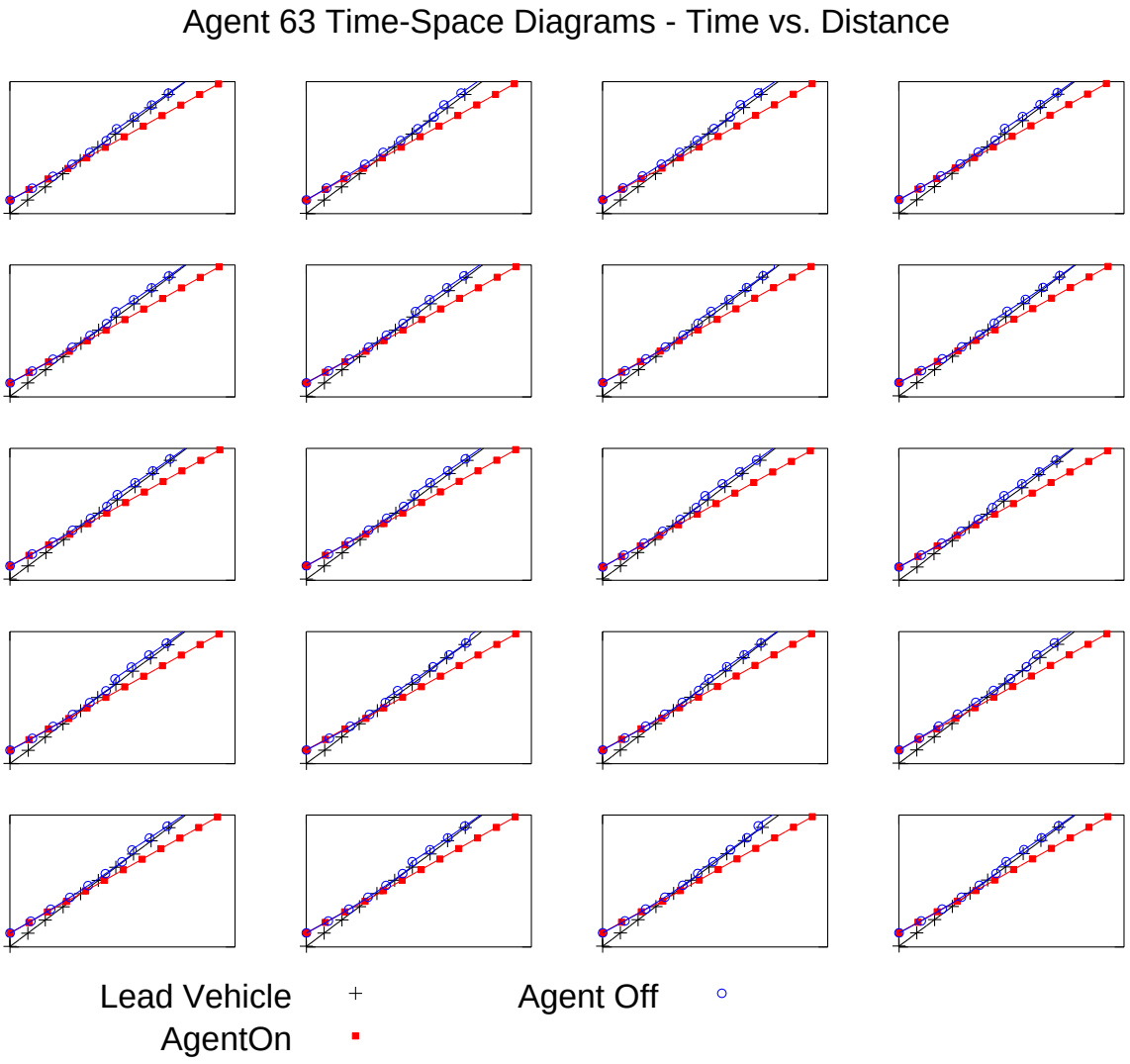

Figure 165. Agent 63 time-space diagrams (time on x-axis, distance on y-axis).

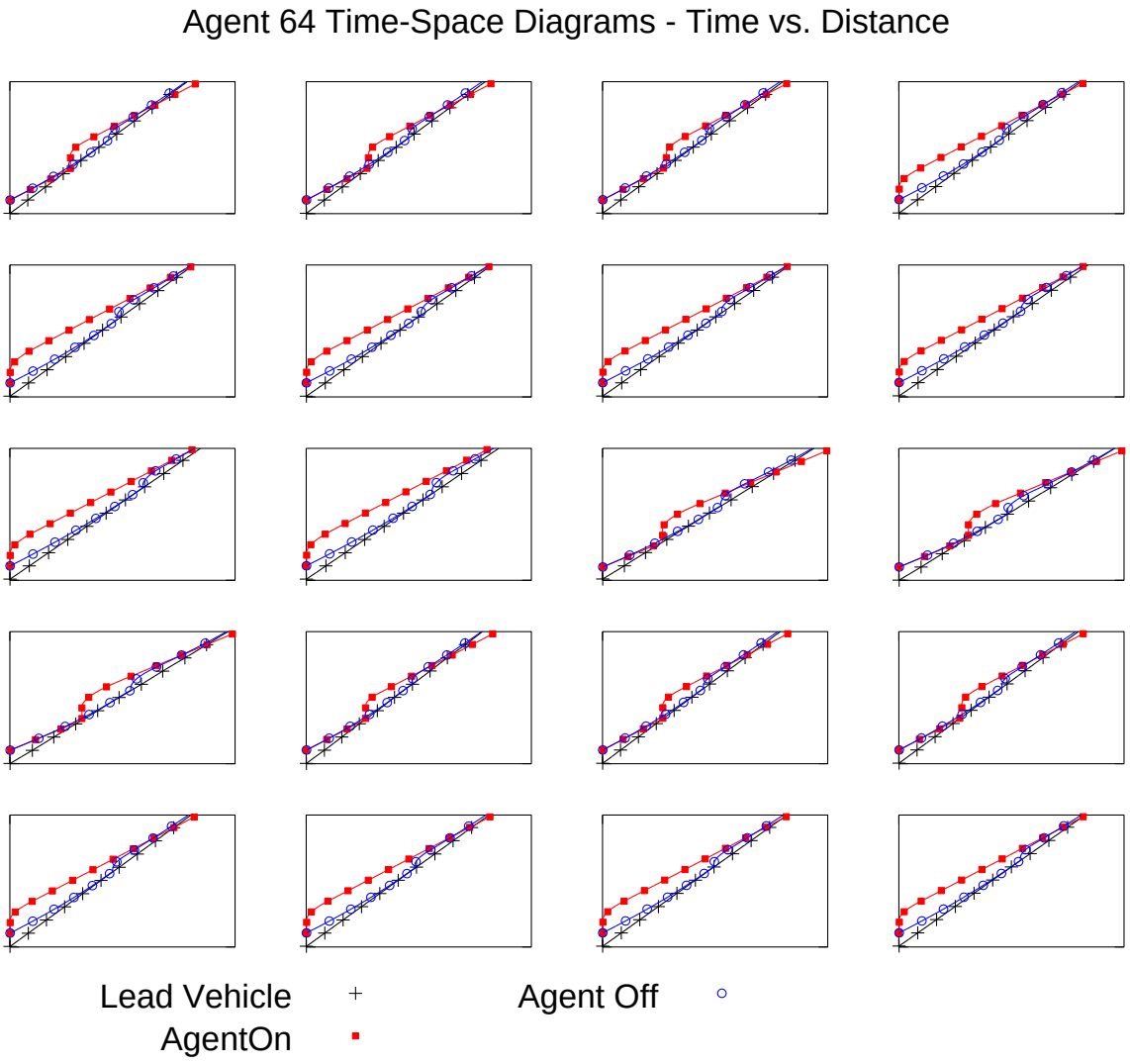

Figure 166. Agent 64 time-space diagrams (time on x-axis, distance on y-axis).

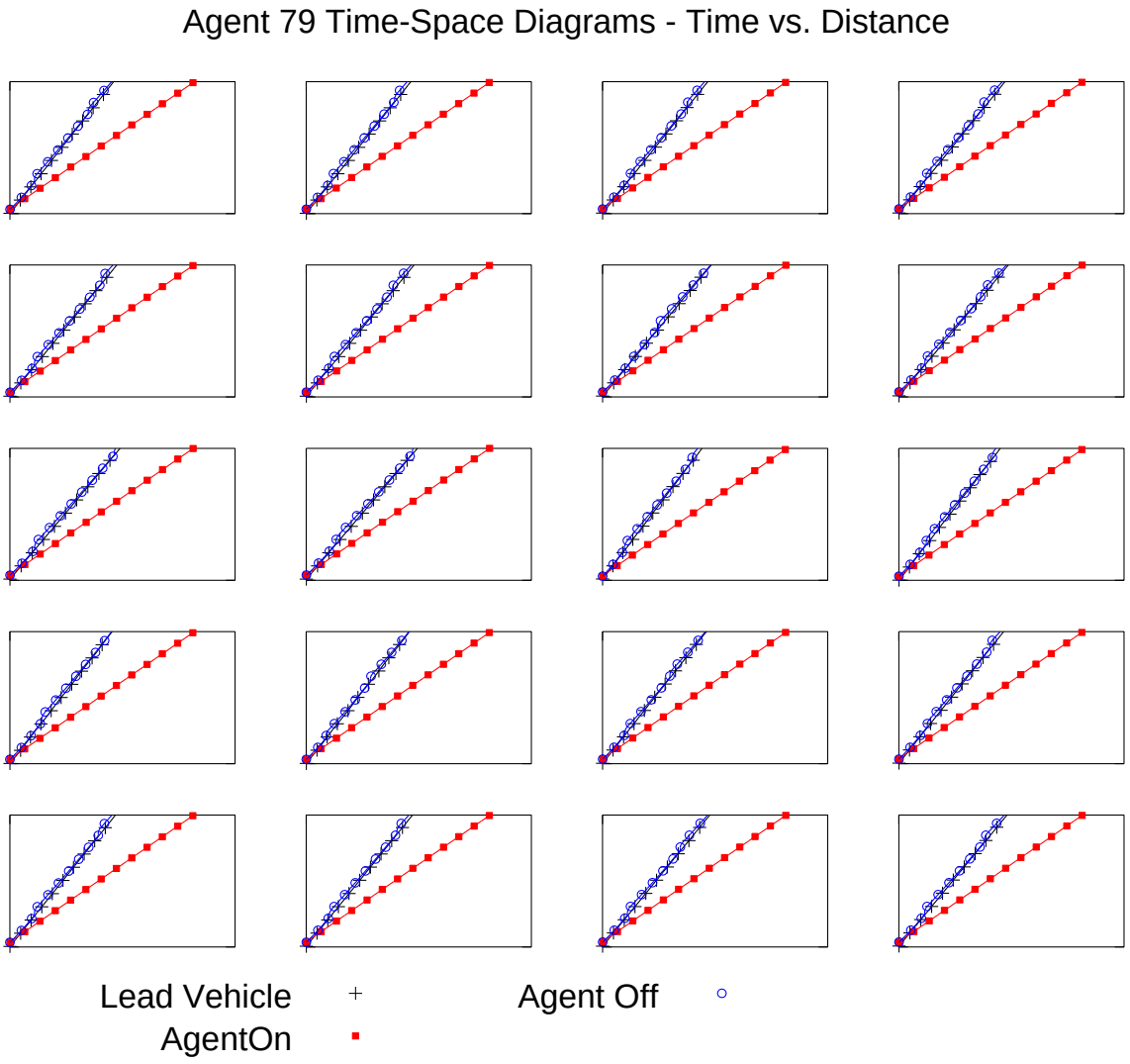

Figure 167. Agent 79 time-space diagrams (time on x-axis, distance on y-axis).

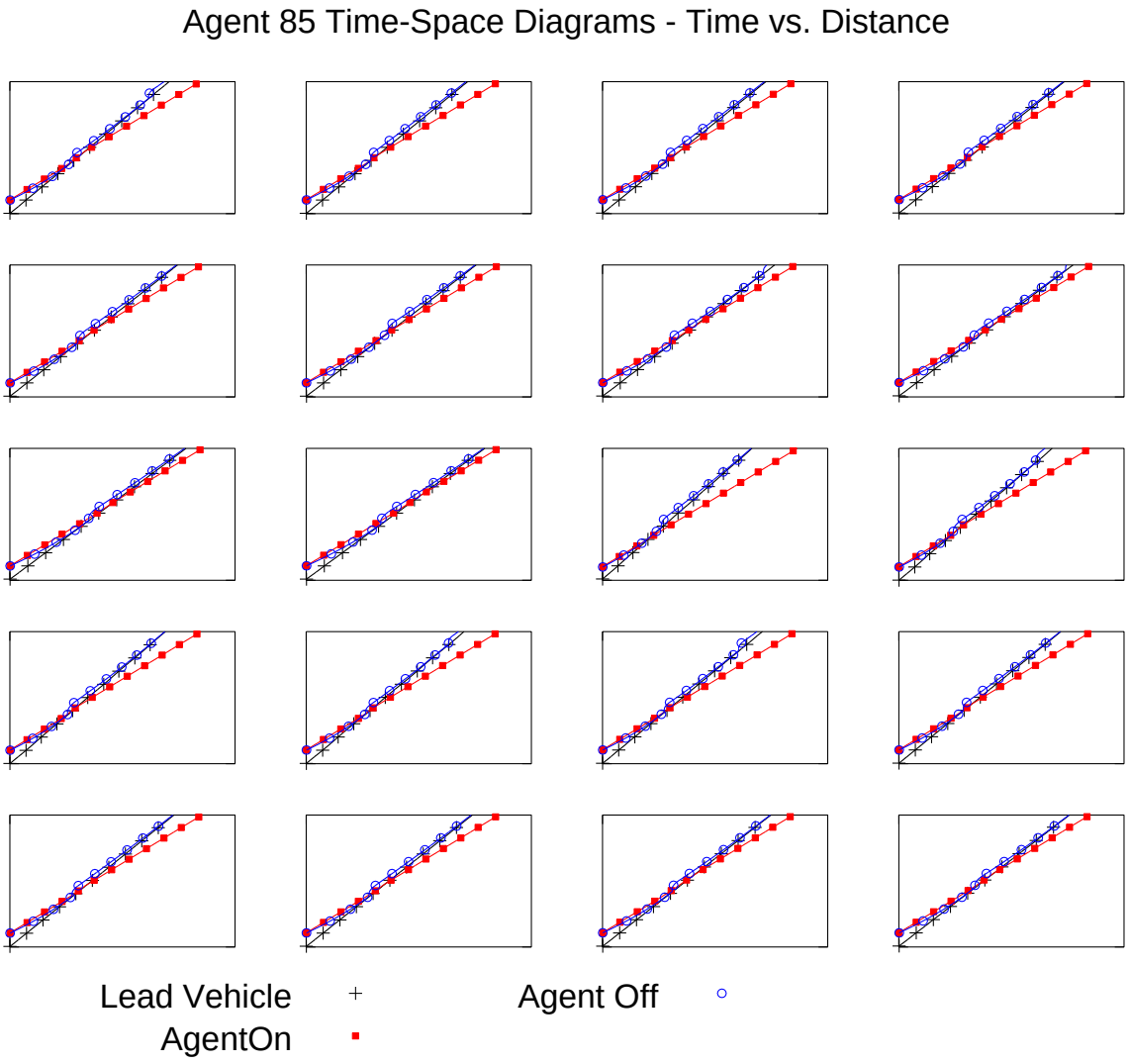

Figure 168. Agent 85 time-space diagrams (time on x-axis, distance on y-axis).

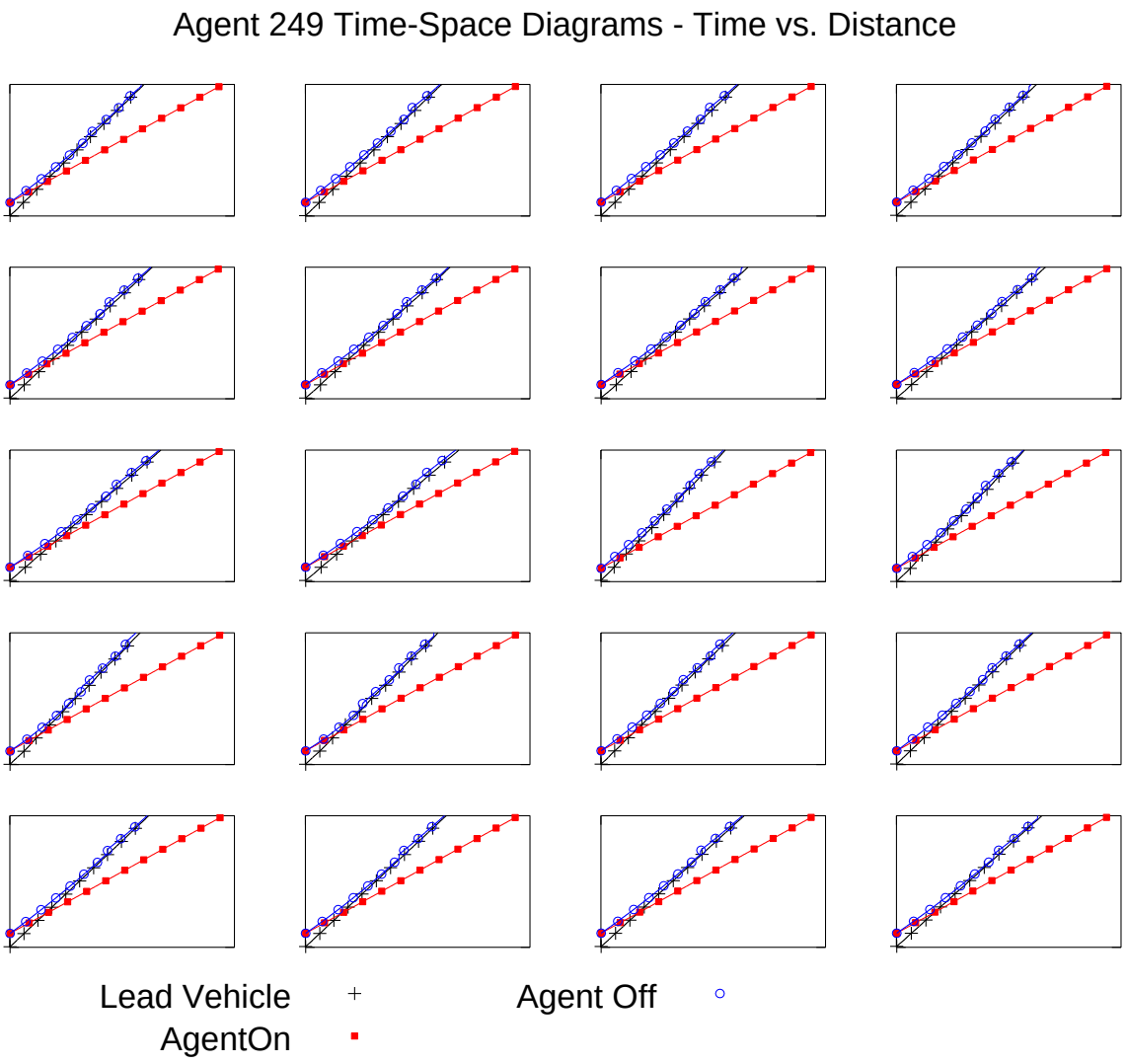

Figure 169. Agent 249 time-space diagrams (time on x-axis, distance on y-axis).

APPENDIX B.

Figure 170. Agent 101A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

Figure 171. Agent 106A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

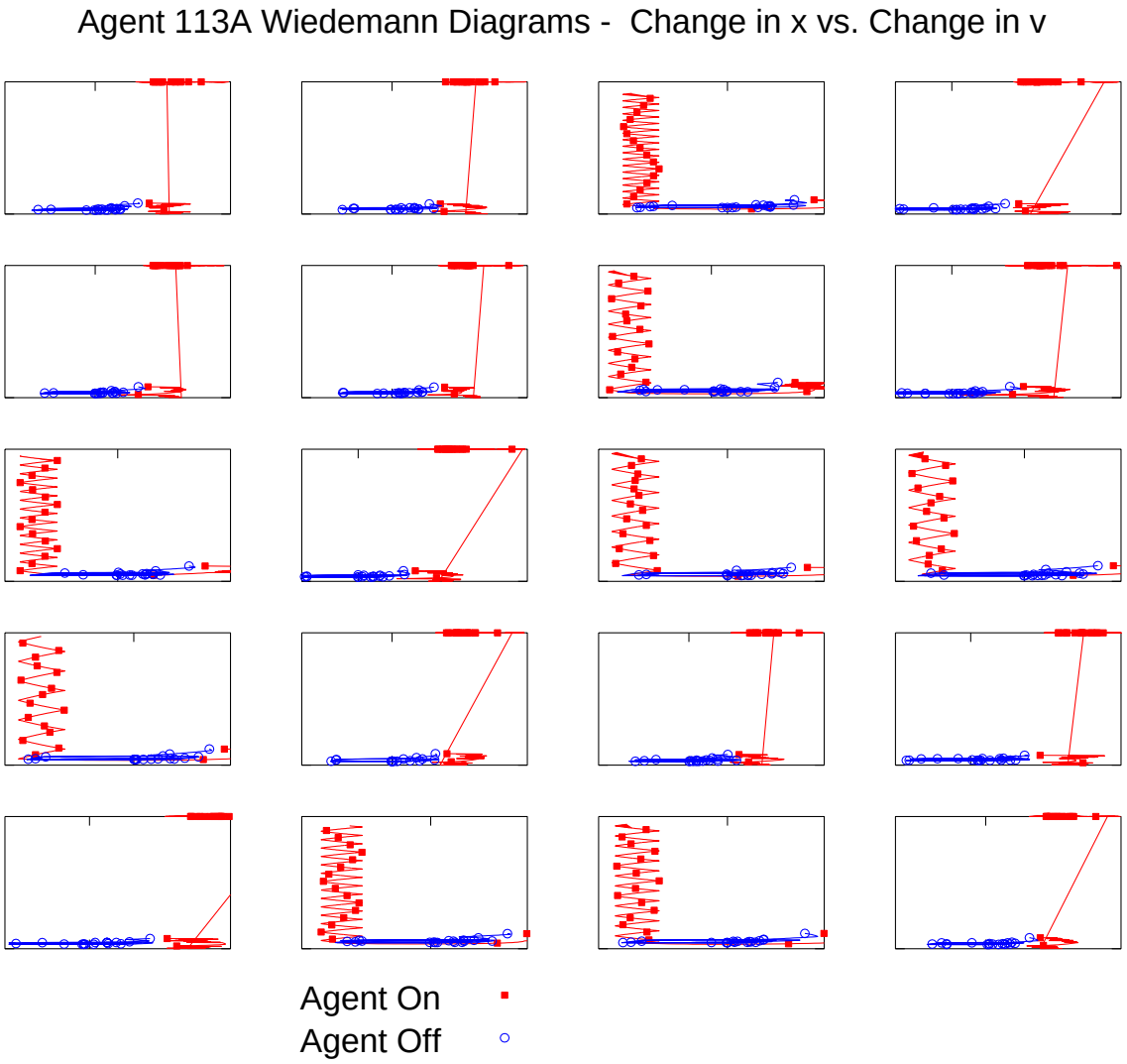

Figure 172. Agent 113A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

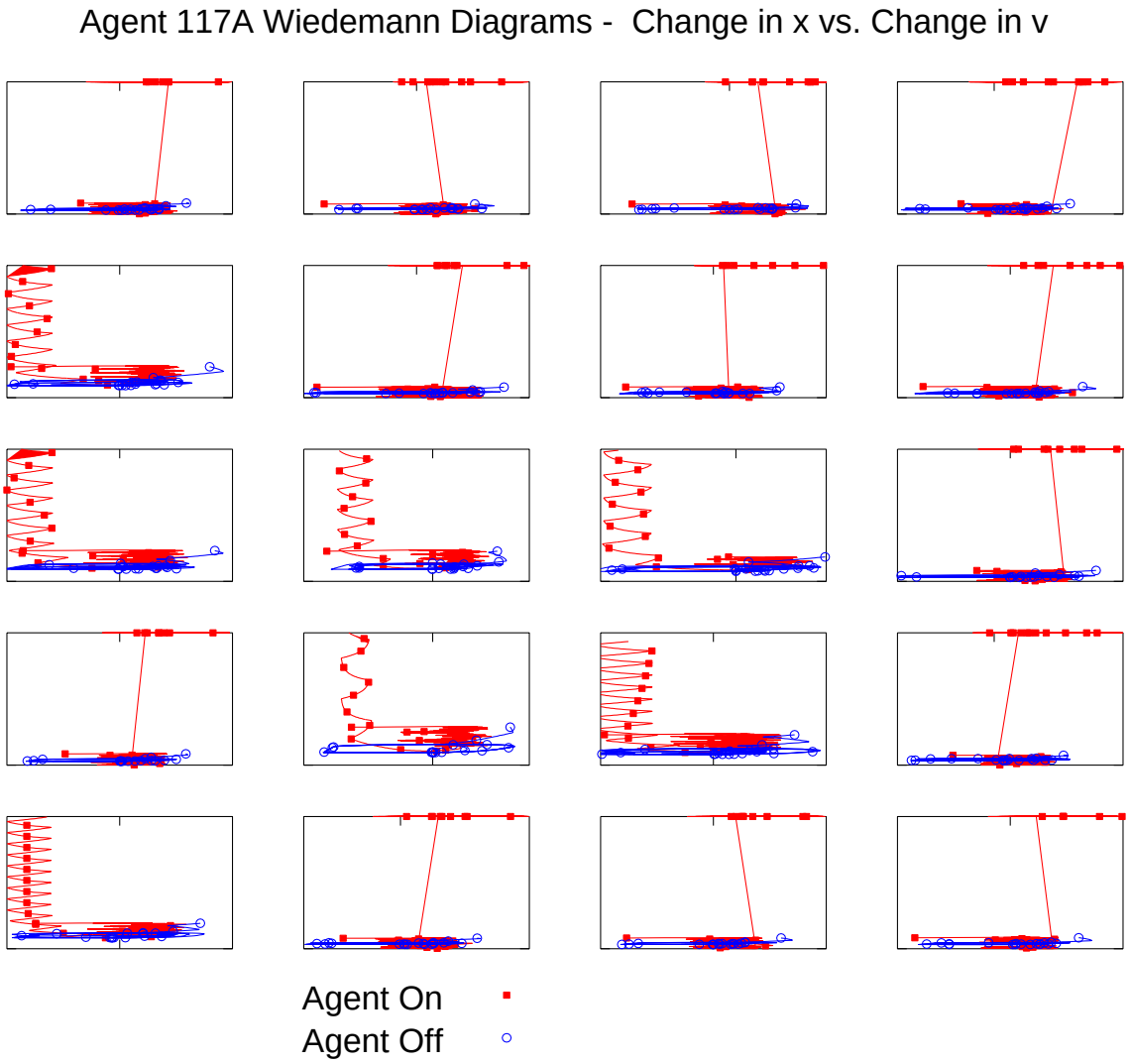

Figure 173. Agent 117A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

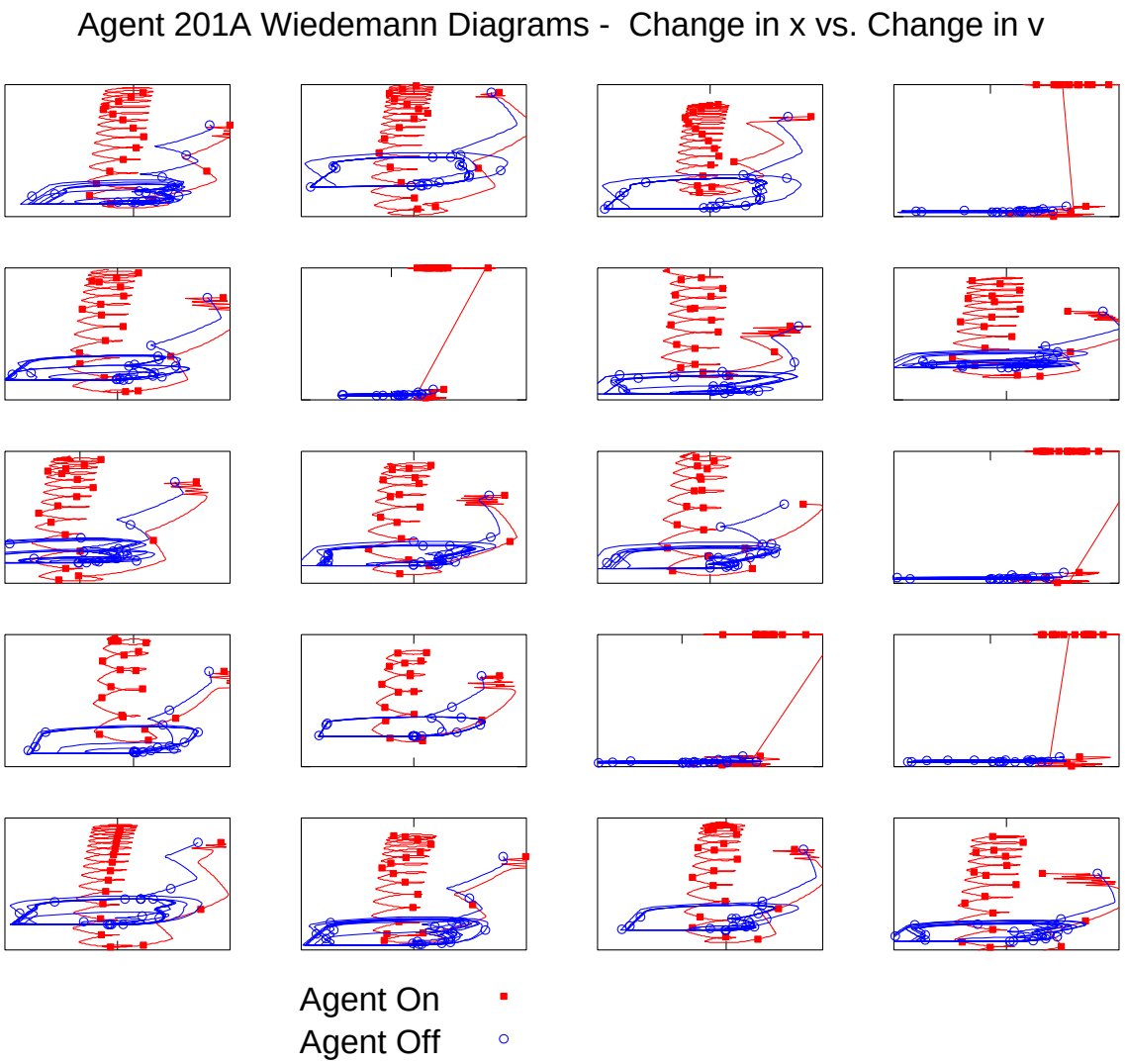

Figure 174. Agent 201A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

Figure 175. Agent 307A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

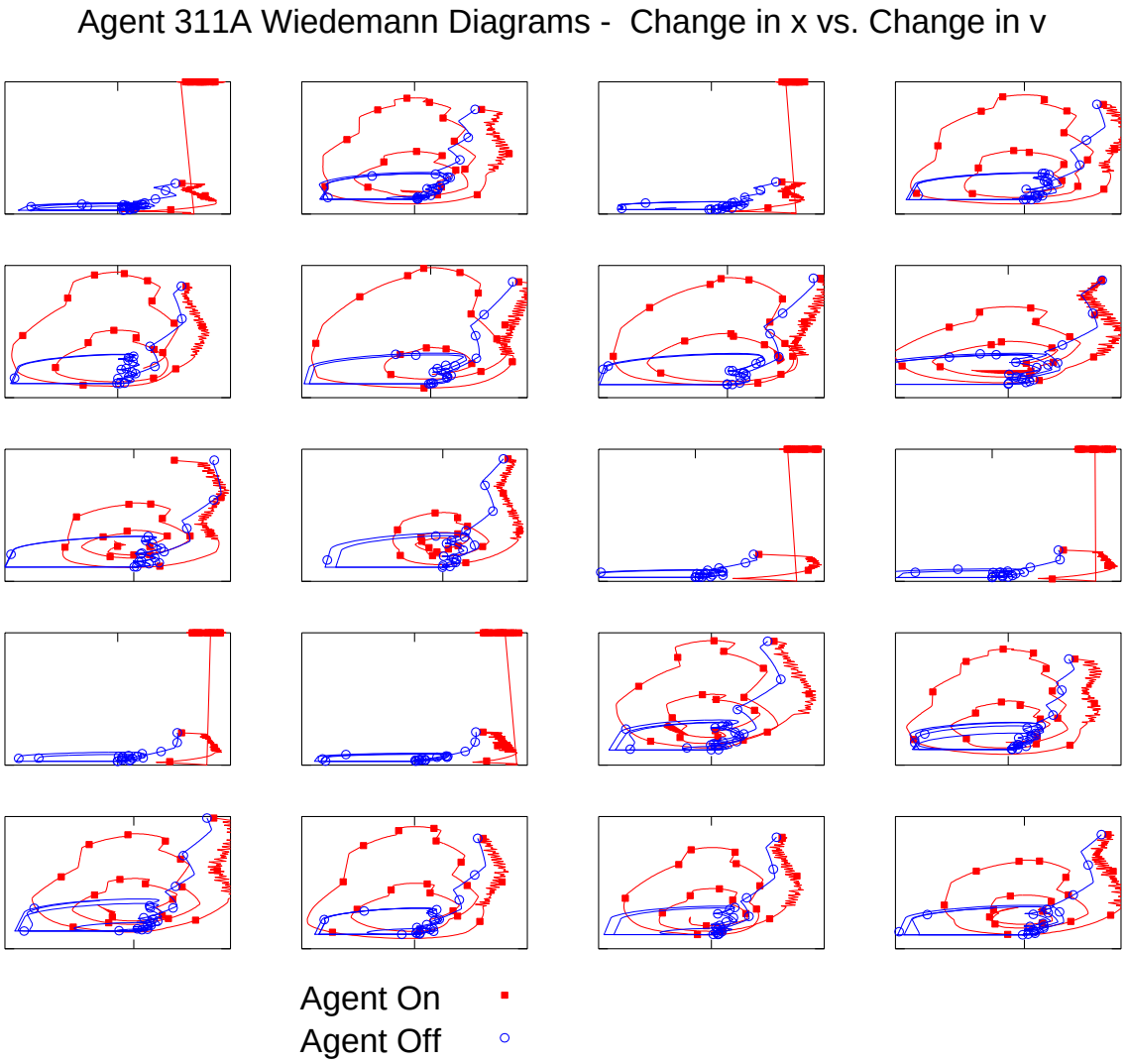

Figure 176. Agent 311A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

Figure 177. Agent 317A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

Figure 178. Agent 415A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

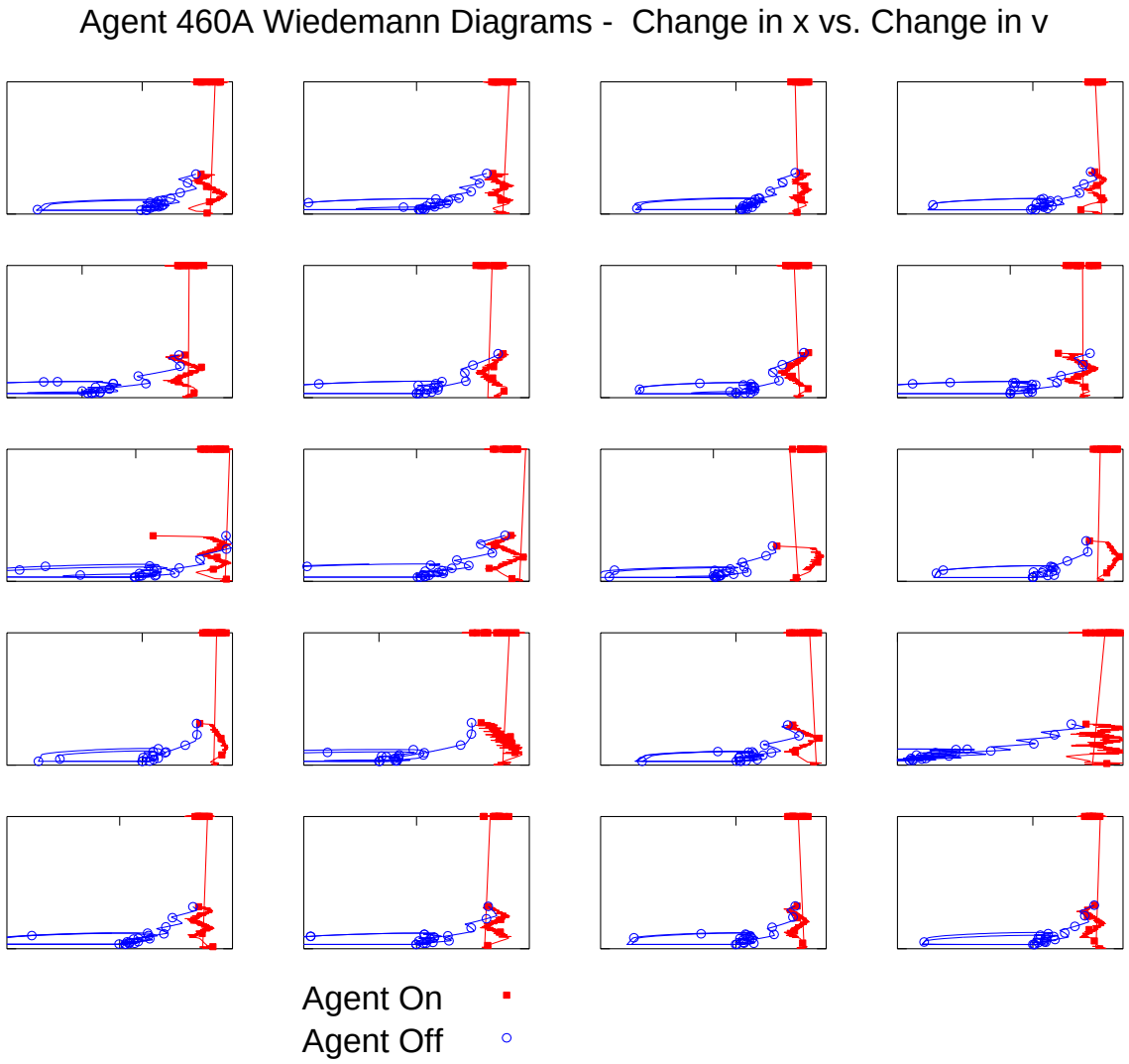

Figure 179. Agent 460A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

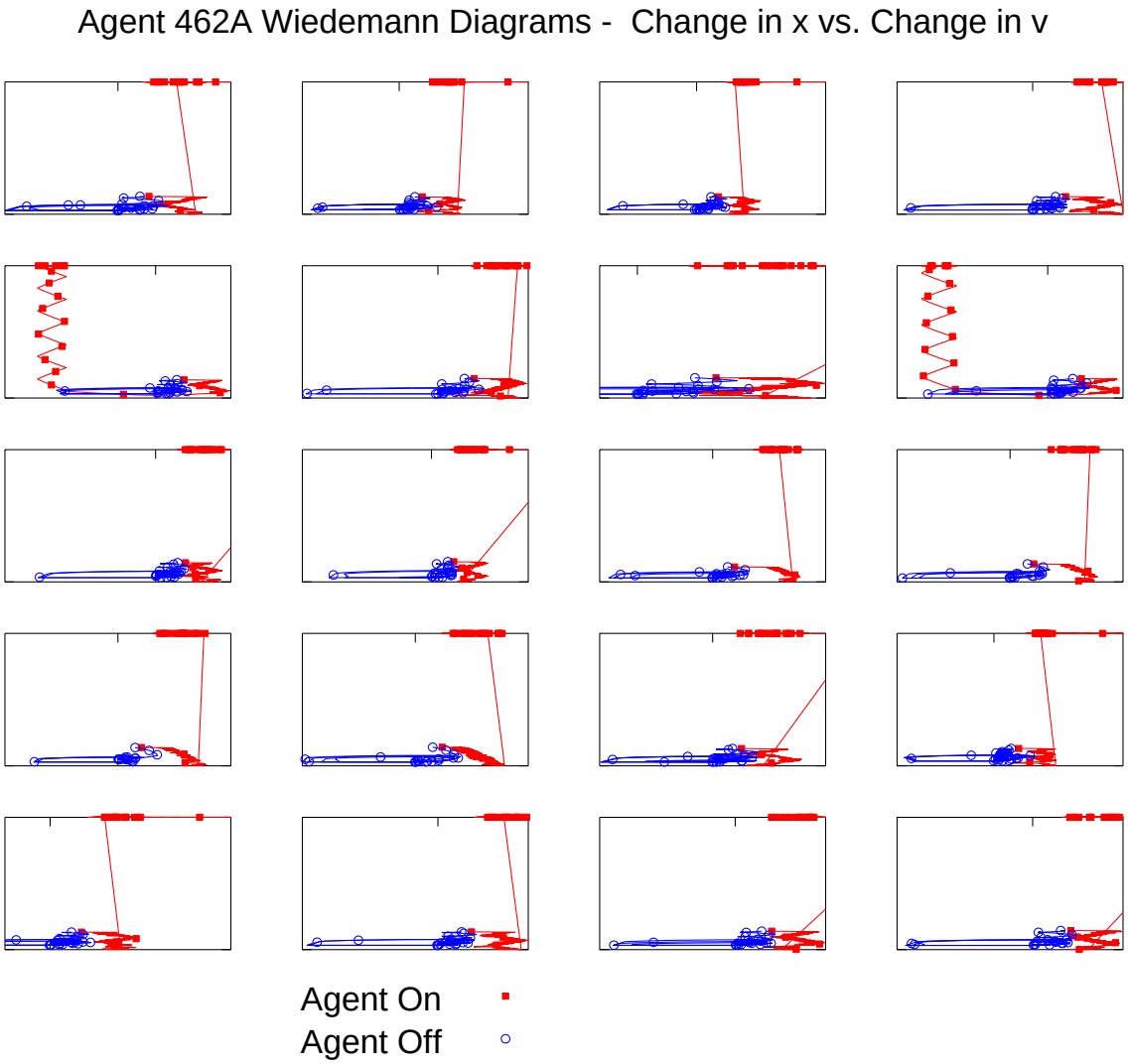

Figure 180. Agent 462A Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

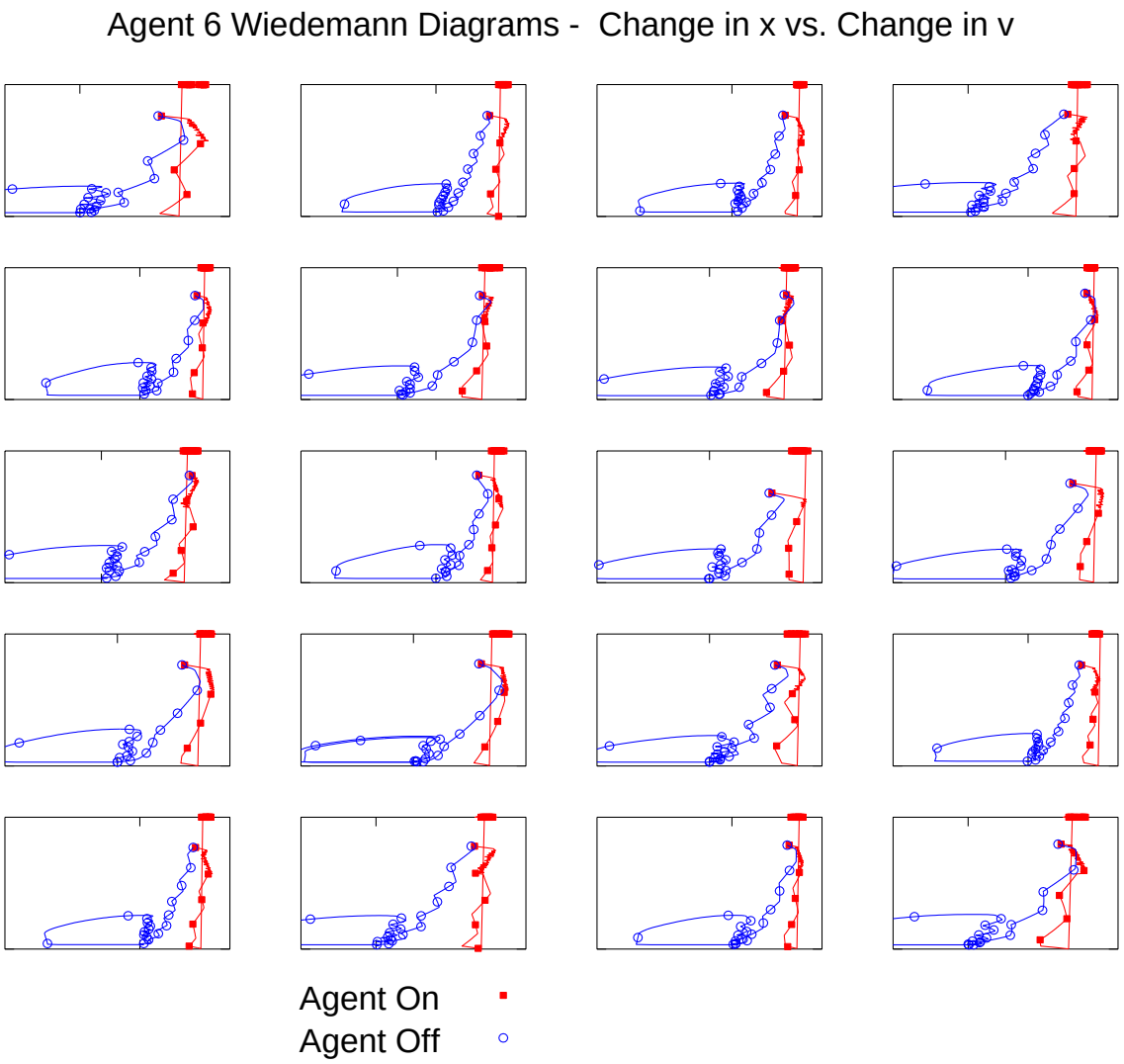

Figure 181. Agent 6 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

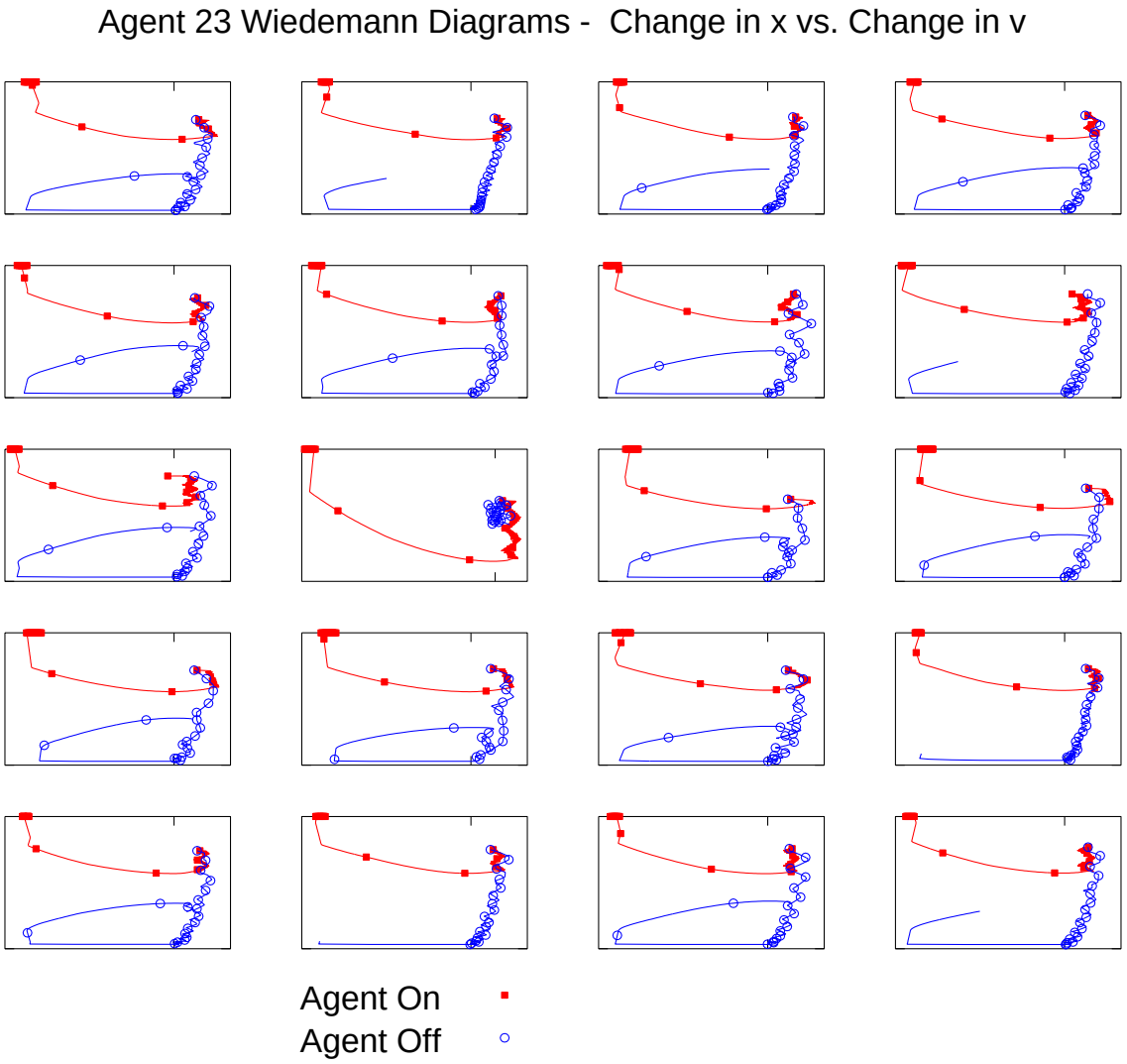

Figure 182. Agent 23 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

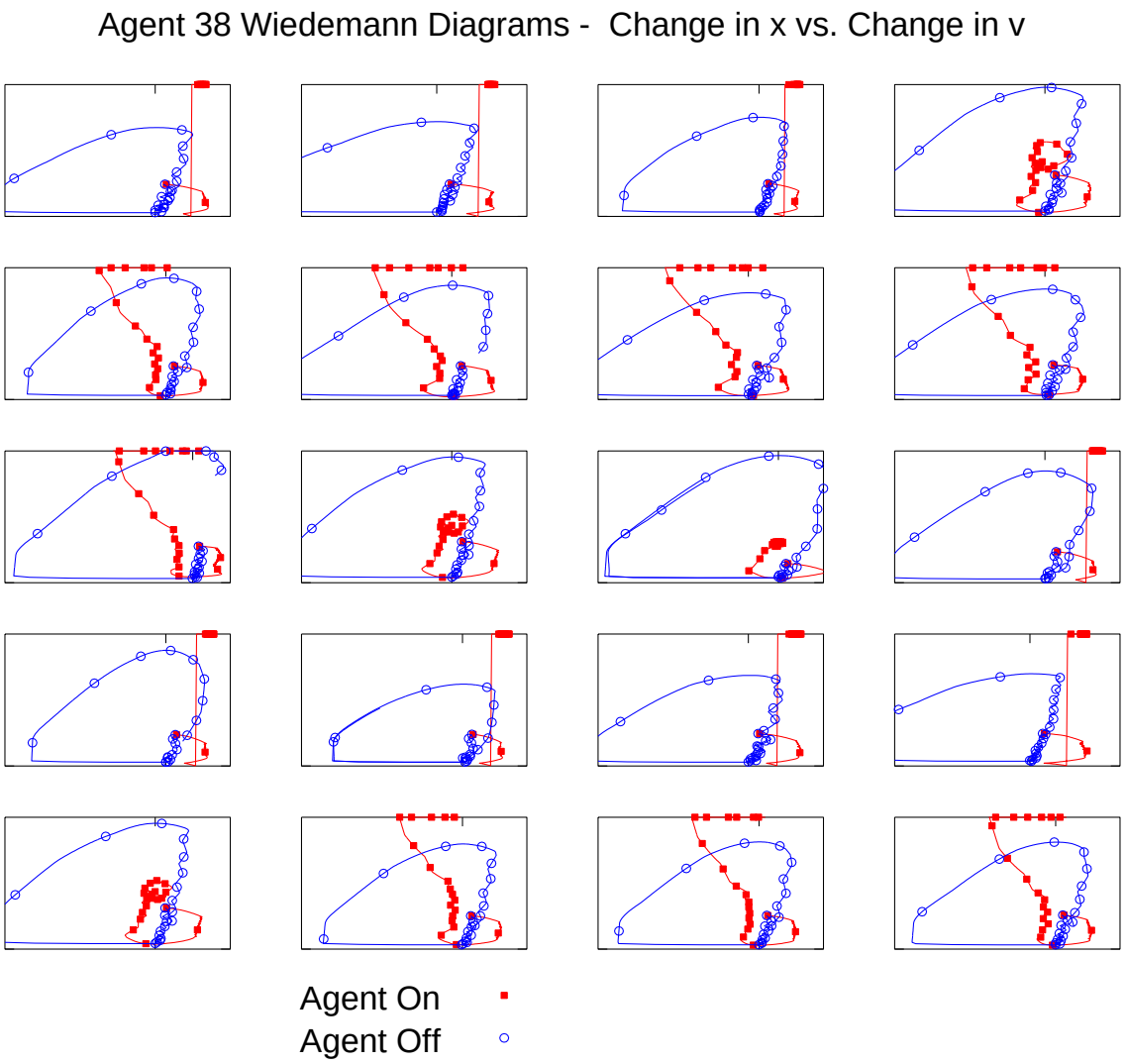

Figure 183. Agent 38 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

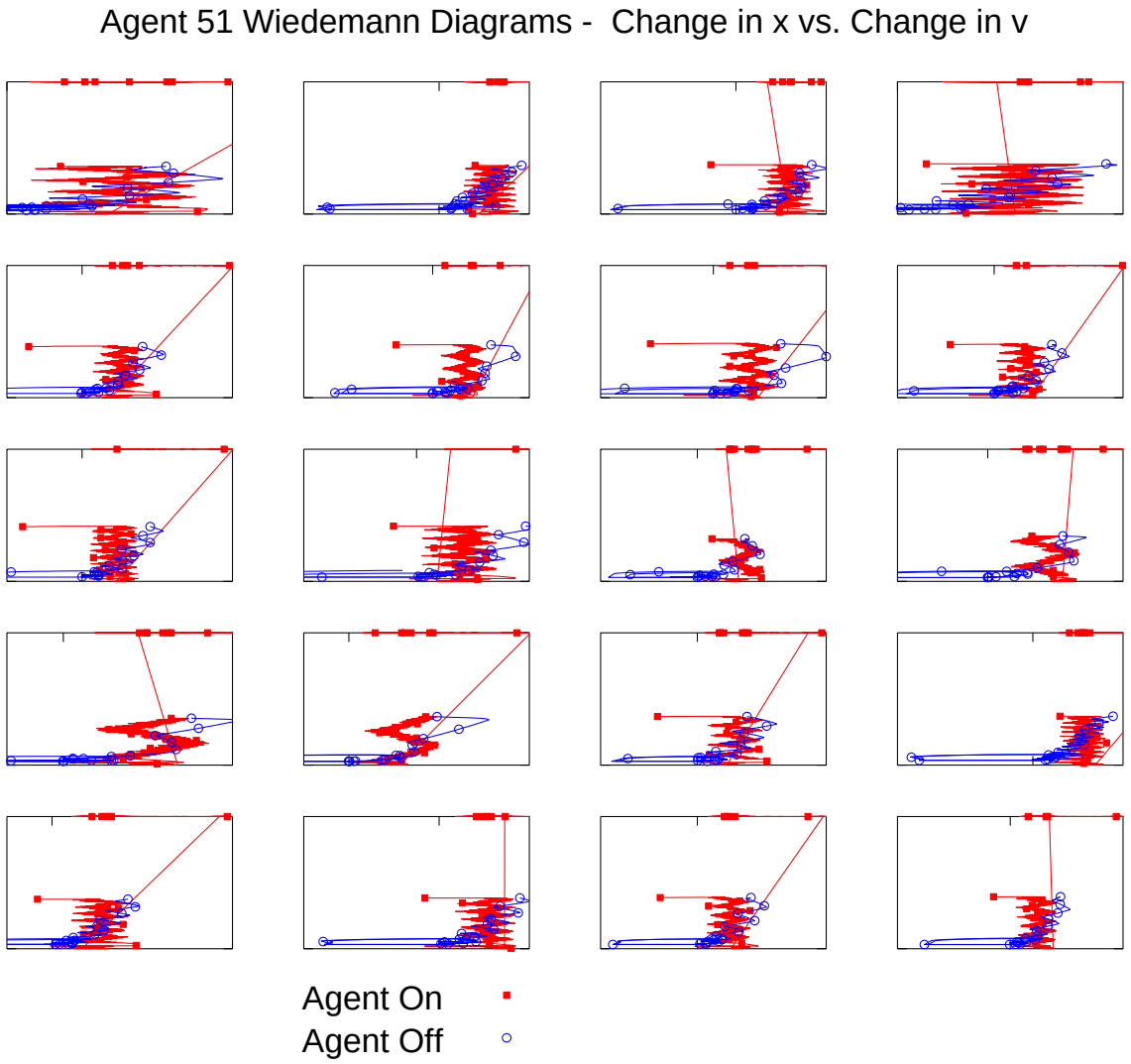

Figure 184. Agent 51 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

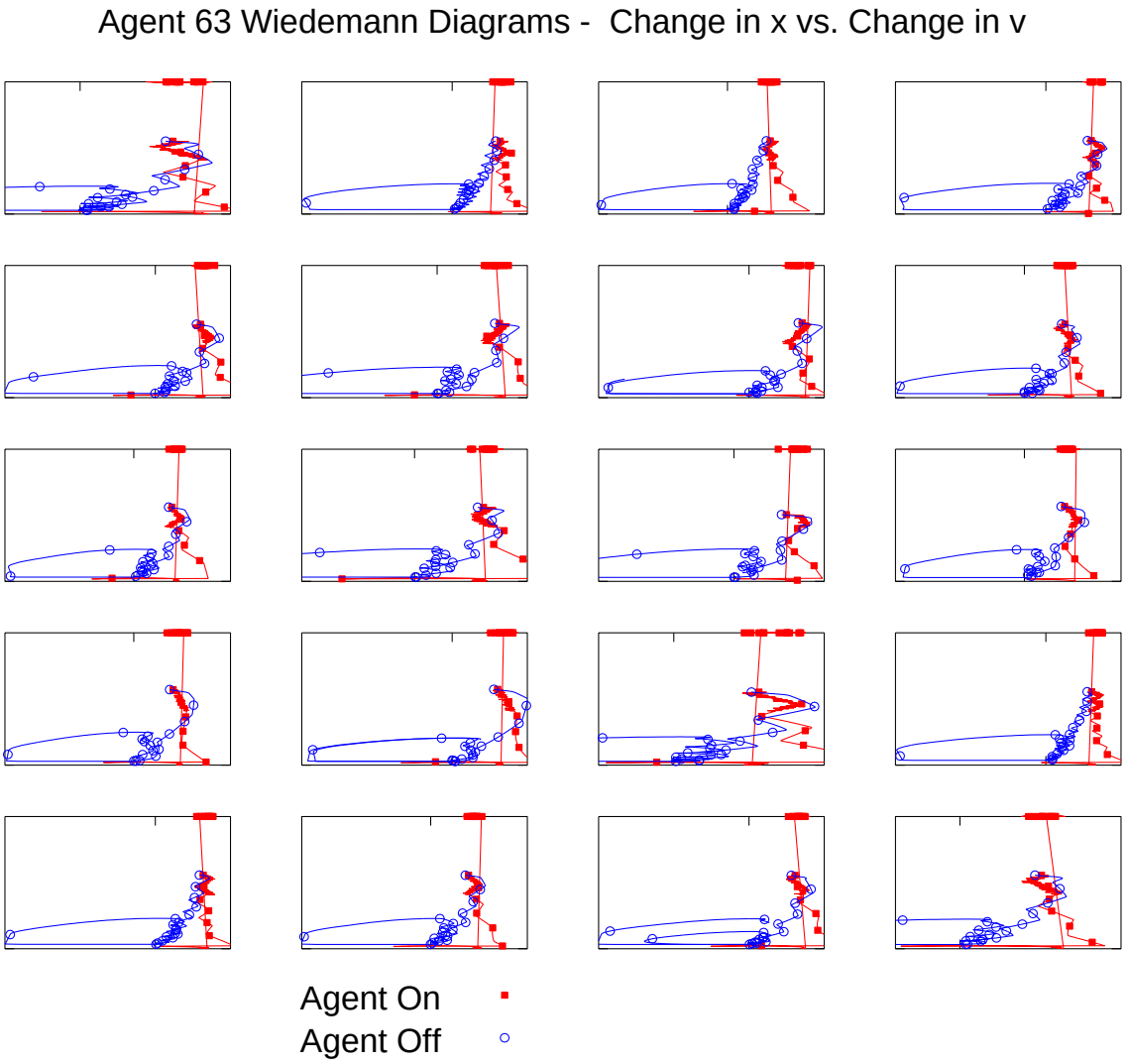

Figure 185. Agent 63 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

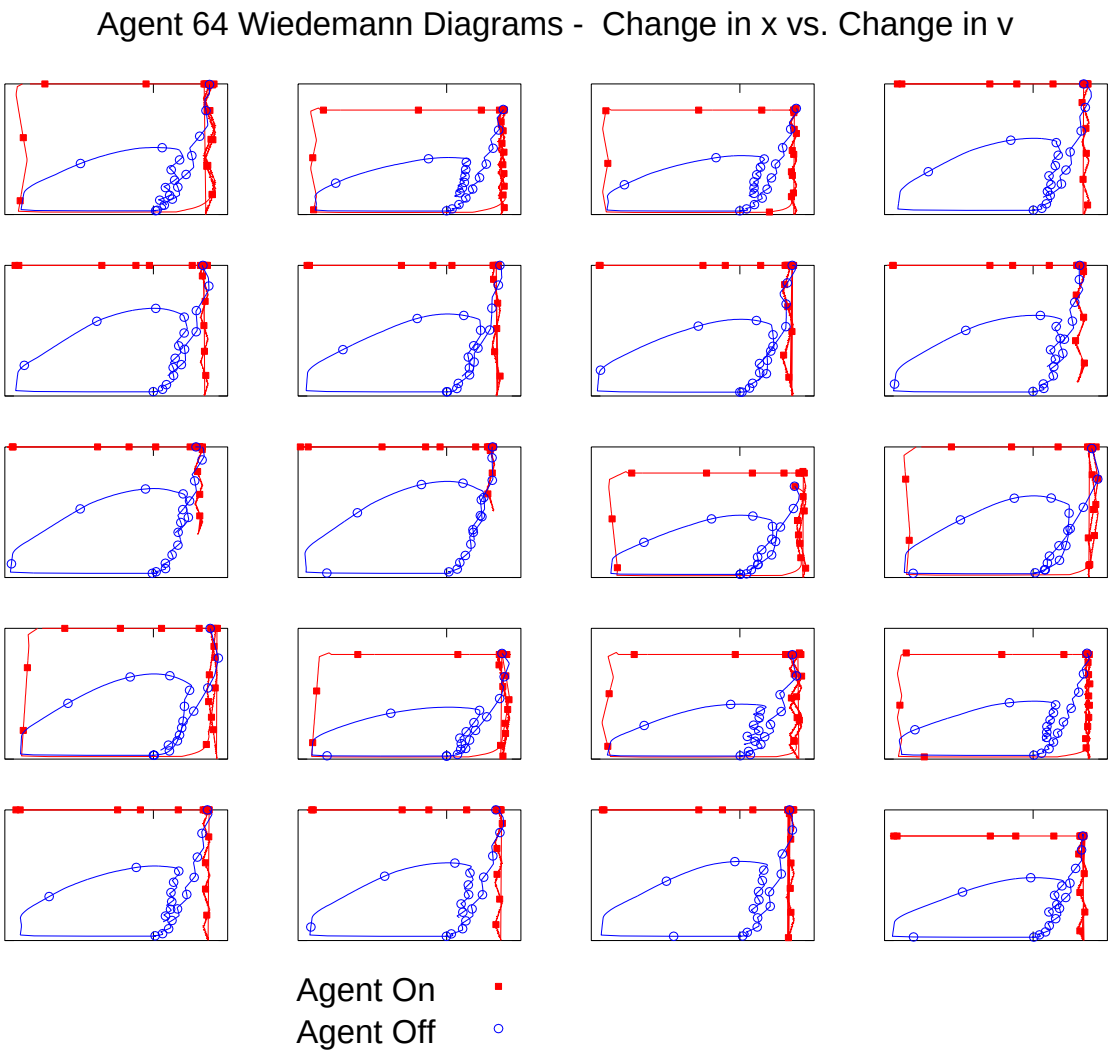

Figure 186. Agent 64 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

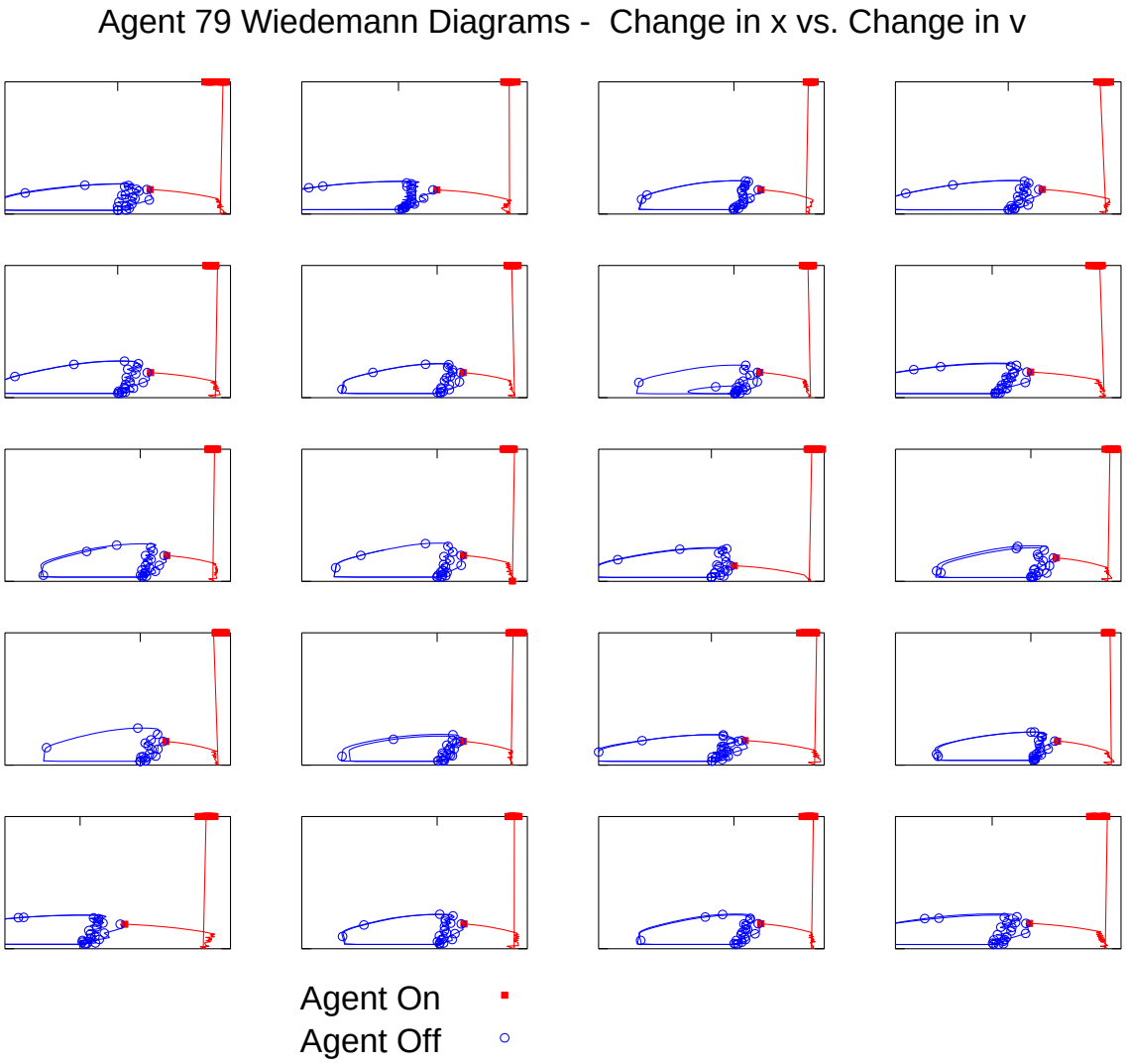

Figure 187. Agent 79 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

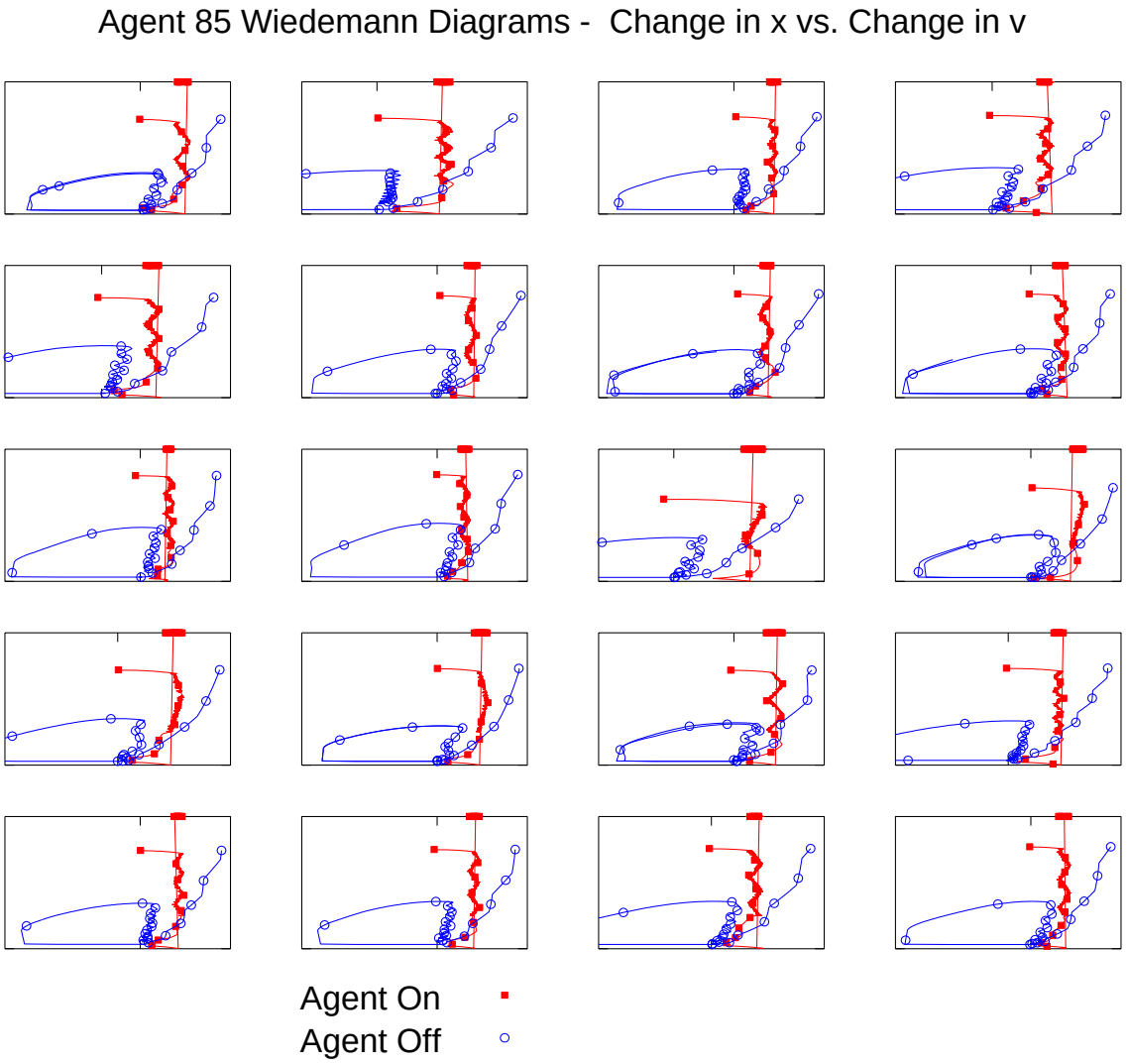

Figure 188. Agent 85 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

Figure 189. Agent 249 Wiedemann diagrams (difference in velocity on x-axis, gap distance on y-axis).

ACKNOWLEDGEMENTS

The authors would like to express thanks to Dr. C.Y. David Yang, the FHWA Agreement Manager, for his continued support and guidance during this project. They would also like to thank individuals at Virginia Tech and the Virginia Tech Transportation Institute who contributed to the study in various ways: Greg Fitch, Shane McLaughlin, Kelly Stanley and Rebecca Olson.

REFERENCES

Abbeel, P., A. Coates, T. Hunter and A. Y. Ng (2008). "Autonomous Autorotation of an RC Helicopter." 14.

Abbeel, P., A. Coates, M. Quigley and A. Y. Ng (2007). "An Application of Reinforcement Learning to Aerobatic Helicopter Flight." 8.

Abbeel, P., V. Ganapathi and A. Y. Ng (2006). "Learning Vehicular Dynamics, with Application to Modeling Helicopters." NIPS: 8.

Abbeel, P. and A. Y. Ng (2004). "Apprenticeship learning via Inverse Reinforcement Learning." International Conference on Machine Learning.

Abdulhai, B., R. Pringle and G. J. Karakoulas (2003a). "Reinforcement learning for True Adaptive Traffic Signal Control." Journal of Transportation Engineering **May/June**: 8.

Abdulhai, B., R. Pringle and G. J. Karakoulas (2003b). "Reinforcement learning for true adaptive traffic signal control." Journal of Transportation Engineering **129**(3): 278-285.

Adam, Z. M., M. M. Abbas and P. Li (2009). "Evaluating Green-Extension Policies with Reinforcement Learning and Markovian Traffic State Estimation." Transportation Research Record(No.2128): 9.

Ahmed, K. I. (1999). "Modeling Drivers' Acceleration and Lane Changing Behavior." PhD Dissertation.

AIACA-Committee (2007). Artifical Intelligence in Transportation. Artifical Intellegence. T. R. Board. Washington, DC**:** 88.

Arentze, T. and H. Timmermans (2003). "Modeling Learning and Adaptation Processes in Activity-travel choice." Transportation **30**: 26.

Avineri, E. and J. N. Prashker (2005). "Sensitivity to Travel Time Variability: Travelers' Learning Perspective." Transportation Research **Part C**: 27.

Bingham, E. (2001). "Reinforcement learning in neurofuzzy traffic signal control." European Journal of Operational Research **131**(2): 232-241.

Bogers, E. A. I., M. Bierlaire and S. P. Hoogendoorn (2007). "Modeling Learning in Route Choice." Transportation Research Record **2014**: 8.

Bonabeau, E. (2002). "Agent-based modeling: Methods and techniques for simulating human systems." **99**.

Boussier, J. M., P. Estraillier, D. Sarramia and M. Augeraud (2006). Using agent-based of driver behavior in the context of car park optimization. 2006 3rd International IEEE Conference Intelligent Systems, IS'06, September 4, 2006 - September 6, 2006, London, United kingdom, Institute of Electrical and Electronics Engineers Inc.

Brackstone, M. (2004). Driver Psychological Types and Car Following:Is There a Correlation? Results of a Pilot Study.

Brockfeld, E., R. D. Kuhne and P. Wagner "Calibration and Validation of Microscopic Traffic Flow Models." 9.

Brunet, C.-A., R. Gonzalez-Rubio and M. Tetreault (1995). Multi-agent architecture for a driver model for autonomous road vehicles. Proceedings of the 1995 Canadian Conference on Electrical and Computer Engineering. Part 1 (of 2), September 5, 1995 - September 8, 1995, Montreal, Can, IEEE.

Champion, A., M.-Y. Zhang, J. M. Auberlet and S. Espie (2002). Behavioral simulation: Towards high-density network traffic studies. 3rd International Conference on Traffic

and Transportation Studies, ICTTS 2002, July 23, 2002 - July 25, 2002, Guilin, China, American Society of Civil Engineers.

Chandler, R. E., R. Herman and E. W. Montroll (1958). "Traffic Dynamics: Studies in Car Following." Operations Research **6**(2): 165-184.

Chatterjee, I. and G. A. Davis (2010). "Bayesian Trajectory-Based Reconstruction of Rear-Ending Events Using Naturalistic Driving Data." TRB 2010 Annual Meeting CD-ROM: 18.

Chen, B., H. H. Cheng and J. Palen (2009). "Integrating Mobile Agent Technology with Multi-agent systems for Distributed Traffic Detection and Management Systems." 10.

Chiabaut, N., L. Leclercq and C. Buisson (2010). "Analysis of driving behavior heterogeneity in experimental NG-Sim data set." TRB 2010 Annual Meeting CD-ROM.

Choy, M. C., R. L. Cheu, D. Srinivasan and F. Logi (2003). "Real-Time Coordinated Signal Control Through Use of Agents with Online Reinforcement Learning." Transportation Research Record(1836): 64-75.

Coates, A., P. Abbeel and A. Y. Ng (2009). "Apprenticeship Learning for Helicopter Control." Communications of the ACM **52**(7): 9.

Dia, H. (2002). "An agent-based approach to modelling driver route choice behaviour under the influence of real-time information." Transportation Research Part C: Emerging Technologies **10**(5-6): 331-349.

Dingus, T. A., S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. A. Perez, J. Hankey, D. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland and R. R. Knipling (2006). The 100-car naturalistic Driving Study; Phase II - Results of the 100-Car Field Experiment. Blacksburg, VA, Virginia Tech Transportation Institute**:** 422.

Doniec, A., R. Mandiau, S. Piechowiak and S. Espie (2008). "A behavioral multi-agent model for road traffic simulation." Engineering Applications of Artificial Intelligence **21**(8): 1443-1454.

FHWA (2005). "NGSIM Task E.1-1: Core Algorithms Assessment." Final Report.

Fitch, G. M., S. E. Lee, S. Klauer, J. Hankey, J. Sudweeks and T. Dingus (2009). Analysis of Lane-Change Crashes and near-Crashes. Blacksburg, VA, Virginia Tech Transportation Institute**:** 88.

Fitch, G. M., H. Rakha, M. Arafeh, M. Blanco, S. Gupta, R. Zimmermann and R. Hanowski (2008). Safety Benefit Evaluation of a Forward Collision Warning System: Final Report. Blacksburg, VA, Virginia Tech Transportation Institute**:** 116.

Forbes, T. W. (1963). "Human Factor consideration in traffic flow theory." Highway Research Board Record **15. HRB**: 60.

Fritzsche, H.-T. (1994). "A model for traffic simulation simulation " Transportation Engineering Contrinution **5**.

Gazis, D. C., R. Herman and R. B. Potts (1959). "Car-Following Theory of Steady-State Traffic Flow." Operations Research **7**(4): 499-505.

Gazis, D. C., R. Herman and R. W. Rothery (1961). "Nonlinear Follow-The-Leader Models of Traffic Flow." Operations Research **9**(4): 545-567.

Ghazavi, S. A., A. Zavabeti and M. Sarvi (2008). Development of a New Driver Behavior Model: Part 1: Development of a Platform to Study Car-Following Behavior Using Real Traffic Data, TRB 2009 Annual Meeting**:** 17.

Gipps, P. G. (1981). "A Behavoural Car-Following Model for Computer Simulation." Transportation Research Board **15B**(2-C): 7.

Hackney, J. (2009). A Model for Coupling Multi-Agent Social
Interactions and Traffic Simulation. TRB 2009 Annual Meeting CD-ROM. Washington, DC.
Hadas, Y. and A. Cedar (2008). "Multiagent Approach for Public Transit System Based on Flexible Routes." 8.
Hamdar, S. H. and H. S. Mahmassani (2008a). Driver Car Following Behavior: From a Discrete Event Process to a Continuous Set of Episodes. College Park, University of Maryland**:** 37.
Hamdar, S. H. and H. S. Mahmassani (2008b). "From existing accident-free car-following models to colliding vehicles exploration and assessment." Transportation Research Record(2088): 45-56.
Hamdar, S. H. and H. S. Mahmassani (2010). "Individual Variations versus Collective Traffic Patterns: Heterogeneity Effect in a Risk-Taking Environment." TRB 2010 Annual Meeting CD-ROM: 20.
Hamdar, S. H., M. Treiber and H. S. Mahmassani (2009). "Calibration of a Stochastic Car-Following Model Using Trajectory Data: Exploration and Model Properties." TRB 2009 Annual Meeting CD-ROM: 19.
Han, Q. and H. Timmermans (2006). "Interactive Learning in Transportation Networks with Uncertainty, Bounded Rationality, and Strategic Choice Behavior." Transportation Research Board **1964**: 8.
Hanowski, R. J., M. Blanco, A. Nakata, J. Hickman, W. A. Schaudt, M. Fumero, R. Olson, J. Jermeland, M. Greening, G. T. Holbrook, R. R. Knipling and P. Madison (2008). The Drowsy Driver Warning System Field Operational Test: Data Collection Methods. Blacksburg, VA, Center for Truck and Bus Safety
Virginia Tech Transportation Institute**:** 295.
Herman, R. and R. B. Potts (1959). "Single lane traffic flow theory and experiment " Proceedings of the Symposium on the Theory of Traffic Flow.
Herrey, E. M. J. and H. Herrey (1945). "Principles of Physics Applied to Traffic Movements and Road Conditions." American Journal of Physics **13**(1): 1-14.
Holland, J. (1988). "The Global Economy as an Adaptive Process." The Economy As a an Evolving Complex System.
Hoogendoorn, S. and R. Hoogendoorn (2010). "A Generic Calibration Framework for Joint Estimation of Car Following Models using Microscopic Data." TRB 2010 Annual Meeting CD-ROM: 17.
Jacob, C. and B. Abdulhai (2006). "Automated adaptive traffic corridor control using reinforcement learning: Approach and case studies." Transportation Research Record(1959): 1-8.
Jiang, R., Q. Wu and Z. Zhu (2001). "Full velocity difference model for a car-following theory." Physical Review E - Statistical, Nonlinear, and Soft Matter Physics **64**(Compendex): 017101/017101-017101/017104.
Kesting, A. and M. Treiber (2008). "Calibrating Car-Following Models using Trajectory Data: Methodological Study." 17.
Kesting, A., M. Treiber and D. Helbing (1999). "General Lane-Changing Model MOBIL for Car-Following Models." Transportation Research Record: 9.
Kiefer, R. J., D. LeBlanc, M. Palmer, J. Salinger, R. Deering and M. Shulman (1999). Forward Collision Warning Systems: Development and Validation of Functional
Definitions and Evaluation Procedures for Collision Warning/Avoidance Systems.

Klauer, S. G., T. A. Dingus, V. L. Neale, J. D. Sudweeks and D. J. Ramsey (2006). The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data. Blacksburg, VA**:** 226.

Knipling, R. R., L. Boyle, J. Hickman, J. York, C. Daecher, E. C. B. Olson and T. Prailey (2004). "Commercial truck and bus safety-Synthesis 4-Individual differences and the "high-risk" commercial driver." Transportation Research Board.

Krauss, S., P. Wagner and C. Gawron (1997). "Metastable states in a microscopic model of traffic flow." Physical Review E. Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **55**(Compendex): 5597-5597.

Lan, L. W., C. Yu-Chiun, L. Zih-Shin and H. Chih-Cheng (2009). "A refined cellular automaton model to rectify impractical vehicular movement behavior." Physica A **388**(Copyright 2010, The Institution of Engineering and Technology): 3917-3930.

Lee, S. E., E. C. B. Olson and W. W. Wierwille (2004). A Comprehensive Examination of Naturalistic Lane-Changes. Blacksburg, VA, Virginia Tech Transportation Institute**:** 232.

Leutzbach, W. (1988). "Introduction to the theory of traffic flow."

Martens, K. and I. Benenson (2008). Evaluating urban parking Policies with Agent-Based Model of Driver Parking Behavior. T. R. Board. Washington, D.C., Transportation Research Board**:** 8.

McLaughlin, S. B., J. Hankey, T. Dingus and S. Klauer (2009a). Development of an FCW Algorithm Evaluation Methodology With Evaluation of Three Alert Algorithms. Blacksburg, VA, Virginia Tech Transportation Institute**:** 120.

McLaughlin, S. B., J. Hankey, S. Klauer and T. Dingus (2009b). Contributing Factors to Run-Off-Road Crashes and Near-Crashes.

Menneni, S., C. P. D. Sun, P.E. and P. Vortisch (2008). "An Integrated Microscopic and Macroscopic Calibration for Psycho-Physical Car Following Models " TRB 2009 Annual Meeting CD-ROM 17.

Nagel, K. and M. Schreckenberg (1992). "A Cellular Automaton Model for Freeway Traffic." Journal of Physics I **2**.

Nakayama, S. (2008). "An Analysis of Transportation System mechanism using the Agent-Based Simulation." 9.

Newell, G. F. (2002). "A simplified car-following theory: a lower order model." Transportation Research Part B: Methodological **36**(3): 195-205.

Ng, A. Y., A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger and E. Liang (2004). "Autonomous Inverted Helicopter Flight via Reinforcement Learning." 7.

Olson, R., R. J. Bocanegra, R. Hanowski and J. Hickman (2009). DRIVER DISTRACTION IN COMMERCIAL VEHICLE OPERATIONS. Blacksburg VA, Center for Truck and Bus Safety

Virginia Tech Transportation Institute**:** 285.

Olstam, J. J. and A. Tapani (2004). Comparison of Car-following models, VTI Meddelande 990A, Swedish National Road Transport Research Institute. **Project Code 40503 and 40485, 2004**.

Osaki, H. (1993). "Reaction and anticipation in the Car-Following Behavior." In Proceedings of the 12 International Symposium on the Theory of Traffic Flow and Transportation.

Ossen, S. and S. P. Hoogendoorn (2004). Car-following Behavior Analysis from Microscopic Trajectory Data

Ossen, S. and S. P. Hoogendoorn (2008). "Validity of trajectory-based calibration approach of car-following models in presence of measurement errors." Transportation Research Record(2088): 117-125.

Panwai, S. and H. Dia (2005). "Comparative evaluation of microscopic car-following behavior." IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS **6**(3): 314-325.

Panwai, S. and H. Dia (2007). "Neural Agent Car-Following Models." IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS **8**(1): 11.

Pipes, L. A. (1953). "Operational analysis of traffic dynamics." Journal of Applied Physics **24**(3): 274-281.

Punzo, V. and A. Tripodi (2007). "Steady-state solutions and multiclass calibration of gipps microscopic traffic flow model." Transportation Research Record(1999): 104-114.

Rakha, H., C. C. Pecker and H. B. B. C. Cybis (2007). "Calibration Procedure for Gipps Car-Following Model." 13.

Rakha, H. and W. Wang (2009). "Procedure for Calibrating the Gipps Car-following Model." TRB 2009 Annual Meeting CD-ROM: 21.

Ranjitkar, P. and T. Nakatsuji (2010). "A TRAJECTORY BASED ANALYSIS OF DRIVERS' RESPONSE IN CAR FOLLOWING SITUATIONS." TRB 2010 Annual Meeting CD-ROM: 21.

Rieser, M., K. Nagel, U. Beuck, M. Balmer and J. Rumenapp (2007). "Agent-oriented coupling of activity-based demand generation with multiagent traffic simulation." Transportation Research Record(2021): 10-17.

Roozemond, D. A. (2001). "Using intelligent agents for pro-active, real-time urban intersection control." European Journal of Operational Research **131**(2): 293-301.

Rossetti, R. J. F. and R. Liu (2005). "An agent-based approach to assess drivers' interaction with pre-trip information systems." Journal of Intelligent Transportation Systems: Technology, Planning, and Operations **9**(1): 1-10.

S.J. Brunson, E.M. Kyle, N. C. Phamdo and G. R. Preziotti (2002). "Alert algorithm development program NHTSA rear-end collision alert algorithm."

Schultz, G. G. and L. R. Rilett (2004). "Analysis of Distribution and Calibration of Car-Following Sensitivity Parameters in Microscopic Traffic Simulation Models." 11.

Sen, S. and K. L. Head (1997). "Controlled optimization of phases at an intersection." Transportation Science **31**(1): 5-17.

Siuhi, S. and M. Kaseko (2010). "PARAMETRIC STUDY OF STIMULUS-RESPONSE BEHAVIOR FOR CAR-FOLLOWING MODELS." TRB 2010 Annual Meeting CD-ROM.

Subranmanian, H. (1996). "Estimation of Car-Following Models." Master Thesis.

Sun, D. J. and L. Elefteriadou (2010). "A DRIVER BEHAVIOR BASED LANE-CHANGING MODEL AND ITS IMPLEMENTATION IN CORSIM." TRB 2010 Annual Meeting CD-ROM: 19.

Toledo, T. (2003). "Integrating Driving Behavior." PhD Dissertation,.

Toledo, T., H. N. Koutsopoulos and M. E. Ben-Akiva (2007). "Integrated Driving Behavior Modeling." 17.

Toledo, T. and D. Zohar (2007). "Modeling duration of lane changes." Transportation Research Record(1999): 71-78.

Treiber, M., A. Hennecke and D. Helbing (2000). "Congested traffic states in empirical observations and microscopic simulations." Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics **62**(Compendex): 1805-1824.

Treiber, M., A. Kesting and D. Helbing (2005). "Delays, Inaccuracies and Anticipation in Microscopic Traffic Models." 28.

Wahba, M. and A. Shalaby (2005). "Multiagent learning-Based Approach to Transit Assignment Problem: A Prototype." Transportation Research Record **1926**: 10.

Wang, F., M. Yang and R. Yang (2008). "Simulation of Multi-Agent based Cybernetic Transportation System." Simulation Modelling Practice and Theory **16**(10): 1606-1614.

Wang, H., W. Wang, J. Chen and M. Jing (2010). "Analysis of Intradriver Heterogeneity in Car-following with Trajectory Data." TRB 2010 Annual Meeting CD-ROM: 20.

Wiedemann, R. (1974). "Simulation des Strassenverkehrsflusses."

Wiedemann, R. and U. Reiter (1992). "Microscopic Traffic Simulation, the simulation."

Xin, W., J. Hourdos, P. Michalopoulos and G. Davis (2008). "The less-than-perfect driver: A model of collision-inclusive car-following behavior." Transportation Research Record(Compendex): 126-137.