A comparison of driving characteristics and environmental characteristics using factor analysis and k-means clustering algorithm

Heejin Jung

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Civil and Environmental Engineering

Antoine G. Hobeika

Shinya Kikuchi
Montasir M. Abbas
Linbing Wang

August 8, 2012
Blacksburg, VA

Keywords: NGSIM, driving characteristics, factor analysis, k-means clustering, CMEM

Copyright 2012

A comparison of driving characteristics and environmental characteristics using factor analysis and k-means clustering algorithm

Heejin Jung

ABSTRACT

The dissertation aims to classify drivers based on driving and environmental behaviors. The research determined significant factors using factor analysis, identified different driver types using k-means clustering, and studied how the same drivers map in each classification domain. The research consists of two study cases. In the first study case, a new variable is proposed and then is used for classification. The drivers were divided into three groups. Two alternatives were designed to evaluate the environmental impact of driving behavior changes. In the second study case, two types of data sets were constructed: driving data and environmental data. The driving data represents driving behavior of individual drivers. The environmental data represents emissions and fuel consumption estimated by microscopic energy and emissions models. Significant factors were explored in each data set using factor analysis. A pair of factors was defined for each data set. Each pair of factors was used for each k-means clustering: driving clustering and environmental clustering. Then the factors were used to identify groups of drivers in each clustering domain. In the driving clustering, drivers were grouped into three clusters. In the environmental clustering, drivers were clustered into two groups. The groups from the driving clustering were compared to the groups from the environmental clustering in terms of emissions and fuel consumption. The three groups of drivers from the driving clustering were also mapped in the environmental domain. The results indicate that the differences in driving patterns among the three driver groups significantly influenced the emissions of HC, CO, and NO_x. As a result, it was determined that the average target operating acceleration and braking did essentially influence the amount of emissions in terms of HC, CO, and NO_x. Therefore, if drivers were to change their driving behavior to be more defensive, it is expected that emissions of HC, CO, and NO_x would decrease. It was also found that spacing-based driving tended to produce less emissions but consumed more fuel than other groups, while speed-based driving produced relatively more emissions. On the other hand, the defensively moderate drivers consumed less fuel and produced fewer emissions.

ACKNOWLEDGEMENTS

First of all, thank God for your love and blessing. When I despaired and felt lonely, you always consoled and encouraged me. I praise lord.

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

I would like to express my deepest gratitude to my advisor, Dr. Antoine G. Hobeika, for his excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research.

I would like to thank Dr. Montasir M. Abbas, who led me to the research of the environmental impacts of driving behavior changes on the transportation systems in the field and practical issues beyond the textbooks.

I would also like to thank Dr. Shinya Kikuchi and Dr. Linbing Wang for guiding my research for the past several years and helping me to develop my background in transportation engineering.

Special thank goes to Dr. Sanghoon Bae, who is a mentor of my life, and the best supporter for my research.

I would like to thank Danny Kim, who as the best friend, was always willing to help and give his best suggestions. I could not stand it out in Blacksburg without him.
I also would like to thank Taehyoung Kim as my good partner and assistant. It would have been a lonely lab without him.

Many thanks to Dr. Sangjun Park, Dr. Gwibo Byun, Dr. Byoung-Do Kim, Won Hoi Hwang, Hyun-Hang Park, and other members in Korean community for cheering me up. Also, many thanks to Dongjin Lim, brothers, and sisters in Korean Baptist Church of Blacksburg for praying for me.

Finally, I would also like to thank my parents, two elder sisters, and younger sister. They were always supporting me and encouraging me with their best wishes.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Mobility is an essential requirement for human survival and societal interaction. Efficient mobility systems are also essential facilitators of economic development. On the other hand, economic growth has necessarily increased the demand for vehicles and transportation infrastructure, including road networks and public transit network, and has led to not only growing congestion and air pollution but also rising traffic fatalities (Metz et al., 2007). Motorized transportation consumes a quarter of total world energy use and produces a quarter of world energy-related greenhouse gas emissions (Metz et al., 2007; IEA, 2006a). Road vehicles consume more than three-quarters of total transportation energy, and light-duty vehicles (LDVs) account for almost half of all road vehicles use (Metz et al., 2007; *Fulton et al.,* 2004). About 95% of transportation energy comes from petroleum-based fuel, largely diesel and gasoline, which produce about 31% and 47% of total transportation energy respectively. They have been linked to carbon dioxide (CO_2) emissions from transportation: the amount of released CO_2 is approximately proportional to vehicle energy use. CO_2 is widely believed to be one of most important causes of the global warming. According to the Climate Change 2007 report, transportation was responsible for 23% of world energy-related greenhouse gas, and road transportation produces about three-quarters of that (IEA, 2006b). The amount of greenhouse gas from transportation has been increasing faster than from any other source over past decades. Today, many countries' concerns about transportation, therefore, likely focus on local traffic and pollution, and the global warming issue in transportation is addressed in the context of the broader goal of sustainable development.

The greenhouse gas involves CO_2, methane (CH_4), and nitrous oxide (N_2O). The combustion of various fuel types in mobile sources directly produce the greenhouse gases carbon monoxide (CO), non-methane volatile organic compounds (NMVOCs), sulfur dioxide (SO_2), particulate matter (PM), and oxides of nitrogen (NO_x). They cause or contribute to local or regional air pollution, depending on various factors, including traffic conditions, road conditions, weather conditions, vehicular conditions, drivers' conditions, and driving behavior. Road transportation energy use also depends on these factors. Over the past decades, many researchers in various fields have focused on reducing emissions and fuel consumption through optimization of these factors. Car manufacturers have endeavored to reduce emissions and fuel consumption by improving existing technology and developing new powertrains based on hybrid technology. For example, Fiat decreased about 18% of their CO_2 emissions in 2008 compared with 1995 in Europe (Fiat, 2010; ECC, 2009). They also set an overall target of reducing the average CO_2 emissions from new passenger cars to 120 g/km by 2015 and 95 g/km by 2020. This approach is complemented by an effort to look at the entire vehicle life cycle to find ways to reduce environmental impacts. However, the emissions and fuel consumption associated with driving is not just about vehicles, but also about how people drive. The behavioral change approach includes purchasing decisions of environment-friendly cars, trip frequency, choice of travel mode, driving behavior, and fuel choice. Eco-driving is considered to be one of the behavioral change approaches to reducing emissions and fuel consumption.

Background and Motivation

Efficient reduction of fuel consumption and emissions will require the integration of technological changes by vehicle manufacturers and behavioral changes by drivers. In past decades, many technological studies have been conducted in the mechanical engineering, chemical engineering, and industrial engineering fields, and they have successfully achieved a reduction of emissions and fuel consumption. However, the behavioral approach has received less attention in the transportation system engineering field over the past decades because of a lack of understanding of driving behavior, uncertainty about its effect on fuel consumption and emissions reduction, a limited number of techniques to control driving behavior, and other factors. In recent years, however, eco-driving has received more attention because it is a low-cost way to effect environmental change in the transportation system. Eco-driving is a practice of reducing emissions and fuel consumption by adopting a more efficient driving techniques no matter what the vehicle. Many studies, events, and courses related to eco-driving have proved its potential to reduce fuel consumption and emissions. It is widely estimated and accepted that ordinary drivers using eco-driving techniques could reduce fuel consumption and emissions by 5 to 10% (Fiat, 2010). While there is no doubt that eco-driving reduces fuel consumption and emissions, it is doubtful that all drivers could apply eco-driving techniques all the time. It becomes necessary, then, to evaluate how much drivers are applying eco-driving techniques, which requires an environmental impact analysis of different control strategies and changes in driving behavior.

There are three fundamental characteristics of traffic flow: flow, speed, and density. Macroscopic analyses of the past decades have used loop detectors to measure these characteristics in terms of flow rate, average speed, and density rate. Traffic flow has also been observed and studied at the microscopic level, which measures time headway, individual speeds, and distance headways. However, because microscopic measures have been collected by fixed-position detectors, it has been impossible to continuously observe the driving behavior of individual vehicles and thus connect characteristics of individual drivers, like age, gender, etc., with characteristics of driving behaviors. This has limited microscopic-level analysis.

In recent years, we have been able to collect microscopic measurements with high-resolution traffic video surveillance systems, vehicle-to-vehicle and vehicle-to-infrastructure communication systems using high-performance sensors, and the on-board computer systems. The ability to continuously collect measurements for a single vehicle has made it possible to characterize individual driving behavior. This advancement in microscopic measurement will facilitate the study of individual characteristics of human drivers and improve transportation technologies that adjust to human driving behavior.

Several research programs have supplied microscopic trajectory data online. For example, the Next Generation Simulation (NGSIM) Program established by the Federal Highway Administration (FHWA) collected trajectory data of all vehicles on highways and corridors using image-capture technology to improve its micro-simulation application, CORSIM (FHWA, 2012). Because the NGSIM data includes high-resolution (10 Hz) microscopic trajectory data from loop detectors, the NGSIM data includes more detailed information about driving behaviors, including pedal control than other trajectory data. Also, because the NGSIM trajectory data has a large sample size for each 15-minute study period, the trajectory data includes information on various interactions between vehicles, traffic conditions, and macroscopic phenomena.

In another project, the Connective Vehicle Research conducted by IntelliDrive^{SM} has collected second-by-second trajectory data with environmental data using on-board devices (OBD) to study advanced vehicles and infrastructures (The IntelliDrive Michigan Tested, 2005). The IntelliDrive program is a suite of technologies and applications that use wireless communications to provide connectivity between vehicles, vehicles and roadway infrastructure, vehicles and wireless communication devices, and wireless communication devices and roadway infrastructures. This program aims to connect e-payment transactions, signal phase and timing information, vehicle-to-vehicle safety messages, infrastructure communications, real-time network data, and situation-relevant information on the wireless network. Therefore, the IntelliDrive system is expected able to include various integrated information about driving behaviors, traffic conditions, road conditions, and weather conditions. Today, many researchers expect that the new microscopic data would be remarkably helpful for improving not only our understanding of traffic phenomena like capacity drop and traffic hysteresis, but also technologies of transportation systems like microscopic simulation, human driving models, driving assistance systems, and intelligent vehicles.

Environmental issues are a growing concern worldwide, particularly in the field of transportation engineering. To consider the environmental effects of mobile sources, emission models are used to develop and evaluate transportation policies at the local, state, and federal levels (Scora et al. 2006). The agencies mostly depend on the mobile source emission-factor models MOBILE and California's Emission FACtors (EMFAC) modeling suite developed by the U.S. Environmental Protection Agency and the California Air Resources Board, respectively, to evaluate environmental effects. Both models estimate vehicle emissions based on average trip speeds and were built upon regression modeling based on a large number of Federal Test Procedure bag emission measurements. These models, however, are not essential for evaluating the effect of operational improvements upon emissions changes because they are intended to predict emission for a large regional area. Because operational improvements are more microscopic, a more fundamental emission model that estimates emissions and fuel consumption of a single vehicle is required to consider emissions related to vehicle operation: idle, steady-state cruise, acceleration/brake, and so on. Several types of modal emission models have been developed using speed-acceleration matrix, emissions map, or power-demand modal modeling approaches. Since these approaches require microscopic trajectory data including instantaneous accelerations, velocities, and positions of each vehicle in traffic flow as input arguments, the development of a microscopic emission model necessitates the study of human driving behaviors.

In past decades, modeling and simulation of driver behavior has gained a lot of interest with the implementation of several car-following models in commercial software for traffic simulation, such as AIMSUN, VISSIM, DRACULA, CORSIM, PARAMICS, and so on (Panwai et al., 2005; Mehmood et al., 2003). For example, Gipps' model is implemented in AIMESUN, SISTM, and DRACULA, and Pipe's model is implemented in CORSIM. The VISSIM and PARAMICS applications implement Wiedemann's model and Fritzsche's model, respectively. These models require calibration to field data before they are used. The nature and extent of the calibration efforts depends on the purpose of the modeling, the availability of the data, and the available resources.

To facilitate the calibration process, some simulation software allows the analyst to specify the number and behavior of several driver types and the percentage of the population that conforms to each one of these types. For example, CORSIM defines up to 10 driver types, with each type having a different "aggressiveness" level in terms of car-following behavior. VISSIM allows the analyst to define a wide range of driver types and their corresponding car-following and lane-changing parameters. These concepts are implemented in other simulation software as well, leaving it to the analyst to determine the number of driver types and calibrate their modeling parameters, which requires a wealth of field data.

The motivation behind this research can be summarized as the following:

- More attention is given to environmental impact analysis of different control strategies and changes in driving behavior on the transportation system.
- Existing microscopic data (e.g., NGSIM) provides opportunities for more in-depth analysis of driver behavior.
- Improvement in technology provides more opportunities for potentially affecting driving behavior (e.g., IntelliDrive)
- There is a need to understand whether existing techniques of classifying driver behavior based on aggressiveness is sufficient; should the drivers be classified based on environmental impact analysis rather than aggressiveness?

Research Objectives

Because it was expected that changes in driving behavior could influence driving characteristics and environmental characteristics, this research ultimately aimed to classify drivers based on driving data and environmental data. The ultimate objective of this research was to determine the relationship between driving data and environmental data. The particular objectives are the following:

- Determine significant factors or patterns in each classification domain using factor analysis.
- Identify different driver types in each domain using k-means clustering.
- Study how the same drivers map in each domain.

This research was intended to provide a guideline for calibrating simulation software using separated driving groups as input arguments and estimating emissions and fuel consumption based on the characteristics of driving behavior. The research focused on recognition of dissimilarities between individual driving patterns and individual characteristics of emissions and fuel consumption for individual drivers.

Research Process

The research consists of two study cases, each including three or four steps. Before starting either study case, I analyzed NGSIM trajectory data to understand a base traffic condition. Study case 1 had three steps: construction of variables, classification, and a comparative analysis. In the construction of variables, I proposed a new variable and then converted it to variables to be used for classification. In the classification step, I classified drivers into three groups based on the new variables. In the comparative analysis, I designed alternatives to evaluate the environmental impact of driving behavior changes.

Study case 2 included four steps: construction of variables, factor analysis, classification, and comparative analysis. In the construction of variables, I constructed two types of data sets: driving data and environmental data. The driving data included 10 variables representing driving behavior of individual drivers. The environmental data included 5 variables representing emissions and fuel consumption of individual vehicles estimated by microscopic energy and emissions models. In the factor analysis step, I searched for significant factors in each data set. I found a pair of factors for driving data and defined them based on fundamental knowledge of human driving behavior. I also found a pair of primary factors for environmental data and defined the factors based on known tail-pipe mobile emissions and mobile fuel consumption. Each pair of factors was used for each k-means clustering in the classification step. I conducted two clusterings, driving clustering and environmental clustering, and then identified groups of drivers in each clustering domain. In the driving clustering, I grouped drivers into three clusters, and labeled them as speed-based drivers' group, spacing-based drivers' group, and moderate drivers' group. In the environmental clustering, I clustered drivers into two groups, and labeled them as high emitters' group and moderate emitters' group. For the next step, I conducted three comparative analyses. First, I compared the three groups of drivers from the driving clustering to the two groups of drivers from the environmental clustering in terms of emissions and fuel consumption. Second, I set an assumption that both of speed-based and spacing-based drivers of driving clustering would be assigned into high emitters' group of the environmental clustering. I compared two labels for individual vehicles based on the assumption. Through this comparison, I determined the coincidence of the two classifications and evaluated the coincident rate between the two labels. Finally, I compared the drivers mapped in the environmental domain to study a relationship between the two clusterings. See Figure 1.

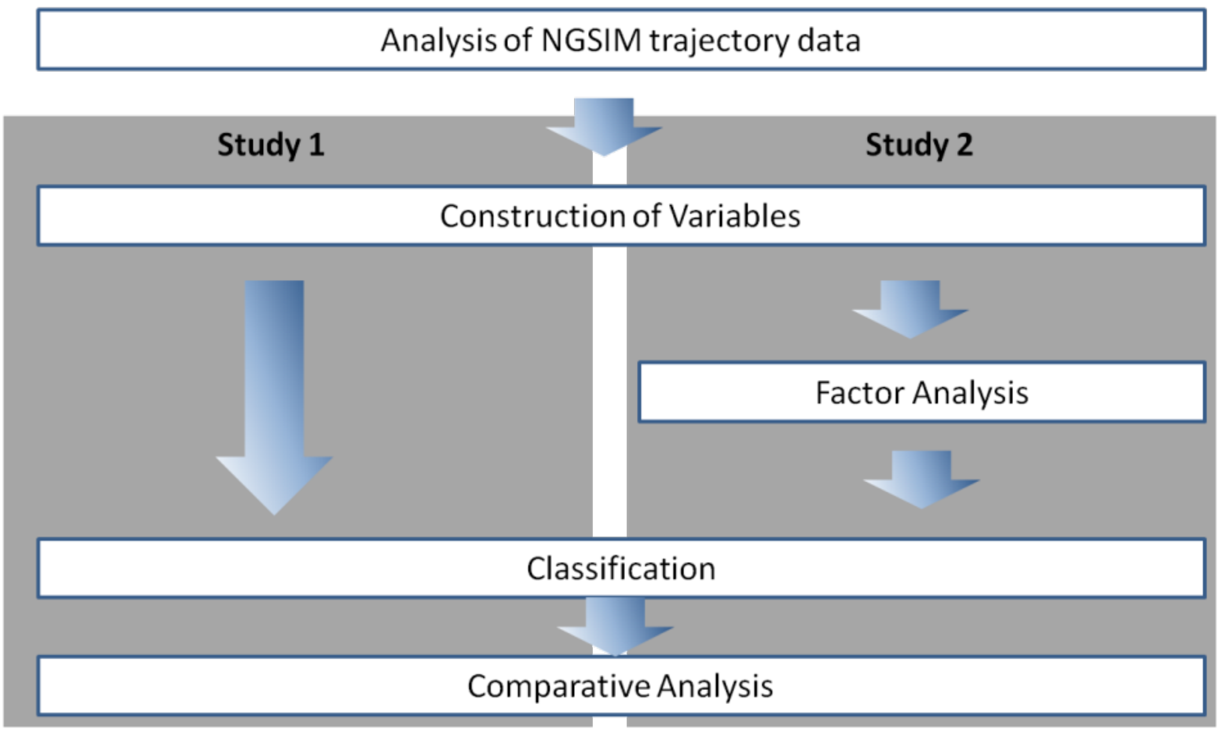


Figure 1. Flow chart of the research.

Dissertation Layout

Chapter 1 introduces the research background, research motivation, and research objectives. Chapter 2 presents the literature review. Chapter 3 briefly introduces the NGSIM program and NGSIM trajectory data. Chapter 4 and Chapter 5 present the construction of driving data and environmental data, respectively. Chapter 6 presents the research methodologies, including factor analysis and the k-means clustering algorithm, used in the dissertation. Chapter 7 and Chapter 8 present the two study cases dealing with classification of driving behavior.

Finally, Chapter 9 provides a summary of the research, a summary of findings, a discussion, the conclusions of the research, and recommended future work.

Chapter 2. Literature Review

This chapter briefly describes the NGSIM data used in this research and how it was broken into three categories. It also introduces the definitions, basic information, and major factors of aggressive drivers, mobile emissions, and fuel consumption. The chapter reviews some other research on driving behavior, emissions, and fuel consumption. Finally, it introduces multivariate data analysis.

Trajectory Data Use

The planning, design, and operation of transportation systems requires information on fundamental traffic flow characteristics and their analysis (May, 1990). Traffic flow characteristics include time headway, flow, time-space projectory, speed, distance headway, and density. Analyses include supply-demand modeling, capacity and level-of service analysis, traffic stream modeling, shock wave analysis, queuing analysis, and simulation modeling. These analyses have supported essential information and guidelines for transportation system designers, planners, and operators. Many researchers have developed models that describe traffic phenomenon because accurate traffic models are essential not only to understand traffic flow but also to estimate traffic flow changes. The improvement of transportation facilities will improve the efficiency of transportation systems. For example, in the past, many transportation planners suggested extending road capacity to mitigate congestion on the highway. The high occupancy vehicle (HOV) lane was also suggested for reduction of travel time in urban areas.

Studies of driving behavior have been conducted at both the macroscopic and microscopic levels. Both ultimately aim to describe traffic phenomenon and search for better solutions to transportation system problems such as congestion and incidents. Many models have been developed in pursuit of this goal. Many studies of driving behavior have usually used observed or simulated trajectory data at the microscopic level. Many models have been developed to mimic human driving behaviors. These models involve several parameters and one or more variables. The performance of a model totally depends on the calibration of its parameters. Many studies use trajectory data to calibrate a driving model or to verify their proposed model agasint existing models. Real-world trajectory data should be used with great caution. Raw data generally involves data noise and errors coming from devices, data mining process, and so on. A reduction of these noises would be one way to improve trajectory data prior to applying it. Punzo et al. (2009) investigated the error in the observed data of NGSIM trajectory data using jerk analysis, platoon consistency analysis, and internal consistency analysis (Punzo et al., 2009). They clarified the concepts underlying the problem of estimation of trajectories from discrete observations, particularly observed positions. They suggested post-processing to minimize the effects of measurement errors. They carried out a comprehensive review of the filtering techniques applied to trajectory data and pointed out the requirements of any estimation technique. They used jerk analysis to infer actual errors of acceleration data, found that accelerations are not the best measurements available, and recommended that they not be used without proper processing. They also evaluated platoon consistency and internal consistency using spectral analysis. They found significant errors in acceleration data and suggested very careful use of NGSIM trajectory data.

Filtering techniques, smoothing techniques (a subset of filtering), and averaging techniques (a subset of smoothing) have been widely used to mitigate the noise in coordinates, trajectories, speed, acceleration, and inter-vehicle spacing. Filtering is just a general name given to the process of systematically modifying data arranged in a sequence or array (Rayner, 1971). Smoothing replaces the low-scale or moderately medium-scale with the high-scale and partially removes the medium-scale disturbances from original data. Thiemann et al. proposed a smoothing algorithm to decrease the noise in the NGSIM information on positions, velocities, and accelerations (Thiemann et al., 2008). They estimated a smoothing time interval based on velocity time series and the variance of the processed acceleration time series. They used this information to calculate the distribution of the density function of velocity, inverse distance, time gaps, and times-to-collision classified by several ranges of velocities and velocity differences. In addition, they investigated the lane-changing process and formulated a quantitative criterion for lane changes. They also used the trajectory data to obtain parameters required to calibrate car-following models and lane-changing models for the microscopic analysis of single vehicle movements.

A general way to calibrate existing models is to apply trajectory data. Calibration is one of the optimization methods that minimize the error function between observed data and estimated data. In calibration of a human driving model, trajectory data is usually used as observed data and is compared with estimated trajectory data produced by the human driving model.

Talebpour et al. proposed and extended the sequential risk-taking model introduced by Hamdar et al. and captured the effects of surrounding traffic conditions on driving behavior (Talebpour et al., 2011; Hamdar et al., 2008). They introduced different behavior (response) regimes linked to situational complexity associated with surrounding traffic conditions. The different regimes were separated according to the same situation in different surrounding traffic conditions. They calibrated the proposed model with NGSIM data and conducted sensitivity analysis with respect to the model parameters, including parameters that describe the surrounding condition. As a result, decreasing the surrounding density, decreasing the headway with followers, and having a lower speed than leaders and followers increased the tendency toward higher acceleration rates.

Kim et al. investigated the correlation of three parameters in conventional car-following models: Gipps' model, the Helly linear model, and the intelligent driver model (Kim et al., 2011). They calibrated the three models using a downhill simplex (gradient-free optimization) method and then conducted factor analysis to determine the parameters relationships. They also conducted a correlation analysis to calculate Pearson correlation coefficients as quantitative measures that can be used in the simulation sampling procedure. They then sampled parameter sets with and without correlation based on empirical distributions of calibrated parameters and simulated car-following movements for each model. They found that parameters of car-following models drawn simply from uncorrelated marginal distributions could yield unreliable results in simulation and consequently inaccurate interpretation. However, the use of parametric distributions with an estimated correlation structure may not necessarily reduce the error due to ignoring correlation if the underlying distributional assumption does not sufficiently hold for both marginal and joint distribution.

Trajectory data generally include not only trajectory of a vehicle but also various instantaneous data derived from its trajectory. The information includes vehicle identification number, type of vehicle, global locations, local locations, speeds, accelerations, and other characteristics. This information could be used to develop new driving behavior models or to discover new theories about traffic phenomenon. Trajectory data is generally used to verify new models, and verifying is very similar to calibrating. Yang et al. proposed a methodology to estimate rear-end conflict risk of vehicles on freeway merge sections as a probabilistic measure (Yang et al., 2011). They combined two components: estimation of the merging probability and estimation of the potential risk. The estimation of the merging probability is the merging probability of a vehicle given its position on a merge lane. The estimation of the potential risk is the probabilistic risk of a merging vehicle conflicting with vehicles around it as a function of a surrogate safety measure: modified time-to-collision. In the first part, they used NGSIM trajectory data to find the underlying probability density function of the merging decision. They evaluated the conflict risk of each merging vehicle at each time step, aggregated the conflict risk over time and space, and then created a risk map for describing the level of conflict risk. They demonstrated the implementation of the proposed method for traffic conflict analysis in detail in their paper, and then they concluded that their proposed methodology can be used to evaluate the safety level of merge sections and to develop real-time traffic control strategies to reduce conflicts associated with merging traffic.

Yeo et al. proposed a new integrated car-following and lane-changing model, consistent with the kinematic wave theory, for oversaturated freeway flow (Yeo et al., 2008). They developed this model as part of the NGSIM project sponsored by FHWA. Their model includes not only mandatory and discretionary lane changing with cooperation, but also a new on-ramp merging model. The proposed model also accounts for the relaxation process following lane changing. The advantage of this model is that it includes a small number of parameters able to be readily measured in the field: free-flow speed in mph, jam gap in feet, wave travel time in seconds, maximum acceleration in feet per squared second, and maximum deceleration in feet per squared second. It also has three lane-changing parameters: sensitivity to speed difference in seconds, exit lane changing parameter in feet per lane, and target distance to exit location in feet. They simulated AIMSUN SDK and installed the proposed model instead the existing car-following and lane-changing models in AIMSUN. They simulated the model on the I-80 and US-101 and then validated the model using the trajectory data and the aggregate data speeds and flows from loop detectors at each site. The validation showed that the proposed model accurately tracked the propagation of congestion. The mean error of speeds was 0.38 mph, and root-mean-square error (RMSE) was 2.08 mph. The simulated wave speeds were 11.3mph, very close to the measured wave speeds of 11.4 mph.

Sadabadi et al. proposed a finite element method (FEM) to find an approximate numerical solution of the velocity-based equivalent of the first-order continuum traffic flow model, such as speed-based Lighthill-Whitham-Richard model (LWR-v model). The FEM provided a theoretical framework to understand and analyze traffic processes on a variety of roadway facilities (Sadabadi et al., 2011). Sadabadi et al. compared the performance of the proposed FEM for an NGSIM dataset to an existing numerical method such as a finite difference method (FDM). The FDM used a standard Godunove scheme to solve the LWR-v model, similarly to a cell transmission model. The proposed FED used one-dimensional simplex elements with the assumption of a first-order interpolation function. They adopted a Galerkin approach to minimize the integral of the weighted residuals and derive the set of element characteristic matrices and vectors. To evaluate the two solution methods, they used a data set belonging to a segment of US-101 in Los Angeles, California, on June 15, 2005. Both methods provided accurate approximations of the observed speeds, with overall biases and mean absolute errors in the range of 1 to 3 mph and 4 to 6 mph, respectively. In addition, both methods produced major shock waves that passed through both the upstream and downstream well, but smaller shock waves originating somewhere between the two end boundaries did not.

Many researchers have been interested in microscopic simulation models concerning not only individual human driving behaviors, including free agent, car-following, and lane-changing, but also the impact of individual drivers' physical and psychological characteristics on their behaviors. Pueboobpaphan et. al reviewed and categorized studies presenting the relationship between individual drivers, vehicle characteristics, and traffic flow stability. They then defined traffic flow stability (Pueboobpaphan et al., 2010). Using stability analysis, they found that the traffic flow stability depends on driver characteristics.

Rong et. al examined the interactions among three factors: driver characteristics, driving behavior, and traffic flow characteristics (Rong et al., 2011). They classified drivers into three categories—aggressive, conservative, and moderate—based on driver characteristics and driving behavior using the k-means clustering algorithm. They then calibrated the driving behavior parameters of the three types of drivers using the experimental data collected in a driving simulator. The parameters included reaction time, expected speed, and critical gap for changing lanes. They analyzed the effects of the driver characteristics on the traffic flow and the traffic flow stability. They found that aggressive drivers mostly kept smaller headways and more frequently changed lanes, resulting in smaller gaps.

Winsum reviewed the driver behavior literature to discuss psychological factors in car-following behaviors and then described a model with time-to-collision as the psychological factor to cover the variation of human behavior (Winsum, 1999). Boer also discussed following issues, including task scheduling and attention management, satisficing instead of optimal performance evaluation, and perceptual rather than Newtonian input, and Boer then presented a general driving modeling framework emphasizing the car-following task (Boer, 1999).

Using simulation and traditional statistical analysis, Casucci et al. did not find significant differences in braking behaviors between drivers with different levels of experience (Casucci et al., 2010). They studied the differences in the overall braking sequence between prudent and aggressive driving behavior and found that aggressive driver completed the overall sequence in a much shorter time than did prudent drivers. They also found that sensation seekers (similar to aggressive drivers) frequently exceeded the legal posted speed limit and had much higher cruising speed than prudent drivers did.

The method of estimating driving behaviors on the basis of driver characteristics has two limitations. First, most psychological characteristics are hardly measureable and frequently change due to unpredictable factors on the roads. Second, limitations of funding and time make the sample size of subjects in such studies too small to generalize findings. Hence, many research projects have used trajectory data from microscopic simulations or examinations to model driving behaviors without accounting for psychological and physical characteristics of drivers such as age, gender, and driving experience.

Many researchers have tried to calibrate existing driving models to driving behavior data collected from the real world. Brockfeld et al. compared 11 microscopic models by calibrating and validating the models using the average speed and flow collected for four days by double loop detectors located at eight stations on a multilane I-80 freeway (Brockfeld et al., 2005). They calculated the absolute error between the aggregated speeds of the collected data and the model-simulated data, using a particular parameter set for a certain vehicle pair using Theil's U-value, and then minimized the error to calibrate these models. Lownes et al. presented a sensitivity analysis of VISSIM simulation capacity output under 10 driver behavior parameters including stopped condition distance, headway time, variation of following, threshold for entering following, upper and lower following thresholds, speed dependency of oscillation, oscillation acceleration, stopped condition acceleration, acceleration at 50 mph (80 km/h), and look-back distance (Lownes et al., 2006).

Another approach to developing microscopic models that represent human driving behaviors is separating driving behaviors into several parts. Since driving behaviors are the results of a complex decision process, it is very difficult to analyze them logically. Many researchers have tried to convert the complex driving decisions to a set of simple linear systems with the physical or psychological characteristics of individual drivers. They used clustering algorithms or statistical methods to simplify driving behavior models. Ma et al. analyzed collected trajectory data using regime boundary identification and analysis of statistical relations between perceptual variables and driver acceleration response (Ma et al., 2007). Using the fuzzy C-means clustering algorithm, they classified the car-following data into five regimes: approaching, stable following, continuous acceleration, braking, and opening. They then analyzed the statistical relations using correlation and regression methods to determine the relationships between physical variables and driver acceleration. They found that car-following behavior is generally a nonlinear process; opening, braking, and acceleration regimes can be modeled using a simple multiple linear regression method; the stable following regime is mostly nonlinear; both leading and following vehicles are generally significantly correlated in the car-following stage; and an appropriate car-following model should not only reflect psychophysical personal difference, but also describe the behavioral difference in each regime. Sultan et al. separated the car-following data sequence of a single vehicle that into two types of interloops: opening and closing processes (Sultan et al., 2004). Rigolli et al. compared the classification performance of two automatic methods of classifying driver behavior using only data provided by vehicle trackers (Rigolli et al., 2005).

Driving Behaviors

Trajectory data can be used to characterize longitudinal driving behavior using measurable parameters at the microscopic level (*Wang, 2010*). Measureable vehicle data can be classified into three categories: vehicle context data, vehicle state data, and driver behavior data. The vehicle context data involves information such as vehicle location and road condition that is collected by laser radar, camera, and global positioning system (GPS). The vehicle state data include vehicle speed, acceleration, and yaw rate that is collected by gauge-equipped vehicles. The driver behavior data are movement of hand and foot or pedal position and steer angle, collected by monitoring cameras or sensors. Collected vehicle data are synchronized and aggregated by a driver data analysis tool and then converted to parameters, including mean value and standard deviation of mean time headway, time to collision, mean value and standard deviation of time headway, mean value of maximal acceleration and deceleration, mean value of brake pedal activation and accelerator pedal activation, mean value of accelerator pedal release, and mean delay time from accelerator release to brake activation.

Basically, because human behavior is a nonlinear process, predictions of human driving behavior are typically unacceptable (*Ma et al., 2007*). However, classification can be a selection of methods to linearize complex driving behavior for modeling driving behavior. There are two types of classification for driving behavior: classification of drivers' type and classification of regimes. The classification of drivers' type classifies drivers into several groups such as aggressive drivers and defensive drivers. The classification of regimes divides a driving behavior into several regimes such as approaching, following as well as opening regime. Ma et al. suggested statistical method of analyzing driver behavior in terms of regime data analysis to obtain more acceptable predictions of human behavior. They classified the driver behaviors into five regimes: approaching, opening, following, acceleration, and braking. They then assumed that the driving behavior of each regime is a linear process. They used a statistical method (correlation and regression analysis) to classify driving behaviors into the five regimes. Rong et. al examined the interactive relationship of three factors: driver characteristics, driving behavior, and traffic flow characteristics (*Rong et al., 2011*). They classified drivers into three categories—aggressive, conservative, and moderate—based on driver characteristics and driving behavior using the k-means clustering algorithm. Then they calibrated the driving behavior parameters of the three types of drivers using experimental data collected in a driving simulator. The parameters included the reaction time, expected speed, and critical gap for changing lanes. They analyzed the impacts of the driver characteristics on the traffic flow and the traffic flow stability. They found that aggressive drivers mostly kept smaller headways and more frequently changed lanes, with smaller gaps as a result.

However, driving behavior research using classification had a limitation caused by sample size. The sixth column of Table 1 presents the sample size of the different classification research studies. Ma et al. classified the driving patterns of 30 drivers into five regimes using the fuzzy c-means clustering algorithm (*Ma et al., 2007*). They also classified drivers into four groups using their proposed driving style classification algorithm.

Table 1. Classification methods and sample size in conventional studies.

Authors	Year	Parameters	Classification method	Clusters/categories	Sample size
Ma et al.	2007	Speed Headway distance Speed difference Acceleration	Fuzzy c-means algorithm	Approaching Opening Stable following Continuous acceleration Braking	30 drivers using one instrumented vehicle.
Murphey et al.	2009	Acceleration Jerk	Driver's driving style classification algorithm (DS_classification)	Calm driving Normal driving Aggressive driving No speed	Using driving cycles No driving data.
Wang et al.	2010	Time headway Time to collision Maximum deceleration	k-mean clustering	Prudence Stability Conflict proneness Skillfulness	398 data segments of 45 drivers
Rong et al.	2011	7 components through principal component analysis with 15 parameters	Cluster analysis similar to k-means algorithm	Aggressive driver Conservative driver Moderate driver	32 drivers

Rong et al. determined the relationship between driving behavior and traffic flow (*Rong et al., 2011*). They used the 2003 Beijing traffic survey data to set traffic conditions and collected 32 individual drivers' data, including age, experience, and gender. They conducted their experiment in two parts: the driver adaptive experiment and the driving simulator experiment. They collected data on 15 indicators for each driver. Then they developed seven components using principal component analysis with the 15 components and classified drivers into three classes using clustering analysis.

Murphey et al. proposed a new approach to classifying driving style using jerk analysis (*Murphey, et al., 2009*). They suggested four categories of driving style: calm driving, normal driving, aggressive driving, and no speed. They assumed that drivers can be calm sometimes and aggressive other times in one trip. Thus they separated driving regimes using a Driving Style Classification Algorithm (DS_Classification) based on jerk analysis with two thresholds.

Wang et al. characterized 45 drivers' behavior based on four categories: prudence, stability, conflict proneness, and skillfulness (*Wang, 2010*). They classified 45 drivers into two clusters for each category using the k-means clustering algorithm. Using simulation and traditional statistical analysis, Casucci et al. did not find significant differences in braking behaviors between drivers with different levels of experience (*Casucci et al., 2010*). They studied the differences in the overall braking sequence between prudent and aggressive driving behavior and found that aggressive driver completed the overall sequence in a much shorter time than did prudent drivers. They also found that sensation seekers (similar to aggressive drivers) frequently exceeded the legal posted speed limit and had much higher cruising speed than prudent drivers did.

Aggressive Drivers

When classifying driving patterns, the aggressiveness of driving behavior has been an important measure for recognizing dissimilarities between individual driving behaviors. While aggressiveness has been defined several times, the definition of aggressive driving is not clear.

Aggressive driving was defined as the following by Mizell (*Mizell, 1997*):

"An accident in which an angry or impatient motorist or passenger intentionally injures or kills another motorist, passenger, or pedestrian, or attempts to intentionally injure or kill another motorist, passenger, or pedestrian, in response to a traffic dispute, altercation, or grievance."

Aggressive driving also includes angry or vengeful motorists intentionally driving their vehicle into a building or other structure or property (*Mizell, 1997; Miles et al., 2003*).

The New York penal law defines the aggressive driving as the following:

"The unsafe operation of a motor vehicle in a hostile manner, without regard for the safety of other users of the road. Aggressive driving includes frequent or unsafe lane changes, failing to signal, tailgating, failing to yield right of way, and disregarding traffic controls (*Pataki, 1998; Miles et al., 2003*)."

NHTSA has proposed a definition of aggressive driving as the following:

"Some behaviors typically associated with aggressive driving include: exceeding the posted speed limit, following too closely, erratic or unsafe lane changes, improperly signaling lane changes, failure to obey traffic control devices(stop signs, yield signs, traffic signals, railroad grade cross signals, etc.). Law enforcement agencies should include red light running as part of their definition of aggressive driving. NHTSA calls the act of red light running as one of the most dangerous forms of aggressive driving (*Miles et al., 2003*)."

Because aggressive driving includes any unsafe driving behaviors by ordinary drivers on the road, aggressive driving has been defined in terms of traffic safety (*Lee et al., 2010*). Aggressive driving behaviors have been mostly determined by perception, with various determinants. The assessment of the aggressive driving, therefore, has been inconsistent case by case. For example, Lee et al. had 12 measures, including tailgating, passing on the shoulder, failing to yield to merging traffic, making obscene gestures. Vanlaar et al. used 6 measures: excessive speeding, fail to signal, tailgating, weaving in/out traffic, failing to stop at sign, and running red light (*Vanlaar et al., 2008*).

Another problem with assessing aggressive driving is that it is hard to convert aggressiveness of a driver to a numerical value because the aggressive driving is related not only to personality, but also to the driver's psychological circumstance (*Miles et al., 2003*). That is, the aggressive driving attitude would frequently change according to the driver's stress level, even across similar situations.

Overview of Emissions and Fuel Consumption

Current Status

Transportation activity is increasing around the world as economies grow (*Metz et al., 2007*). The most pressing problems associated with increasing transportation activity are traffic fatalities and injuries, congestion, air pollution, and petroleum dependence. These problems are especially acute in the most rapidly growing economies of the developing world. Mitigating greenhouse gas (GHG) emissions can take its place among these other transportation priorities by emphasizing synergies and co-benefits. Transportation predominantly relies on a single fossil resource, petroleum, that supplies 95% of the total energy used by world transportation. In 2004, transportation was responsible for 23% of world energy-related GHG emissions with about three-quarters coming from road vehicles. Over the past decade, transportation's GHG emissions have increased at a faster rate than any other energy-using sector.

The majority of the world's population still does not have access to personal vehicles, and many do not have access to any form of motorized transportation. However, this situation is rapidly changing. Freight transportation has been growing even more rapidly than passenger transportation and is expected to continue to do so in the future.

In 2004, the transportation sector produced 6.3 $GtCO_2$ emissions (23% of world energy-related CO_2 emissions), and its growth rate is highest among the end-user sectors. Road transportation currently accounts for 74% of total transportation CO_2 emissions. The share of non-Organisation for Economic Co-operation and Development (OECD) countries is now 36%, and much agreed-upon evidence of current trends continuing indicates it will increase to 46% by 2030. The transportation sector also contributes small amounts of CH_4 and N_2O emissions from fuel combustion and fluorinated gases (F-gases) from vehicle air conditioning. CH_4 emissions make up between 0.1 and 0.3% of total transportation GHG emissions, N_2O between 2.0 and 2.8% (based on U.S., Japan, and E.U. data only). Worldwide emissions of F-gases (CFC-12+HFC-134a+HCFC-22) in 2003 were 0.3–0.6 $GtCO_2$-eq, about 5–10% of total transportation CO_2 emissions.

Future Trends

World transportation energy use will increase at the rate of about 2% per year, with the highest rates of growth in emerging economies, and total transportation energy use and carbon emissions will be about 80% higher than current levels by 2030.

Improving energy efficiency offers an excellent opportunity for transportation GHG mitigation through 2030. Carbon emissions from new light-duty road vehicles could be reduced by up to 50% by 2030 compared to currently produced models, assuming continued technological advances and strong policies to ensure that technologies are applied to increasing fuel economy rather than horsepower and vehicle mass. The total mitigation potential of the energy efficiency options applied to light-duty vehicles would be around 0.7–0.8 $GtCO_2$-eq by 2030 at costs less than $100 (U.S.) per ton of CO_2. Data is not sufficient to provide a similar estimate for heavy-duty vehicles. The use of current and advanced biofuels could reduce emissions another 600–1,500 $MtCO_2$-eq by 2030 at costs less than $25 (U.S.) per ton of CO_2. The mitigation potential by 2030 for the transportation sector is estimated to be about 1,600–2,550 $MtCO_2$ for a carbon price less than $100 (U.S.) per ton of CO_2.

Overview of Emissions

Motor vehicles did not attract much attention as important air pollutant sources until about 1950 because there were very much larger and uncontrolled air emissions from industry (*de Nevers, 2010*). After these emissions sources were controlled, and natural gas replaced coal as the principal urban heating fuel in the United States, mobile emissions were discovered to be a new source with smog in Los Angeles. The engine of an automobile produces more emissions than other combustion processes using fossil fuel for the following reasons:

1. The engines are often oxygen deficient.

2. The engines preheat their air-fuel mixture.

3. The engines have unsteady combustion, with each flame lasting about 0.005 s.

4. The engines have flames that directly contact cooled surfaces.

Carbon monoxide (CO) is a colorless, odorless, and poisonous gas. CO enters the bloodstream through the lungs and reduces oxygen delivery to the body's organs and tissues. Heavy concentrations of CO are generally observed under highly congested condition. In cities, the major source of the CO may be automobile exhaust. CO is present in the combustion products from any carbon-bearing fuel, gasoline, natural gas, coal, wood, charcoal, forest fires, and so on. CO is produced by incomplete combustion. The amount of CO also depends strongly on the normalized Air-fuel ratio (A/F) (Cooper et al., 1996; *de Nevers, 2010*). Hydrocarbons (HC) are highly poisonous gases and cause air-pollution related deaths. Hydrocarbons reacts with oxides of nitrogen and sunlight to form ozone (O_3). O_3 irritates the eyes, damages the lungs, and aggravates respiratory problems. O_3 is the most widespread and intractable urban air pollution problem. HC is a set of chemical compositions including methane, ethane, acetylene, propylene, formaldehyde, aldehydes, benzene, toluene, and xylenes. The methane, ethane, acetylene, propylene, formaldehyde, and aldehydes must have been formed by incomplete combustion, but the benzene, toluene, and xylenes were present in the fuel. They are the slowest-burning gasoline components, so they have the highest probability of passing into exhaust. Therefore, incomplete combustion is major cause to increase amount of HC. HC composition is measured by chromatography, and the weights of the various components are totaled. Oxides of nitrogen produced by combustion engines include NO and NO_2, and they are light brown gases than can become a critical component of urban haze. NO_2 is an important factor in generating ozone. NO can irritate the lungs and lower resistance to respiratory infections such as influenza.

Nitrogen oxides are produced as the result of high-temperature combustion processes. CO_2 is a colorless, tasteless gas that provides carbonation in soft drinks and sparkling wines. The CO_2 concentration in the Earth's atmosphere is approximately 360 ppm and has been increasing about 1.5 ppm per year from the past 30 years. CO_2 does not directly impair human health, but it is a greenhouse gas that traps the earth's heat and contributes to the potential for global warming. Complete combustion of fuel produces more CO_2.

Emissions and Fuel Consumption in Transportation Systems

In the past decades, emissions have been given attention because of the increasing demand for vehicles and facilities (*de Nevers, 2010*). At present, however, most vehicle emissions are captured by mechanical methods such as pre-catalytic converters for CO and HC; Exhaust Gas Recirculation (EGR) for NO_x; and three-way catalysts for NO, CO, and HC. Today, greenhouse gas and fuel consumption receive more attention than other emissions in transportation systems. Transportation is a significant source of carbon dioxide (CO_2) emissions (Barth et al., 2008). Improvement of traffic operations can reduce CO_2 emissions, particularly by mitigating congestion. Barth et al. found that CO_2 emissions could be reduced up to 20% through three different strategies: congestion mitigation strategies that increase average speed of traffic flow, speed management techniques that decrease high free-flow speeds to more moderate speeds, and shock wave suppression techniques that estimate the acceleration and deceleration events associated with stop-and-go traffic. The reduction of greenhouse gas is also related to reduction of fuel consumption, and traffic congestion is primary factor of increasing fuel consumption (Greenwood et al., 2007).

Changing driving behavior to reduce emissions and fuel consumption has recently been considered by some studies. One driving behavior under consideration is aggressiveness of acceleration maneuvers because aggressive acceleration increases mobile-source emissions and fuel consumption (El-Shawarby et al., 2005).

Eco-Driving

Fuel consumption of vehicles can be reduced through changes in driving practices. Fuel-efficient driving practices,with conventional combustion vehicles include smoother deceleration and acceleration, keeping engine revolutions low, shutting off the engine when idling, reducing maximum speeds, and maintaining proper tire pressure (*IEA, 2001; Metz et al., 2007*). Results from studies conducted in Europe and the United States suggeste a possible improvement of 5–20% in fuel economy through eco-driving training. The mitigation costs of CO_2 through eco-driving training were mostly estimated to be negative (*ECMT/IEA, 2005; Metz et al., 2007*). Eco-driving training can be attained through formal training programs or on-board technology aids. It applies to drivers of all types of vehicles, from mini-cars to heavy-duty trucks. The major challenges are motivating drivers to participate in eco-driving programs and ensuring drivers maintain an efficient driving style long after participating (*IEA, 2001; Metz et al., 2007*). In the Netherlands, eco-driving training is provided as part of driving school curricula (*ECMT/IEA, 2005; Metz et al., 2007*).

Eco-driving can be simply defined as economical and defensive driving techniques that depend on the habits of the driver, external factors, and vehicle maintenance (GTZ, 2005; Sivak et al., 2011). One form of economic driving is a driving style to reduce fuel cost. Traffic conditions, obviously, have a major influence on fuel consumption. In heavy traffic conditions, drivers must accelerate, brake, and change gear frequently, increasing the degree of acceleration resistance and thus increasing fuel consumption. In eco-driving, aggressive driving behavior could include rapid acceleration, driving extremely closing to a preceding vehicle, and heavy braking. Aggressive driving consumed 45% more fuel than normal driving; on the other hand, defensive driving consumed about 22% less fuel than normal driving. Road conditions are very important external factors of eco-driving because they are directly related to rolling and gradient resistances. Weather conditions affect air resistance. Vehicle conditions, such as tire tread, tire pressure, and engine efficiency, could be remarkably improved through appropriate maintenance.

Today, information related to eco-driving can be provided through various devices such as on-board navigation systems and smart-phone applications (Magana et al., 2011). Eco-driving assist systems indicate driving patterns and driving environment and help the driver to drive in a more eco-friendly way. Such driver assist systems microscopically analyze driving behavior to estimate driving behavior changes and to guide eco-driving (Wada et al., 2011; Kamal et al., 2010;Mensing et al., 2011; Ando et al., 2011;Sboohi et al., 2009).

Chapter 3. Trajectory Data

This chapter introduces the Next Generation Simulation (NGSIM) program and research projects using NGSIM data, including loop detector data and trajectory data (particularly information on a study area of I-80), and the structure of NGSIM trajectory data.

NGSIM Data

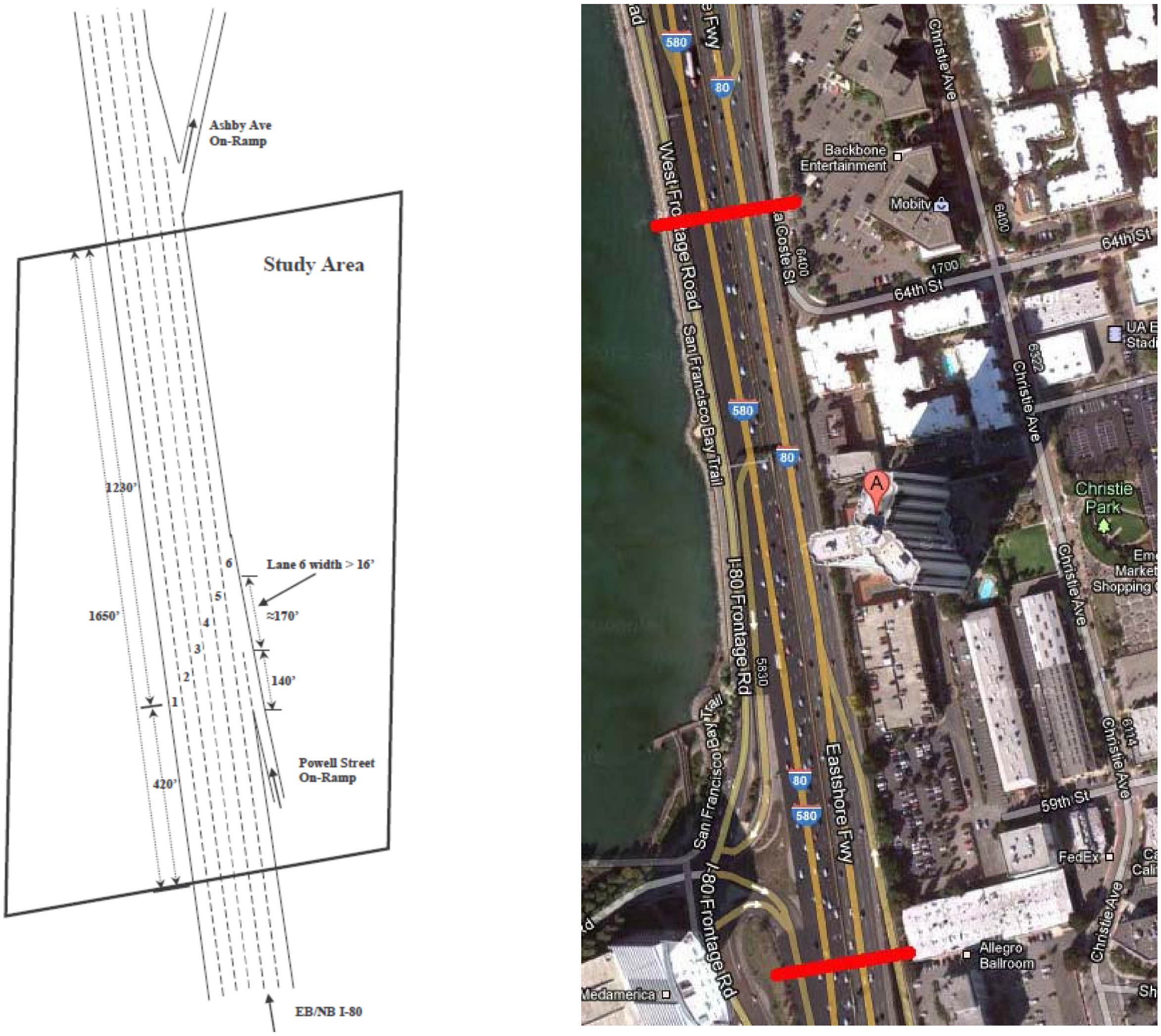
This research mainly used trajectory data obtained from NGSIM program. This program, developed by the Federal Highway Administration (FHWA), aimed to meet the needs of the model users to improve the capability of commercial traffic simulation models (*FHWA, 2010*). The NGSIM program produced core algorithms, validation data sets, and several documentations of the program. The validation data sets are sets of real-world traffic data, including trajectory data as well as loop detector data on their study areas. These data sets were used to validate the core algorithms, including lane changing logic, gap acceptance logic, as well as response to traffic control devices. These data are also expected to be widely used by the traffic simulation community as a resource to assist in the calibration and validation of existing models. A part of the NGSIM program completed by the FHWA provides a data set of vehicle trajectories on the four segments, including two freeways and two arterials. One of the freeway segments is a segment of the Interstate 80 (I-80) in Emeryville, California, and another is a segment of the Hollywood Freeway (US 101) in Universal City, California. The arterial segments are Lankershim Boulevard in Universal City, California, and Peachtree Street in Atlanta, Georgia.

The NGSIM program collected traffic data using several video cameras installed on the buildings near the study areas and converted the data to trajectory data using an image processing algorithm. The trajectory data consists of 18 variables, including vehicle identification number, type of vehicle, profiles of positions, velocities, and accelerations of every single vehicle as well as relative information of car-following collected every 0.1 s.

While there are some disadvantages of the trajectory data, such as a small study area, short observed period, and shortage of information about drivers, the trajectory data holds great possibility. The trajectory data of NGSIM program has large sample size of about 2,000 vehicles for 15 minutes in each segment and high-quality data with a resolution equal to 0.10 s. The data also includes vehicle identification numbers of preceding and following vehicles, which will facilitate microscopic studies of car-following behaviors by matching up trajectory data of three vehicles: subject, preceding, and following vehicles.

Research Area Segment of I-80

The NGSIM trajectory data used in this study were collected on the segment of I-80 in Emeryville (San Francisco), California. This segment of I-80, known as the Berkeley Highway Laboratory (BHL) site, consists of one on-ramp and six main stream lanes (the left-most lane is the high-occupancy vehicle (HOV) lane, and the right-most lane includes a merging area with the on-ramp). Seven video cameras mounted on a 30-story building, Pacific Park Plaza, located at 6363 Christie Avenue and adjacent to I-80, were used to collect data on 1,650 ft of the main lanes and 140 ft of the on-ramp (Cambridge Systematics, 2005a; 2005b; 2005c). The length of the on-ramp that was observed is about 310 ft. The merging area is about 170 ft and starts at 420 ft from the end of the south-bound lanes. All observed vehicles were headed north, and the on-ramp vehicles were coming from Powell Street. See Figure 2.



(a) (b)

Figure 2. (a) Schematic and (b) satellite picture of study area.

Sources: (a):Cambridge Systematics, NGSIM I-80 Data Analysis (4:00 p.m. to 4:15 p.m.) Summary Report, 2005, and (b): Google Map (Address = 6363 Christie Avenue, Emeryville, CA, United States)

Overview of NGSIM Trajectory Data

The NGSIM trajectory data were collected three times on April 13, 2005. The first trajectory data were collected between 4:00 and 4:15 p.m., representing a transitional traffic period that this research calls the non-congestion condition (Cambridge Systematics, 2005a; 2005b; 2005c). The second trajectory data were collected between 5:00 to 5:15 p.m., representing a congested condition that this research calls congestion condition 1. The last trajectory data were collected between 5:15 to 5:30 p.m., representing a congested condition that this research calls congestion condition 2. The trajectory data were transcribed at a resolution of 10 frames per second from the video data collected by the seven cameras.

Tables 3 and 4 show 18 sample items of the trajectory data. In Table 3, the first four columns show the vehicle ID, the frame ID, the total frames, and the global time. Vehicle ID gives the identification number of each vehicle recorded by the ascending time of its entry into the monitored highway section. Frame ID gives the specific time when an individual vehicle enters into the section, measured in tenths of a second from the start time of recording. Total frames represents the total travel time that each individual vehicle appears in the section, measured in tenths of a second. Global time represents the elapsed time since January 1, 1970, in milliseconds. The location data given in columns 5, 6, 7, and 8 were expressed using two types of coordinate systems: local and global. The local coordinate system uses the intersection of the edge of the left-most lane and the entry into the highway section as the origin point. The global coordinate system is based on the CA State Plane III in NAD83. The values in columns 5 and 6 provide the mean lateral and longitudinal coordinates of the front center of the vehicle in feet, using the local coordinate system. The values in columns 7 and 8 show the mean lateral and longitudinal coordinates of the front center of the vehicle in feet, using the global system. The trajectory data includes information about the characteristics of the vehicles: vehicle length (feet) in column 9, the vehicle width (feet) in column 10, and vehicle class in column 11. Vehicle class represents the type of vehicle: motorcycle (1), automobile (2), and truck (3). Vehicle velocity, in column 12, represents the instantaneous velocity of the vehicle in feet per second, and vehicle acceleration, in column 13, represents the instantaneous acceleration of the vehicle in feet per squared second. In Table 4, the lane identification, in column 14, represents the current lane position of the vehicle. Lane 1 is farthest left lane; lane 6 is farthest right lane. Lane 7 is the on-ramp at Powell Street, and lane 9 is the shoulder on the right side. The preceding vehicle, in column 15, and the following vehicle, in column 16, provide the vehicle identification number of the lead vehicle and the vehicle following the subject vehicle, respectively. A value of zero indicates that there was no no preceding or following vehicle. The trajectory data also includes the space headway (feet), shown in column 17, and the time headway (seconds), shown in column 18.

Table 2. Sample trajectory data obtained from NGSIM data (from columns 1 to 9).

Vehicle ID	Frame ID	Total Frames	Global Time	Local X	Local Y	Global X	Global Y	Vehicle Length
2774	8299	803	1113433964800	16.978	1627.675	6042621	2134682	14.3
2774	8300	803	1113433964900	16.985	1630.855	6042620	2134685	14.3
2774	8301	803	1113433965000	17.013	1633.92	6042620	2134688	14.3
2774	8302	803	1113433965100	17.033	1636.919	6042619	2134691	14.3
2774	8303	803	1113433965200	17.057	1640.42	6042619	2134694	14.3
2775	7500	778	1113433884900	15.624	66.027	6042839	2133135	14.8
2775	7501	778	1113433885000	15.626	69.065	6042838	2133138	14.8
2775	7502	778	1113433885100	15.624	72.064	6042838	2133141	14.8
2775	7503	778	1113433885200	15.624	74.565	6042838	2133144	14.8
2775	7504	778	1113433885300	15.623	78.064	6042837	2133147	14.8

Table 3. Sample trajectory data obtained from NGSIM data (from columns 10 to 18).

Vehicle Width	Vehicle Class	Vehicle Velocity	Vehicle Acceleration	Lane Identification	Preceding Vehicle	Following Vehicle	Spacing	Headway
7.4	2	32.32	-10.39	2	0	2796	0	0
7.4	2	29.69	11.2	2	0	2796	0	0
7.4	2	29.69	0	2	0	2796	0	0
7.4	2	29.69	0	2	0	2785	0	0
7.4	2	29.69	0	2	0	2796	0	0
7.4	2	20.34	0	2	2486	0	85.16	4.19
7.4	2	20.34	0	2	2486	0	85.13	4.19
7.4	2	20.34	0	2	2486	0	85.02	4.18
7.4	2	20.34	0	2	2486	0	85.38	4.2
7.4	2	20.34	0	2	2486	0	84.73	4.17

Data Analysis

The NGSIM program collected traffic data, including trajectory data, in a segment of I-80 for three 15-minute time periods: 4:00 to 4:15 p.m., 5:00 to 5:15 p.m., and 5:15 to 5:30 p.m. (Cambridge Systematics, 2005a; 2005b; 2005c). Cambridge Systematics conducted a data analysis of the trajectory data and provided aggregating summaries of vehicle flow and speed, number of lane changes, headway and gap analysis, and input-output analysis of flow. They published data analyses for each time period, including results of the analysis aggregated by time, distance (100 feet), and lane.

This chapter summarizes their reports to define the base traffic condition scenario for three time periods: 4:00 to 4:15 p.m., 5:00 to 5:15 p.m., and 5:15 to 5:30 p.m.. It also compares the flow and speed of three time periods.

Description of Vehicle Types

The NGSIM data classifies vehicles into three categories: (1) motorcycle, (2) automobile, and (3) truck and buses. Figure 3 shows the distributions of vehicle types for the three time periods.

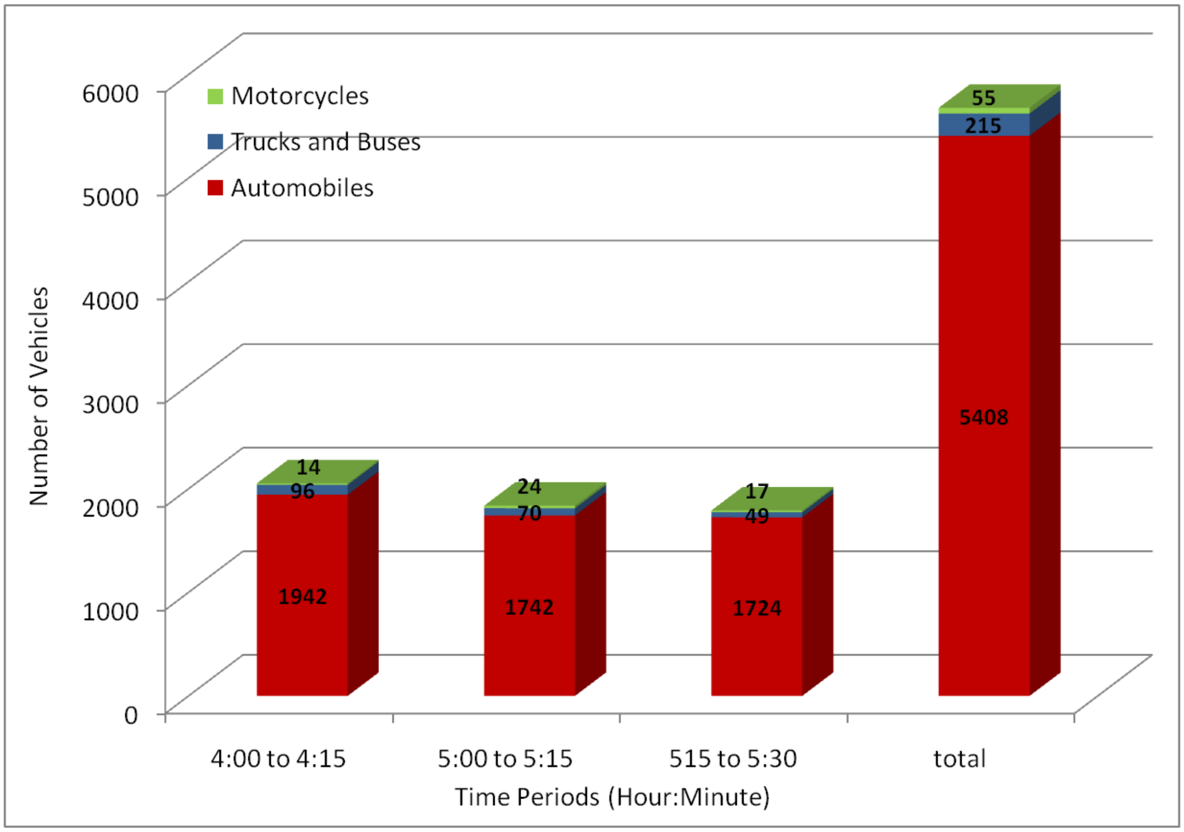


Figure 3. Distribution of vehicle types during each time period and across all time periods.

Traffic Flow

Flow, in terms of vehicles per hour, is calculated by multiplying the number of vehicles observed to pass through the entire study area during a 15-minute period by 4 (Cambridge Systematics, 2005a; 2005b; 2005c). The traffic flow was observed at the midpoint of each study section during the 15-minute time period, and then was converted into the hourly traffic flow in vehicles per hour. Figure 4 shows the average traffic flow for every 5-minute period. The early parts of the entire time period, including the first time period and the first 5 minutes of the second time period, have high traffic flow. At the second 5 minutes of the second time period, the traffic flow significantly decreased. The average traffic flow sharply increased during the first 5 minutes of the third time period. This was because only vehicles that entered and completely passed through the study area within each time period were counted for traffic flow.

some vehicles of traffic flow were ignored because they did not pass through the study area within the time period. In the first time period, most vehicles passed through the entire study area, and there was little reduction of traffic flow. However, in the second time period, many vehicles did not pass through the entire area, so flow was greatly reduced. Traffic congestion, therefore, occurred in the second and the third time periods, but not in the first time period. In the first 5 minutes of the third time period, the sharp increase of traffic flow is observed. This is because many vehicles that did not pass through the study area for the second time period were ignored in the NGSIM trajectory data. In the second time period, a lot of vehicles did not pass through the entire area because of the congestion. Since the travel time for study area is less than 15 minutes under congested condition, most of vehicles entering the study area at the first 5 minutes of the second time period passed through the entire area. However, some vehicles entering the study area at the second 5 minutes of the second time period could not pass through the area within the time period, and were ignored in the NGSIM data. In the last 5 minutes of the second time period, most of vehicles entering the area could not pass the area within the time period, were ignored. In the first 5 minutes of the third time period, most of vehicles did pass through the study area, and the NGSIM involves their trajectory data. Therefore, the traffic flow in the third 5 minutes of the second time period would be similar to the traffic flow in the first 5 minutes of the third time period.

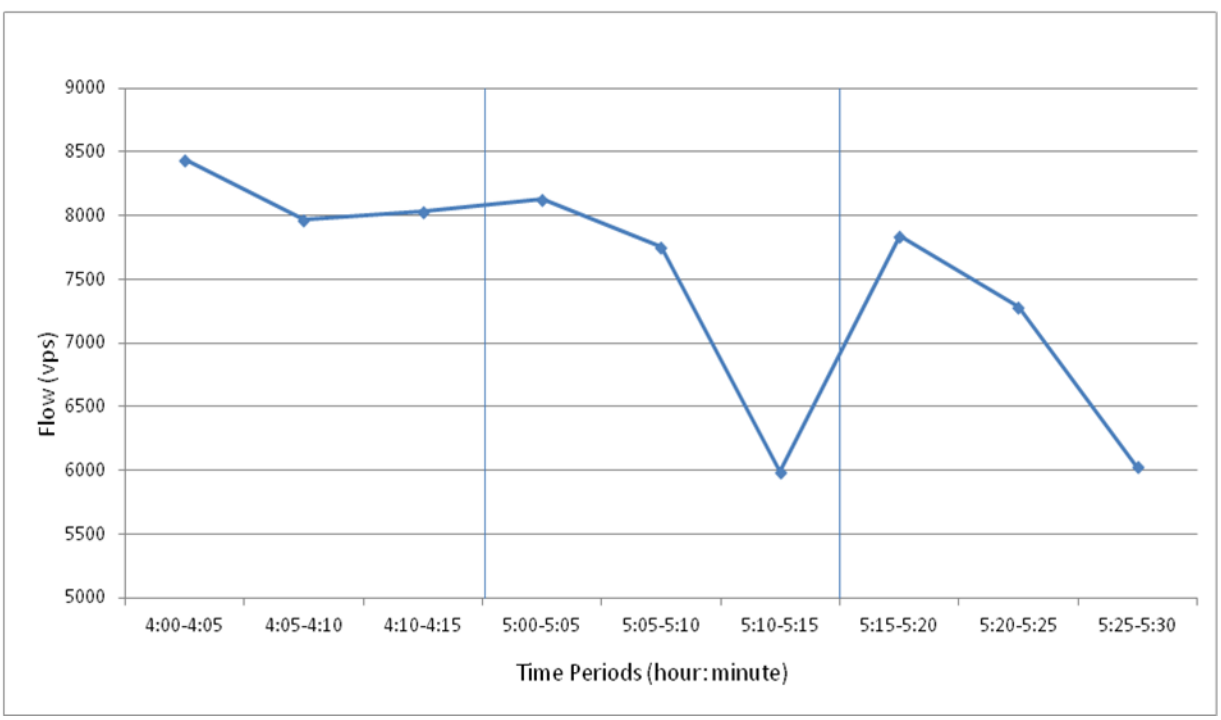


Figure 4. Hourly traffic flow in entire section and all lanes by time period.

Data source: Cambridge Systematics, Summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4: 15p.m.), (5:00 to 5:15), and (5:15 to 5:30).

Figure 5 compares average traffic flow by lane for the three time periods. Traffic flow in lane 3 was the lowest for all time periods, and the traffic flow in the lane 1, the HOV lane, increased during congestion. Traffic flow on lane 6 for all time periods was high because many vehicles took the lane to exit or enter the highway, and it released congestion due to the exit ramp on the downstream.

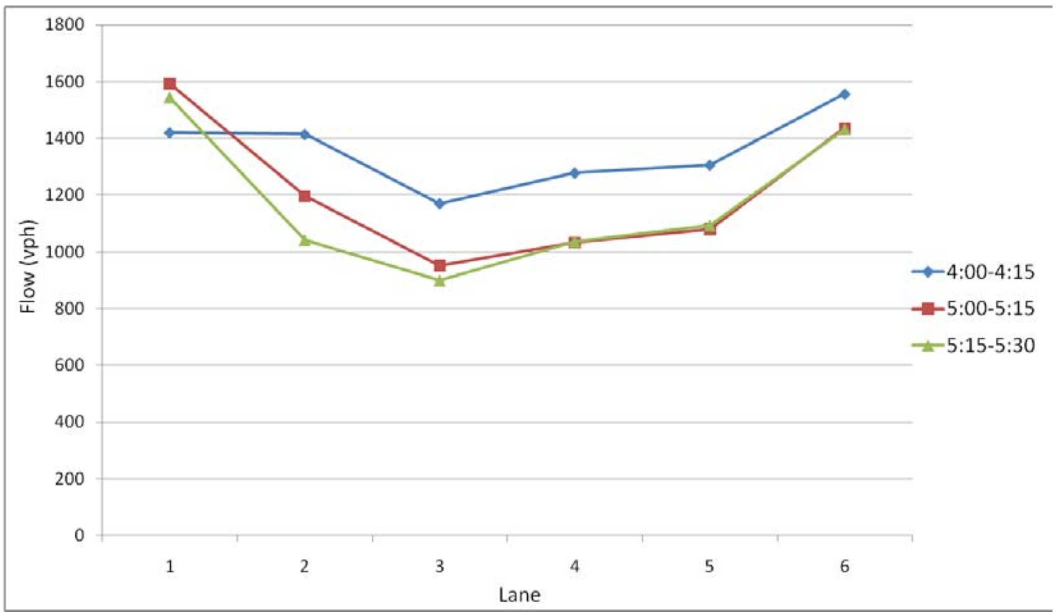


Figure 5. Flow by lane for three time periods.

Data source: Cambridge Systematics, Summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4: 15p.m.), (5:00 to 5:15), and (5:15 to 5:30).

Speed

The speed analysis used two types of average speed: time mean speed (TMS, in mph) and space mean speed (SMS, in mph) (Cambridge Systematics, 2005a; 2005b; 2005c). The time mean speed is calculated at the midpoint of each study section using Equation 1:

$$\mathrm{TMS}(t,s) = \frac{\sum_i v(t,s)_i}{n(t,s)}$$

where TMS(t,s) is the time mean speed in section s during time period t measured at midsection, v(t,s) is the instantaneous speed of vehicle i in section s during time period t measured at midsection, and n(t,s) is the number of vehicles traversing section s during time period t.

The space mean speed is calculated by dividing the sum of trajectory lengths traversed by all vehicles in a section by the sum of time taken to transverse these sections (Equation 2):

$$\mathrm{SMS}(t, s) = \frac{\sum_i d(t, s)_i}{\sum_i tt(t, s)_i}$$

where SMS(t,s) is the space mean speed in section s during time period t, $d(t, s)_i$ is the distance traveled by vehicle i in section s during time period t, and tt is the travel time of vehicle i in section s during time period t.

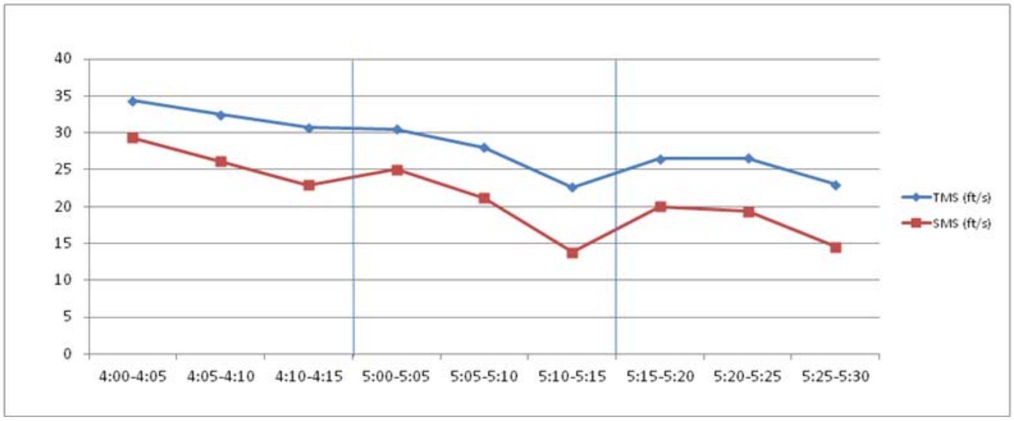


Figure 6. Time mean speed and space mean speed in all sections and all lanes by time period.

Data source: Cambridge Systematics, Summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4: 15p.m.), (5:00 to 5:15), and (5:15 to 5:30).

During the first time period, 2,052 vehicles were observed, consisting of 14 motorcycles, 1,942 automobiles, and 96 trucks and buses. The average flow was 8,144 vehicles per hour. The average TMS was 22.19 mph (32.55 ft/s), and the average SMS was 17.86 mph (26.20 ft/s). Of the 2,052 vehicles, 191 entered the freeway via the on-ramp. This dissertation research used the trajectory data of this time period based on the assumption that the first time period is not under the congestion condition.

In the second time period, 1,836 vehicles were observed, consisting of 24 motorcycles, 1,742 automobiles, and 70 trucks and buses, and the average flow was 7,288 vehicles per hour. The average TMS was 18.72 mph (27.46 ft/s), and the average SMS was 14.04 mph (20.59 ft/s). Of the 1,836 observed vehicles, 205 entered the freeway via the on-ramp.

In the third time period, 1,790 vehicles were observed, consisting of 17 motorcycles, 1,724 automobiles, and 49 trucks and buses, and the average flow was 7,048 vehicles per hour. The average TMS was 17.40 mph (25.52 ft/s), and the average SMS was 12.40 mph (18.19 ft/s). Of the 1,790 observed vehicles, 211 entered the freeway via the on-ramp.

Input-Output Analysis

In all the recorded data, the vehicles that did not completely pass through the study area were ignored (Cambridge Systematics, 2005a; 2005b; 2005c). Table 4 shows the results of an input-output analysis by lane and sub-time period (5 minutes) for the three time periods. In the table 4, the NGSIM trajectory data includes extra vehicles that entered the study area prior to each time periods, or passed through the study area after each time period.

Lane change analysis and sectional analysis

Figure 7 shows that most vehicles did not change lanes, or changed lanes only once on the study area during the entire time period. Figure 8 shows that the traffic entering via the on-ramp merged in section 7.

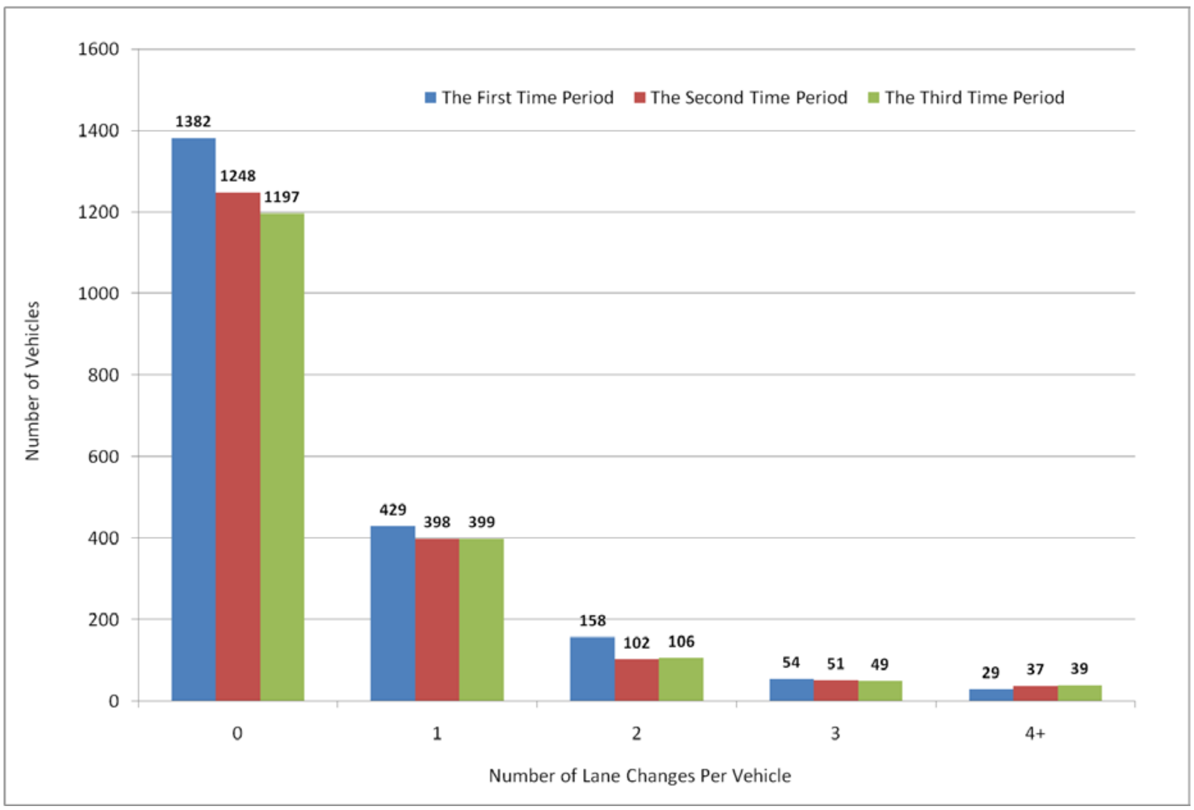


Figure 7. Distribution of vehicles by number of lane changes per vehicle for the three time periods

Table 4. Input-output analysis by lane and sub-time period.

		First time period						Second time period						Third time period					
		3:58:55 -4:00	4:00 - 4:05	4:05 - 4:10	4:10 - 4:15	4:15- 4:15:37	Sum	4:59:27 -5:00	5:00 - 5:05	5:05 - 5:10	5:15 - 5:15	5:15- 5:15:47	Sum	5:12:45- 5:15	5:15 - 5:20	5:20 - 5:25	5:25 - 5:30	5:30- 5:32:14	S
Number of Vehicles Entering	lane 1	8	110	113	125	0	356	6	121	139	122	0	388	9	127	128	126	0	39
	lane 2	9	116	114	102	0	341	9	112	104	69	0	294	15	84	91	57	0	24
	lane 3	7	100	93	82	0	282	9	96	81	48	0	234	17	78	89	38	0	22
	lane 4	8	122	95	93	0	318	8	99	93	51	0	251	16	94	89	52	0	25
	lane 5	10	96	109	86	0	301	8	104	88	51	0	251	15	96	83	60	0	25
	lane 6	11	89	98	65	0	263	7	85	72	49	0	213	13	76	78	48	0	21
	On-Ramp	1	62	64	64	0	191	0	81	71	53	0	205	11	81	71	48	0	21
	Sum	54	695	686	617	0	2052	47	698	648	443	0	1836	96	636	629	429	0	17
Number of Vehicles Entering	lane 1	0	113	116	123	7	359	0	122	142	132	9	405	0	123	136	131	8	39
	lane 2	0	125	120	130	13	388	0	108	118	78	16	320	0	99	94	81	13	28
	lane 3	0	101	95	101	13	310	0	91	92	65	11	259	0	86	83	57	17	24
	lane 4	0	105	107	107	11	330	0	89	101	63	15	268	0	94	85	64	15	25
	lane 5	0	104	98	95	12	309	0	98	79	62	9	248	0	94	77	71	13	25
	lane 6	0	109	117	118	12	356	0	106	112	108	10	336	0	117	109	114	9	34
	Sum	0	657	653	674	68	2052	0	614	644	508	70	1836	0	613	584	518	75	17

Data source: Cambridge Systematics, Summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4: 15p.m.), (5:00 to 5:15), and (5:15 to 5:30).

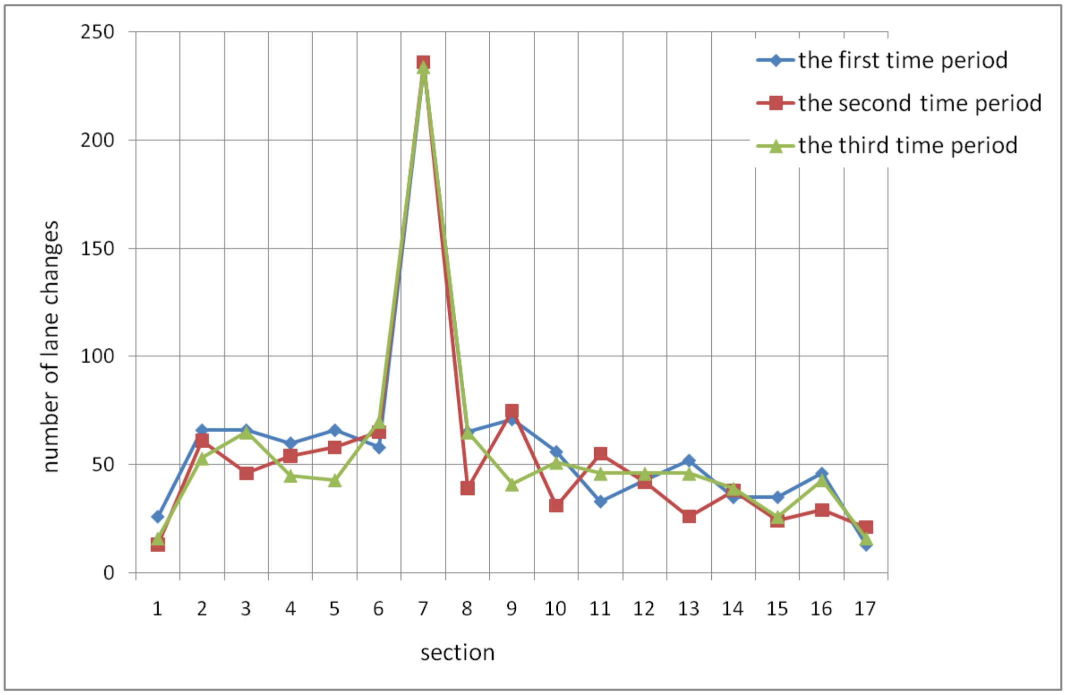


Figure 8. Number of lane changes by section.

Note: On-ramp merge traffic starts in section 7.

Data source: Cambridge Systematics, Summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4: 15p.m.), (5:00 to 5:15), and (5:15 to 5:30).

The data collection effort tracked vehicles over a length of 1,650 feet, and the sectional data analysis was conducted. In the sectional data analysis, the study area was separated into 17 sections, every 100 feet of sectional length. Figure 8 shows number of lane changes counted by section, and the vehicles entering through the on-ramp were counted in section 7. Figure 9 through Figure 11 show the average traffic flow and average speed by sections over the three time periods.

Time Headway Analysis

Figure 12 shows average time headway by lane in seconds. Because the time headway is calculated by distance headway divided by speed, a low time headway does not necessarily mean there is a short distance between a subject vehicle and its preceding vehicle (Cambridge Systematics, 2005a; 2005b; 2005c). The lowest average time headway in lane 1 was caused by the high average speed of vehicles in that lane.

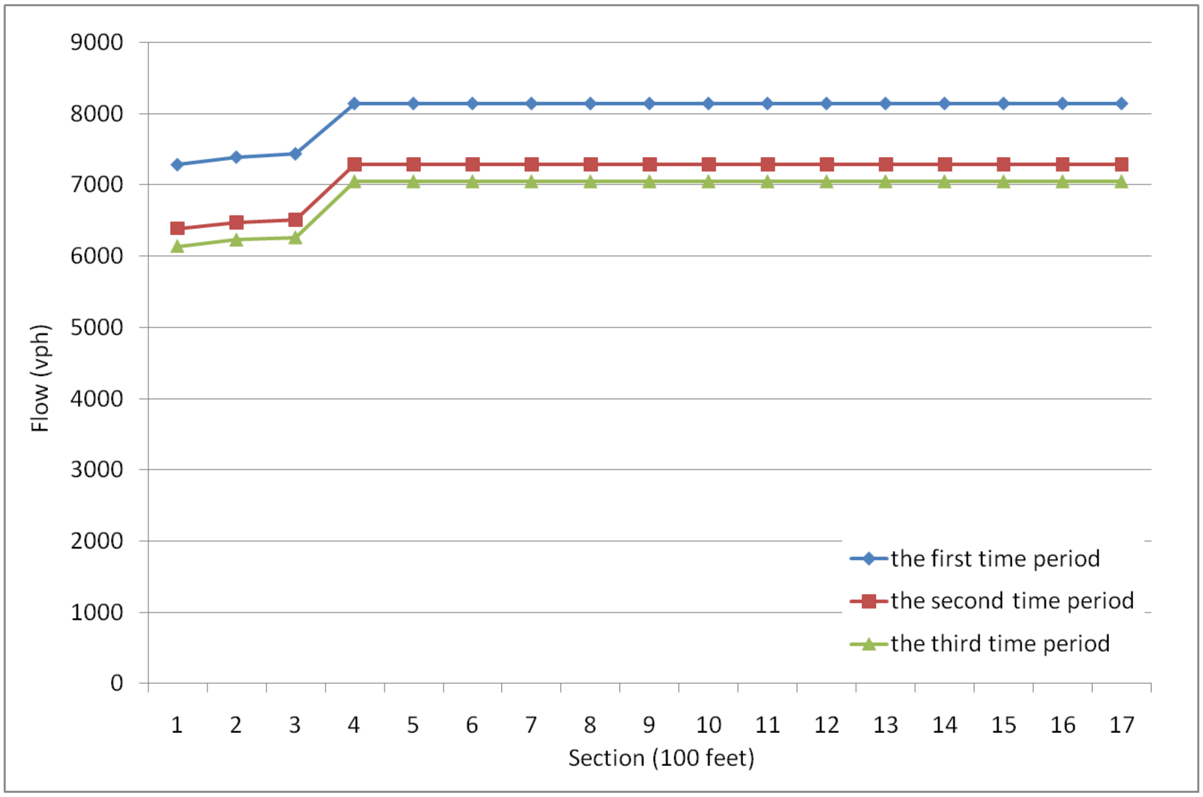


Figure 9. Average traffic flow by section for three time period.

Data Source: Cambridge Systematics, summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4:15 p.m.).

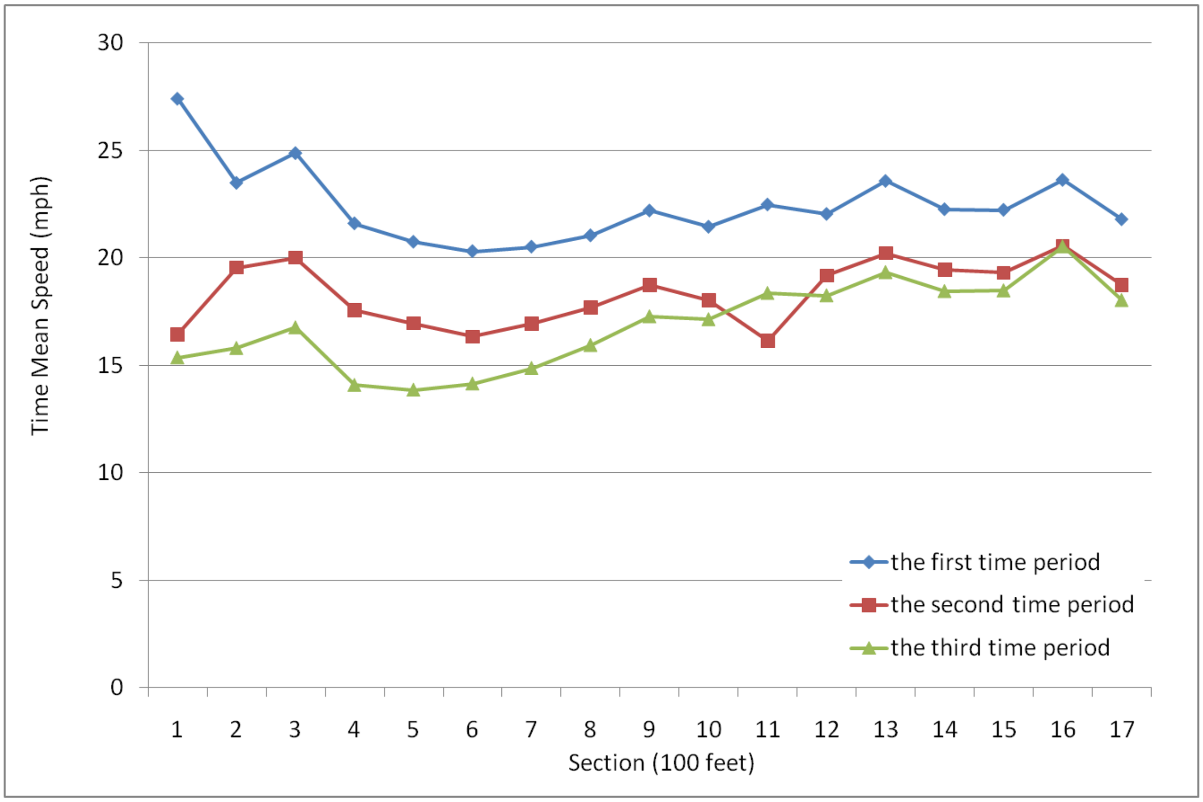


Figure 10. Time Mean Speed (TMS) by section for three time period.

Data Source: Cambridge Systematics, summary report of NGSIM I-80 Data Analysis (5:00 p.m. to 5:15 p.m.).

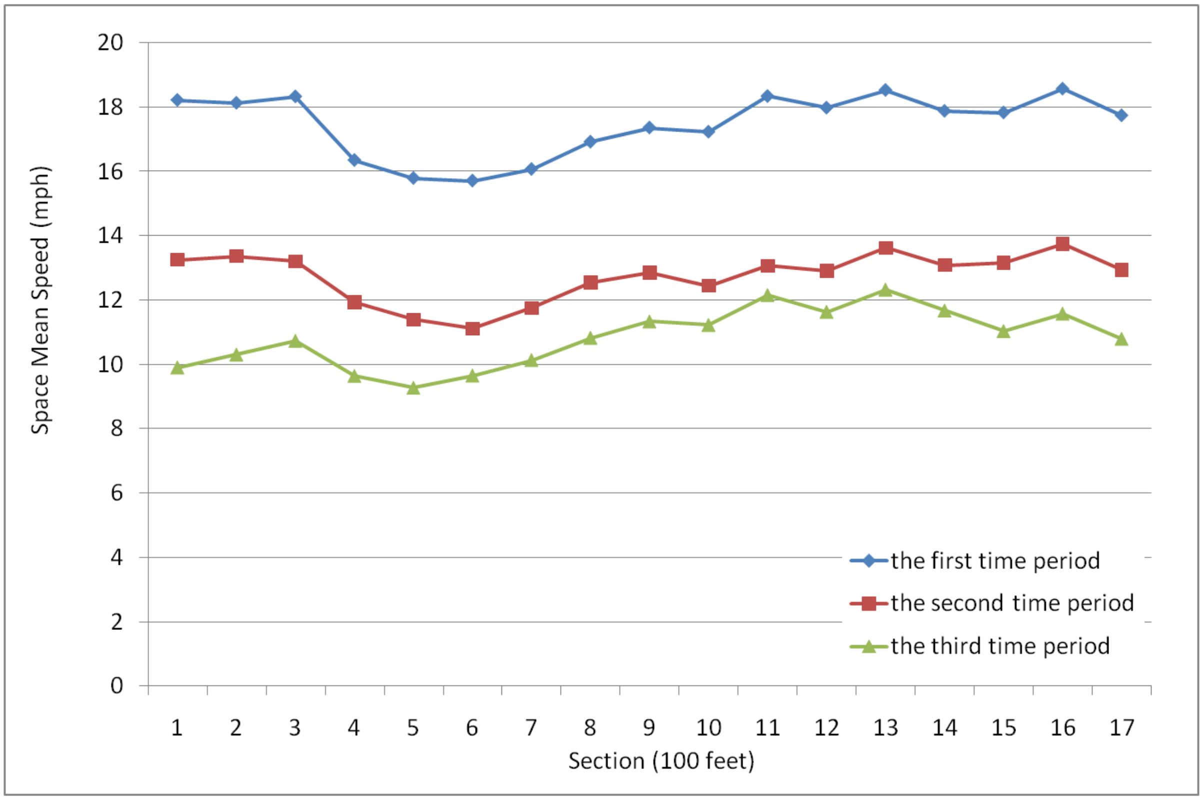


Figure 11. Space Mean Speed (SMS) by section for three time period.

Data Source: Cambridge Systematics, summary report of NGSIM I-80 Data Analysis (5:15 p.m. to 5:30 p.m.).

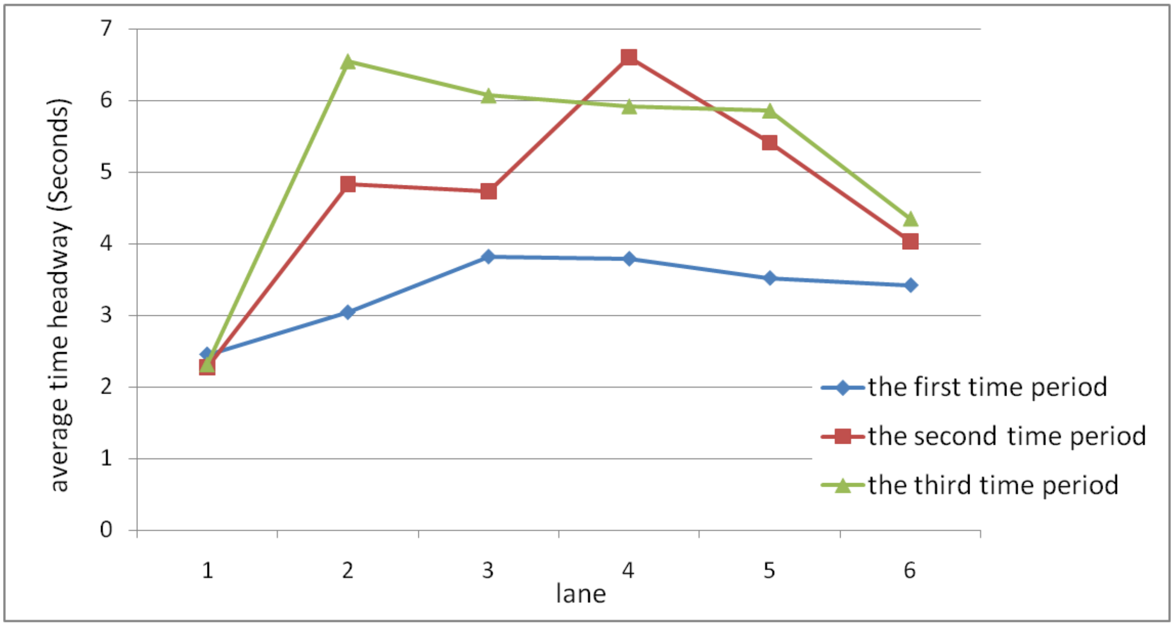


Figure 12. Average headway by lane.

Data source: Cambridge Systematics, summary report of NGSIM I-80 Data Analysis (4:00 p.m. to 4:15 p.m., 5:00 p.m. to 5:15 p.m., and 5:15 p.m. to 5:30 p.m.).

Findings

Based on the summary reports of the NGSIM I-80 data analysis, the base traffic environment in the research area can be summarized as follows:

1. Most of observed vehicles are automobiles.
2. NGSIM trajectory data includes only vehicles that entered and exited the research area within the research time period; there was some loss of vehicles from the entire traffic flow.
3. Most observed vehicles did not change lanes or changed lanes only once.
4. The merging area was the section 7,600 ft to 700 ft from south bound.
5. Most vehicles probably did not keep constant time headway, but rather kept constant distance headway (spacing).

Chapter 4. Methodologies

This chapter introduces two major methodologies used in this research: factor analysis and the k-means clustering algorithm. It also introduces the statistical analyses for study cases 1 and 2.

Factor Analysis

Overview

Factor analysis is a statistical method for investigating whether a number of variables are linearly related to a smaller number of unobservable factors, based on the common factor model (Everitt et al., 2001; Johnson et al., 2007; DeCoster, 1998; Abdi, 2003). The essential purpose of factor analysis is to describe the covariable relationship among many variables in terms of a few underlying, but unobservable, random quantities, called factors. The common factor model proposes that each observed response or measure is influenced partially by underlying common factors and also by unique factors, neither of which can be observed. Generally, factor analysis results in a smaller number of factors than the original total number of variables. By determining factor classifications through the factor analysis with measured data, and by using these factors as variables instead of using the observed responses, we can reduce the number of variables to a set that are more describable and simpler.

Basically, there are two types of factor analysis: exploratory and confirmatory. The exploratory factor analysis (EFA) attempts to discover the nature of the constructs influencing a set of responses. The primary objectives of the EFA are to determine the number of common factors and the strength of the relationship between each factor and each observed measure. The confirmatory factor analysis (CFA) tests whether a specified set of constructs is influencing responses in a predicted way. The primary objective of the CFA is to determine the ability of a predefined factor model to fit an observed set of data. When classifying a group of items, EFA is commonly used to determine what features are most important. This research conducted the EFA for classification. The EFA is performed through seven basic steps (De Coster, 1998):

1. Measure variables on the same experimental units, and then collect the measurements.
2. Calculate the correlation or covariance between each variable, and then construct the correlation matrix.
3. Determine the number of factors. There are a number of methods to determine the optimal number of factors by examining data. For example, the Kaiser criterion is to drop all components with eigenvalue less than 1.0. That is, the Kaiser criterion recommends using a number of factors equal to the number of the eigenvalues of the correlation matrix that are greater than 1.0. In another method, the Cattell Scree Test recommends plotting the eigenvalues for the correlation matrix in descending order and then using a number of factors

equal to the number of eigenvalues that occur prior to the last major drop in eigenvalue magnitude.

4. Extract the initial set of factors. There are a number of different extraction methods, including maximum likelihood, principal component, and principal axis extraction. The best method is generally the maximum likelihood extraction.
5. Rotate factors to find a final solution. There are theoretically an infinite number of ways to define factors for the same amount of covariance. Rotating factors allows a determination of a factor solution that is equal to that obtained in the initial extraction but that has the simplest interpretation. There are many different types of rotation (Abdi, 2003). We can categorize them into two major categories: orthogonal and oblique. Orthogonal rotations produce uncorrelated factors, and oblique rotations produce correlated factors. The Varimax method is widely known as the best orthogonal rotation. The oblique rotations are less distinguishable, and the three most commonly used are Direct Quartimin, Promax, and Harris-Kaiser Orthoblique.
6. Linearly relate each measure to each factor. The strength of this relationship is contained in the respective factor loading produced by the rotation. This loading can be interpreted as a standardized regression coefficient, regressing the factor on the measures.
7. Construct factor scores that are a linear combination of all of the measures, weighted by the corresponding factor loading for additional analyses using factors as variables.

The factor analysis can be considered an extension of principal component analysis or principal factor analysis. Both methods attempt to approximate the covariance matrix, but the approximation of the factor analysis is more elaborate than the approximation of the principal component analysis. See Figure 13.

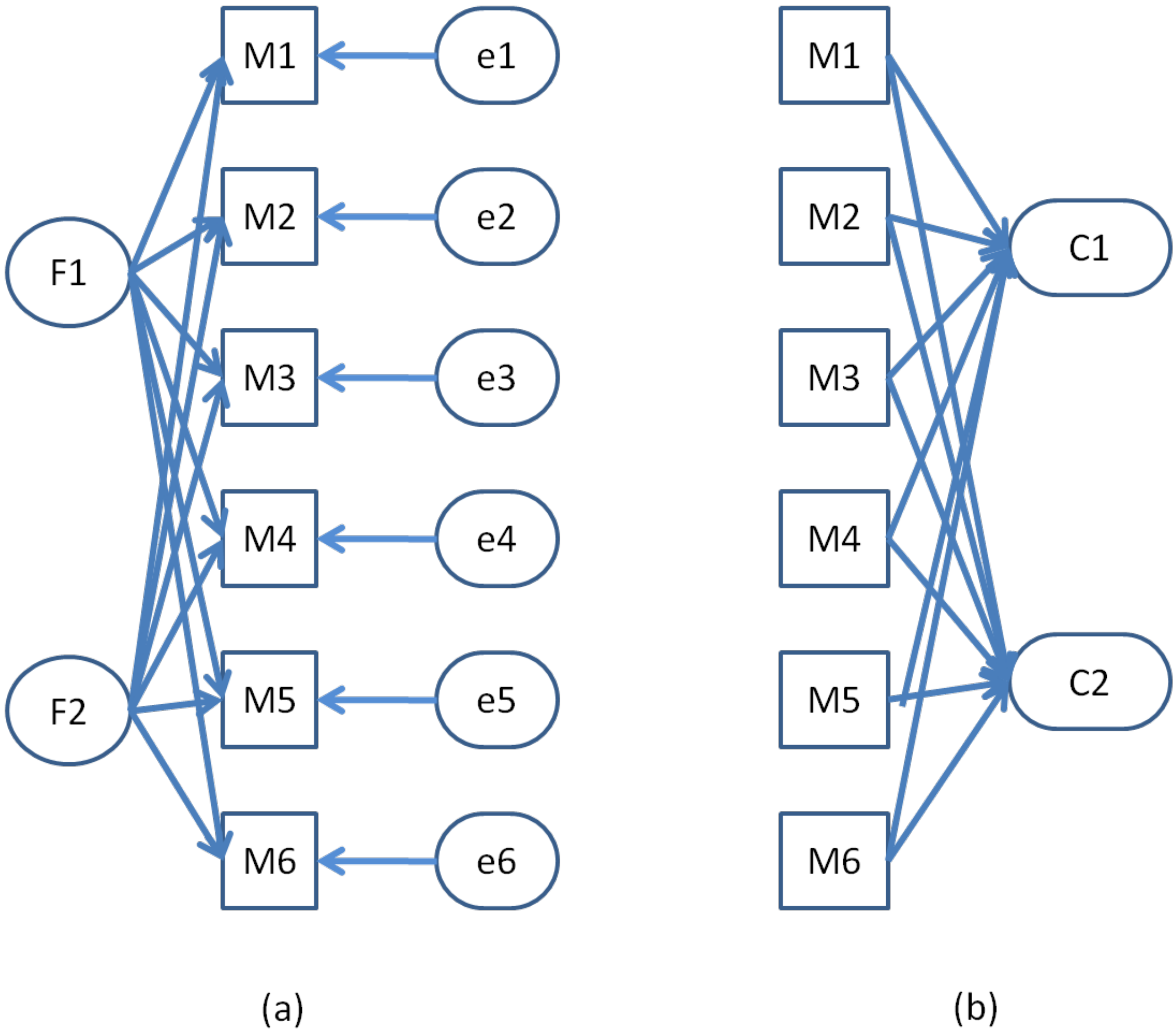


Figure 13. Comparison of models of (a) factor analysis and (b) principal component analysis.

Note: F is factor, M is measure, e is error, and C is component.

Basic Factor Analysis Model

Assume that there are p observed or manifest variables, assumed to be linked to a smaller number of k unobserved latent variables by a regression model of the form in Equation 3:

$$x_1 = \beta_{1,1}f_1 + \beta_{1,2}f_2 + \cdots + \beta_{1,k}f_k + e_1$$

$$x_2 = \beta_{2,1}f_1 + \beta_{2,2}f_2 + \cdots + \beta_{2,k}f_k + e_2$$

$$\vdots$$

$$x_p = \beta_{p,1}f_1 + \beta_{p,2}f_2 + \cdots + \beta_{p,k}f_k + e_p$$

where k<p, x_i is i^{th} observed variables, and f_j is j^{th} unobserved variables.

These equations may be written more concisely as Equation 4:

$$\vec{\mathbf{x}} = \mathbf{\Lambda}\vec{\mathbf{f}} + \vec{\mathbf{u}}$$

where

$$\vec{\mathbf{x}} = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix}, \quad \mathbf{\Lambda} = \begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,k} \\ \vdots & \ddots & \vdots \\ \beta_{p,1} & \cdots & \beta_{p,k} \end{pmatrix}, \quad \vec{\mathbf{f}} = \begin{pmatrix} f_1 \\ \vdots \\ f_k \end{pmatrix}, \quad \vec{\mathbf{u}} = \begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix}.$$

Assume that the residual terms $u_1, \ldots, u_p$ are uncorrelated with each other and with the factors $f_1, \ldots, f_k$. The manifest variables are independent, and the correlations of the observed variables arise from their relationships with the factors as factor loading. Because the factors are unobserved, their location and scale can be arbitrarily fixed. Thus, it is possible to assume that the factors occur in standardized form with mean 0 and standard deviation 1. The factor analysis assumes that the factors are uncorrelated with one another.

In the factor analysis, the Λ represents the factor loading matrix, and the covariance matrix of observed variables, Σ, is defined as Equation 5:

$$\Sigma_x = \Lambda\Lambda^T + \Psi$$

where Ψ is the specific variances with elements of the diagonal matrix.

The factor analysis determines the best loadings and residual terms to account for observed variables, which allows for the determination of the value of factors from the value of measures. There are two main estimation techniques to estimate parameters: principal factor analysis and maximum likelihood factor analysis. This research used the factor analysis model using the maximum likelihood estimation technique for study case 2.

K-means Clustering Algorithm

The k-means algorithm is one of the most popular and well-known clustering algorithms. It can be classified as a partitioning method of clustering methods (Theodoridis et al., 2009). This algorithm is called an isodata or c-means algorithm. This algorithm is a special case of the generalized hard clustering algorithmic scheme when point representatives are used, and the squared Euclidean distance is adopted to measure the dissimilarity between objects. The major advantage of the k-means algorithm is its computational simplicity, which makes it an attractive candidate for a variety of applications because the time complexity of this algorithm is O(Nmq). The k-means algorithm, therefore, is often suitable for a large amount of data.

The k-means clustering algorithm measures dissimilarity between objects and then assigns them into k clusters based on the dissimilarity, called distance. Since the k is constant, determining the correct number of clusters is generally required prior to the clustering. A widely used way to determine the number of clusters is to use silhouette values, which represent how close each point in one cluster is to points in the neighboring clusters. This measure ranges from +1 to -1. A silhouette value of +1 indicates that points are not distinctly in one cluster or another, and a silhouette value of -1 indicates that points are probably assigned to the wrong cluster. The correct number of clusters is determined by comparing average silhouette values. To find the correct number of clusters, the average silhouette values were repeatedly calculated after the subjects were clustered into incremental number of clusters. The correct number of clusters is the specific number of clusters with the highest average silhouette value.

The k-means algorithm repeatedly conducts expectation and maximization steps until the solutions converge. In the expectation step, the algorithm assigns all objects to k clusters whose centroids are closest to each object. In the maximization step, the algorithm finds point t, which minimizes the sum of distances from all objects in that cluster, and then updates the found point as a centroid of each cluster.

The k-means algorithm proceeds as the following:

Step 1. Initialization: Select k points from set of objects, and then set them as the centroid of k clusters.

Step 2. Expectation: Calculate the distance of each centroid to each object, and then assign each object to the cluster that has a centroid closest to that object using labeling.

Step 3. Maximization: Update all centroids.

Step 4. Distortion: Compute distortion as the sum of the entire distance from each centroid to each object that each cluster owns.

Step 5. Check the end condition: Repeat steps 2 through 5 until condition is complete.

Chapter 5. Study Case 1: Contribution of Aggressive Drivers to Automobile Tailpipe Emissions under Acceleration and Braking Conditions

This chapter describes the first research case: driving behavior based on controlling pedals. Five processes are proposed, consisting of one or more data points of instantaneous acceleration in the NGSIM trajectory data. Each process represents a single driver's behavior of controlling pedals, either brake or accelerator. The environmental characteristics were also studied due to difference between accelerations of individual drivers.

Proposed Target Operating Acceleration

This research microscopically analyzed the car-following behaviors on the basis of one or more instantaneous acceleration data points. A single movement by a driver, such as the movement of a driver's foot, may take several deci-seconds. This research considered the target operating acceleration instead of the instantaneous acceleration to more clearly represent characteristics of individual drivers' acceleration. This is based on the assumption that drivers attempt one movement for one purpose, such as acceleration or deceleration. That purpose would be represented by the magnitude of the acceleration in the movement. This required the division of the trajectory data into several drivers' movements.

Five Processes

Drivers' movements were distinguished by using a "process" concept, representing a single movement as unit measurement of a single driving behavior. The lowest level classifies driver movements into five categories: pushing accelerator pedal, pushing brake pedal, releasing accelerator pedal, releasing brake pedal, and no pedal movement. These five categories of driver movement correspond to five defined processes: accelerating process, braking process, recovery A process, recovery B process, and constant speed process. The accelerating and braking processes are defined as a set of continuously instantaneous data from the moment the driver initiates pushing the accelerator or brake pedal to the initial release of the pedal, respectively. The recovery A and B processes are defined as a set of continuous data from the moment the driver initiates releasing the accelerator or brake pedal, respectively, to the moment the pedal is completely released. The constant speed process is a set of continuous data that indicates that the driver is not pushing any pedal, and the acceleration is zero. See Figure 14.

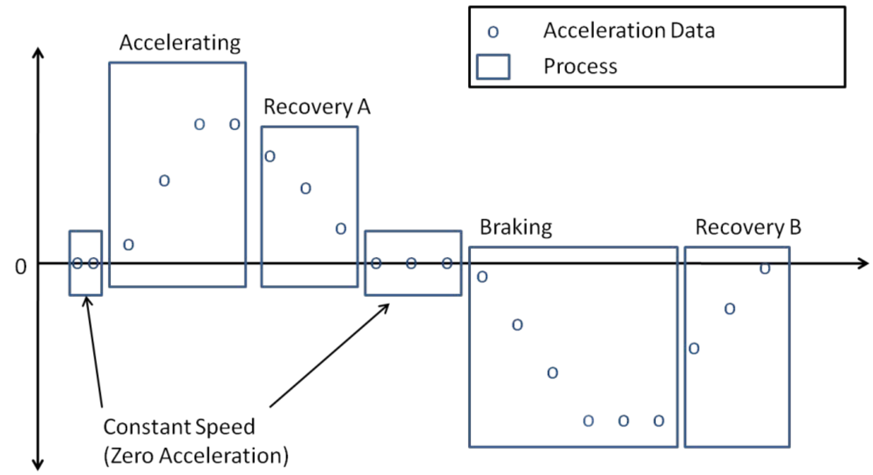


Figure 14. Conceptual explanations of the five processes.

Partitioning of Processes and Determining Target Acceleration

A process is a set of one or more instantaneous acceleration data observed from the moment a movement is initiated to the moment the movement is completed. It is assumed that the process includes a period to reach a target operating acceleration. The target operating acceleration of the accelerating or braking process is the highest or lowest acceleration points, respectively, of the process. Other processes have target operating accelerations of 0. Table 5 shows the rules for recognizing types of processes and for determining the target operating acceleration for each process. Figure 15 shows the flow chart of the algorithm that determines the type of process based on the rules. In Figure 15, Acc_t is an instantaneous acceleration at time t, and $dAcc_t/dt$ is jerk, variation of acceleration, from time t-1 to time t. Figure 16 shows the elapsed time versus acceleration and depicts the separated processes and the target operating acceleration for each process.

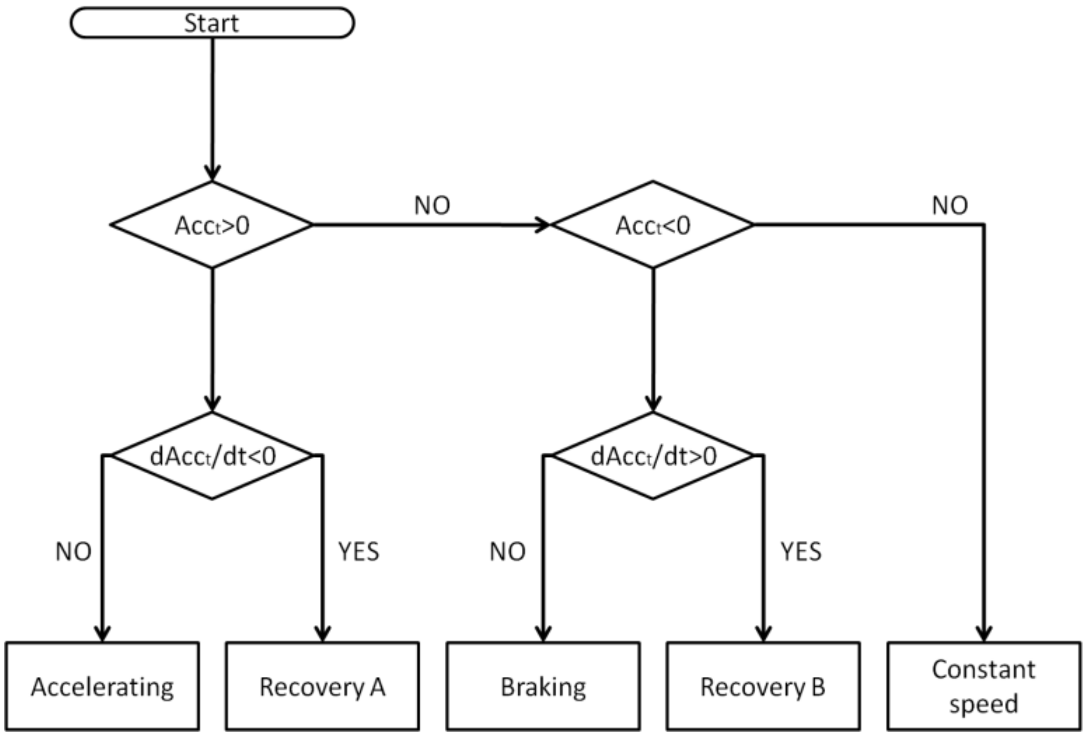


Figure 15. Flow chart of the separating processes algorithm.

Table 5. Rules for process identification and operating acceleration determination.

Process	**Condition 1 (value of acceleration)**	**Condition 2 (jerk)**	**Target operating acceleration**
Accelerating	Positive	Not negative	Highest value
Braking	Negative	Not positive	Lowest value
Recovery A	Positive	Negative	Zero
Recovery B	Negative	Positive	Zero
Constant Speed	Zero	-	Zero

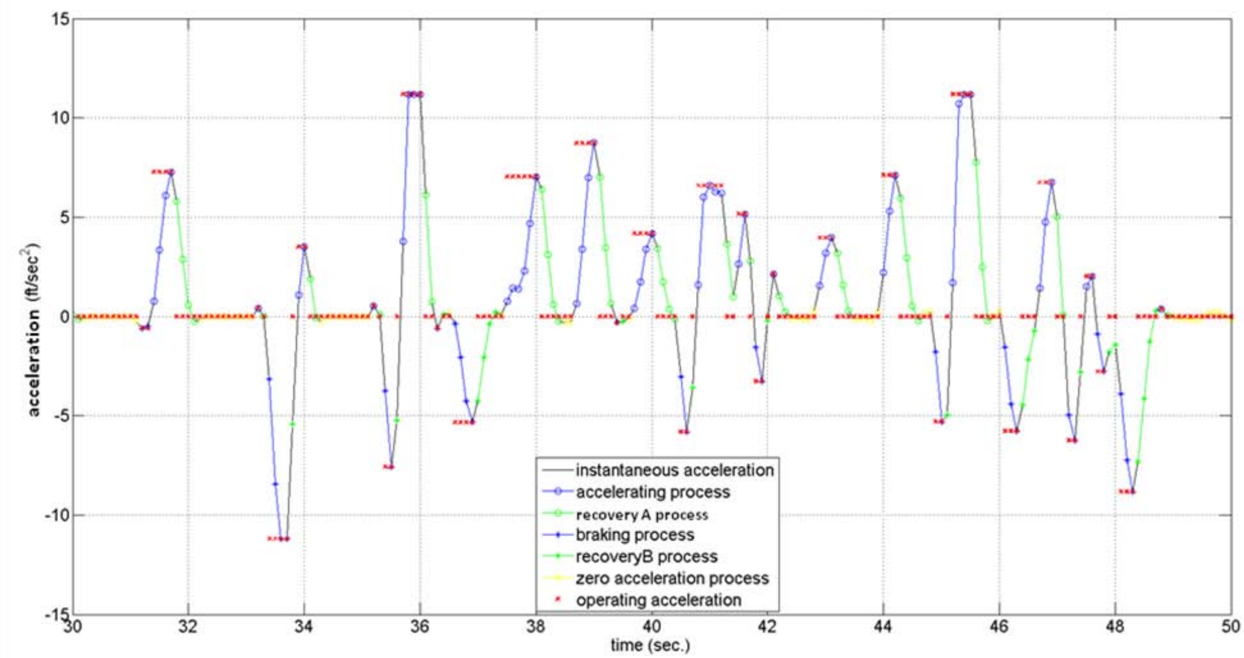


Figure 16. Time versus acceleration diagram for an example recognized process and its target operating acceleration.

Estimation of Emissions and Fuel Consumption by VT-Micro

The Virginia Tech Microscopic Energy and Emission Model (VT-Micro Model)

The VT-Micro model was developed through experimentation with numerous polynomial combinations of speed and acceleration levels using chassis dynamometer data collected by the Oak Ridge National Laboratory (ORNL) (Ahn et al. 2004; Rakha et al. 2004; Rakha et al. 2011). The ORNL data consist of nine normally emitting vehicles, including six light-duty automobiles and three light-duty trucks, to produce an average vehicle consistent with average vehicle sales in the United States in terms of engine displacement, vehicle curb weight, and vehicle type. The data contained information from 1,300 to 1,600 individual measurements of speed and acceleration records for each vehicle and a combination of measures of effectiveness (MOE) in terms of fuel consumption and emissions depending on the operation of the vehicle. Typical values of acceleration ranged from -1.5 to 3.7 m/s^2 at increments of 0.3 m/s^2 (-5 to 12 ft/s^2 at 1-ft/s^2 increments). The speed values varied from 0 to 33.5 m/s (0 to 121 km/h or 0 to 110 ft/s) at increments of 0.3 m/s (1 ft/s).

The VT-Micro-developed regression model based on the ORNL data consists of two formulas, depending on whether the driver is accelerating or braking, as shown in Equations 4 and 5. The value of vehicle acceleration is combined with linear, quadratic, and cubic terms of speed and acceleration to obtain the fuel consumption and emission rates (Rakha et al. 2004). The developed models resulted in good fits to the ORNL data (R^2 in excess of 0.92 for all MOEs).

For accelerating (Equation 4):

$$MOE = e^{\left(L_0 + L_1 a + L_2 a^2 + L_3 a^3 + L_4 u + L_5 u^2 + L_6 u^3 + L_7 ua + L_8 ua^2 + L_9 ua^3 + L_{10} u^2 a + L_{11} u^2 a^2 + L_{12} u^2 a^3 + L_{13} u^3 a + L_{14} u^3 a^2 + L_{15} u^3 a^3\right)}$$

For braking (Equation 5):

$$MOE = e^{\left(M_0 + M_1 a + M_2 a^2 + M_3 a^3 + M_4 u + M_5 u^2 + M_6 u^3 + M_7 ua + M_8 ua^2 + M_9 ua^3 + M_{10} u^2 a + M_{11} u^2 a^2 + M_{12} u^2 a^3 + M_{13} u^3 a + M_{14} u^3 a^2 + M_{15} u^3 a^3\right)}$$

where MOE represents one of the following outputs: the instantaneous fuel consumption in liters/s or the emission rate of HC, CO, NO_x, or CO_2 in mg/s. u is the instantaneous vehicle speed in km/h, and a is the instantaneous vehicle acceleration in km/h-s, variation of acceleration in km/h per second. Ls and Ms represent the regression model coefficients for acceleration and braking, respectively. The regression models corresponding to each type of emission or fuel consumption use different sets of 16 Ls and 16 Ms coefficients for accelerating and braking conditions.

Smoothing Trajectory Data

The VT-Micro model uses three types of trajectory data, collected every second:

1. instantaneous speed data
2. instantaneous acceleration data
3. vehicle type

Thus, the NGSIM trajectory data must be smoothed from deci-second-based trajectory data to second-based trajectory data. Acceleration and speeds from NGSIM trajectory data were recollected at every second and made into input data for VT-Micro.

Outputs of VT-Micro Model

The VT-Micro model produces profile data of individual vehicle fuel consumption (liters/s) and four emissions (g/s):

1. hydrocarbons (HC)
2. carbon monoxide (CO)
3. oxides of nitrogen (NOx)

4. carbon dioxide (CO2)

Total emissions per vehicle were computed using the profile data.

High Emitter Existence

The existence of high emitters in the observed vehicles in each time period was verified, and the rates of high emitters in all observed vehicles was estimated. Vehicles were sorted in ascending order of each emission and then divided into 100 equal parts. Twenty of the 100 parts were chosen, and then the highest emitters in each part were selected. Figure 17, Figure 18, and Figure 19 show comparisons of the selected emitters' emissions and fuel consumptions in the first time period, the second time period, and third time period, respectively. These figures show that high emitters could constitute approximately 15% of all emitters under non-congested conditions, and the contribution of high emitters to total emissions of CH, CO, and NO_x tends to decrease in proportion to congestion. These figures also show that the difference between low emitters and moderate emitters is reduced out of proportion with congestion.

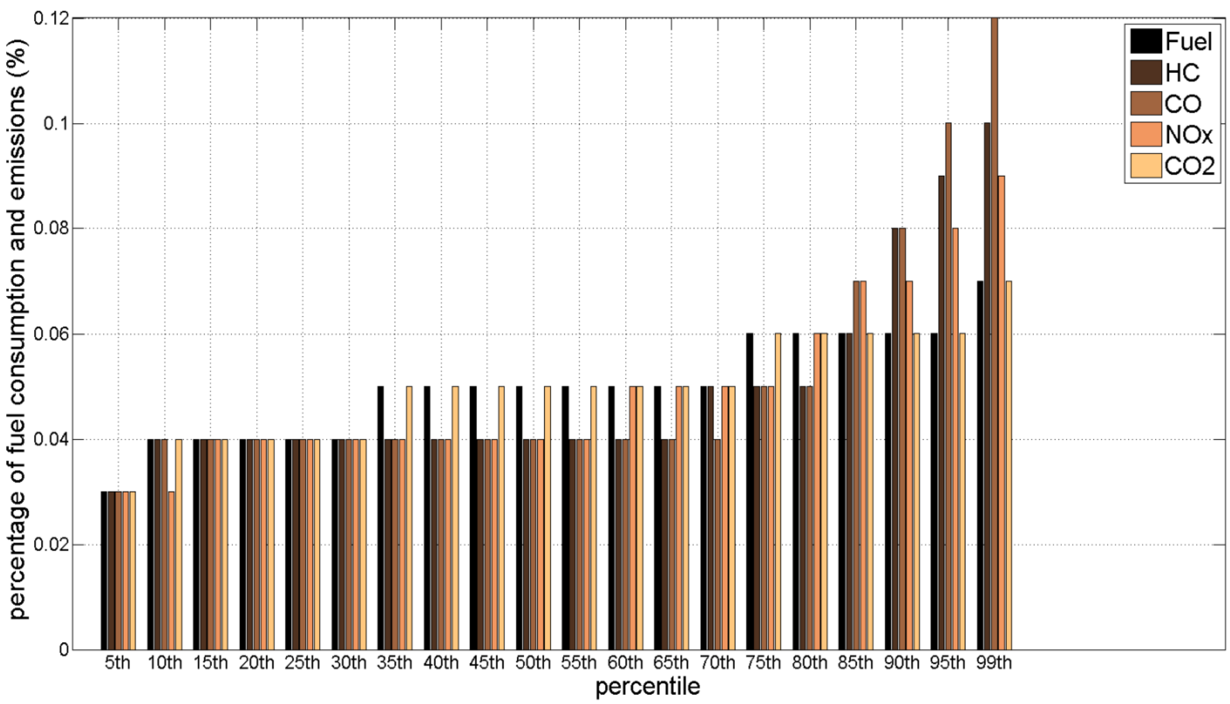


Figure 17. Percentage of fuel consumption and emissions for different percentiles of vehicles in the first time period (non-congested condition).

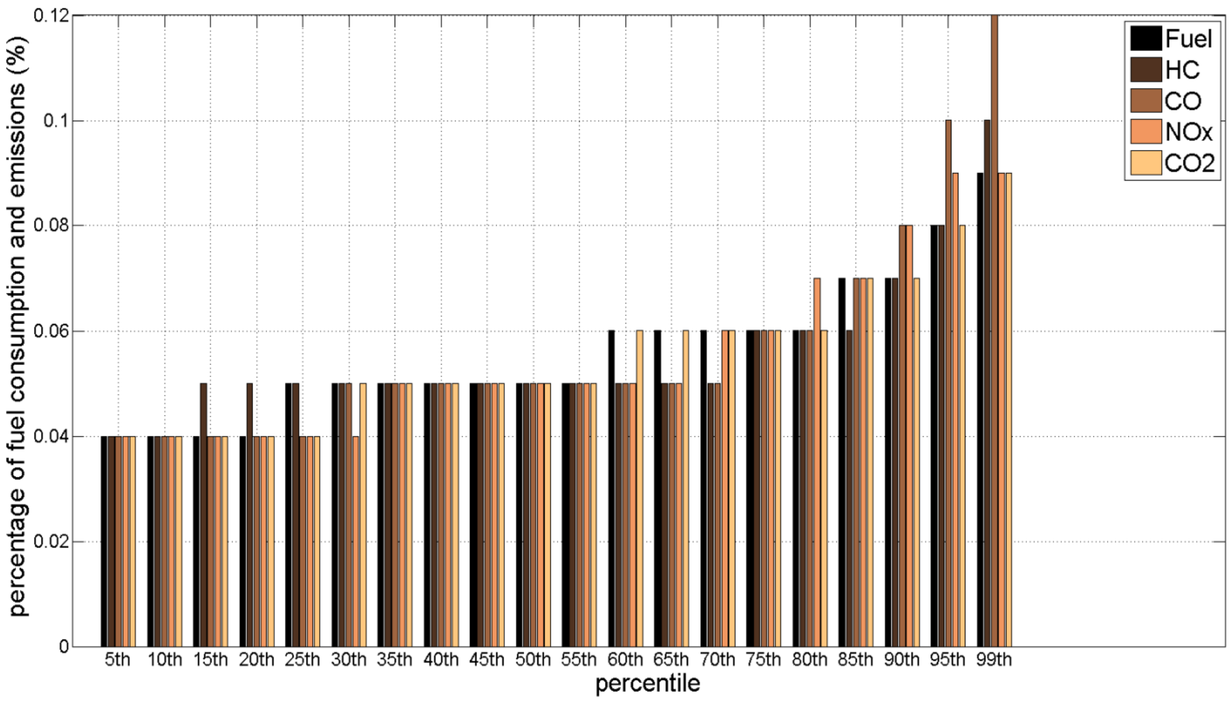


Figure 18. Percentage of fuel consumption and emissions for different percentiles of vehicles in the second time period (congested condition).

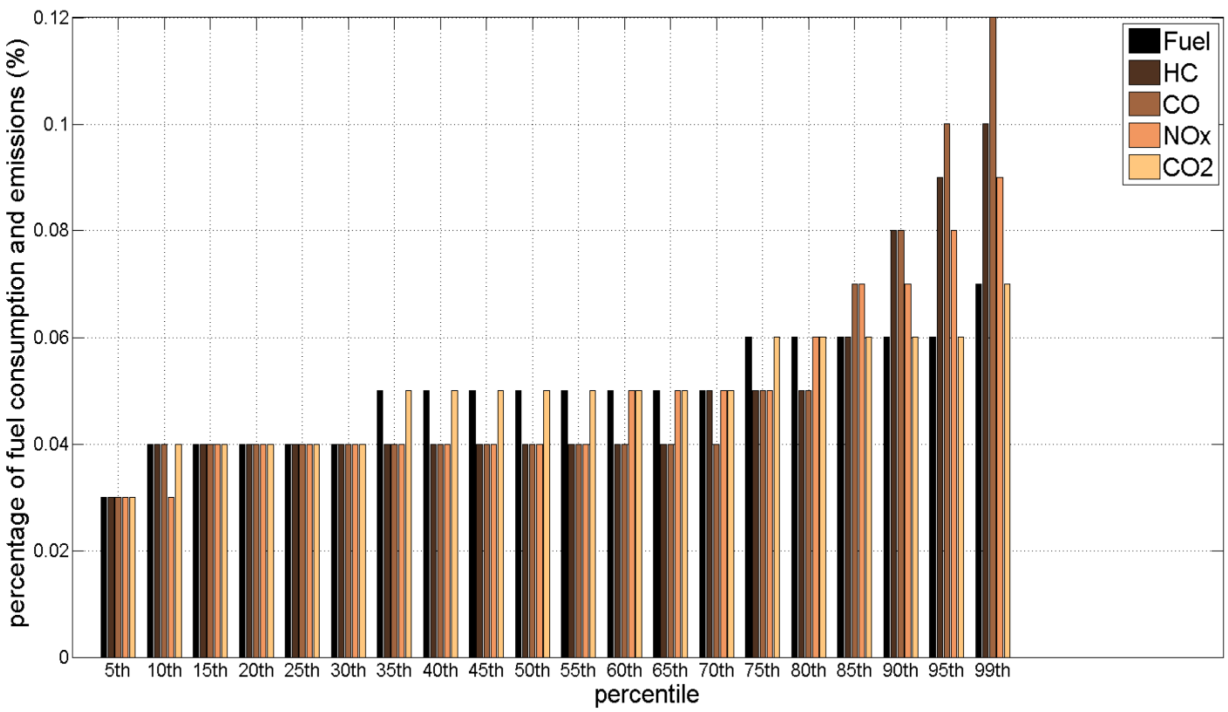


Figure 19. Percentage of fuel consumption and emissions for different percentiles of vehicles in the third time period (congested condition).

Features for Classification

This research focused only on the instantaneous movements of single vehicles. The processes were divided into three groups: accelerating processes, braking processes, and other processes. Average target operating acceleration and average target operating brake of individual vehicles were computed as a key feature of each vehicle for classification. Mean and standard deviations of average operating target acceleration and brake were then computed and used to separate the drivers into three groups: defensive drivers, moderate drivers, and aggressive drivers. Figure 20 shows a conceptual view of the classification. The classification assumes that the average operating acceleration follows a normal distribution, and it then divides drivers into three partitions using thresholds for standard deviation of average target operating acceleration. The standard deviations from mean values of average target operating accelerations to the average target operating accelerations of individual vehicles were computed and used to set feature vectors in two dimensions for classification.

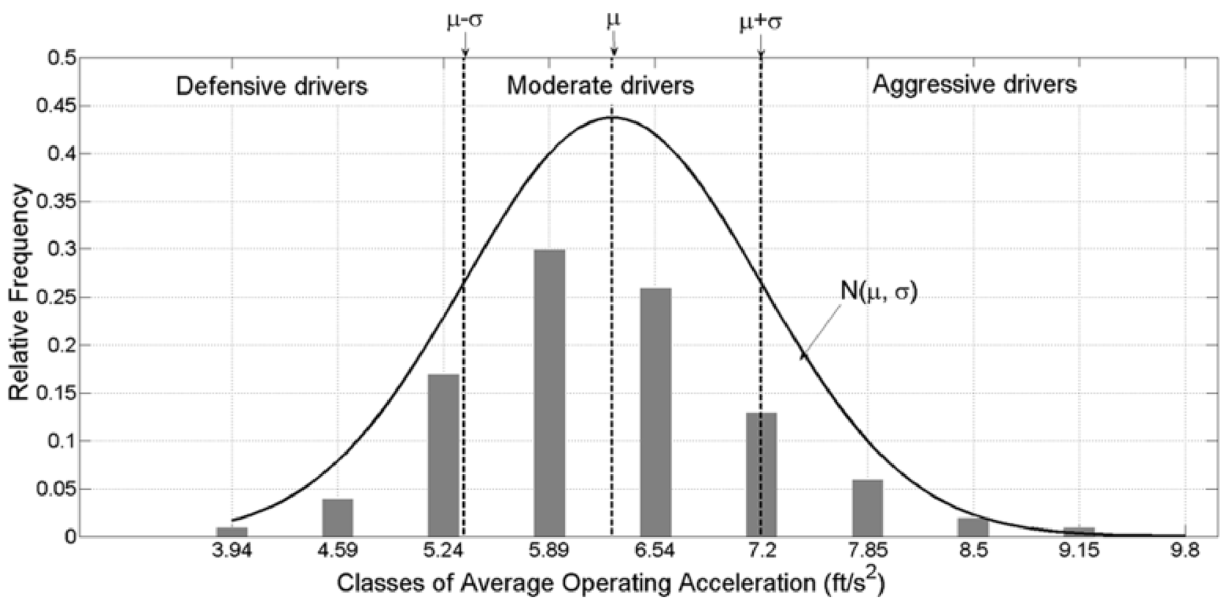


Figure 20. Distribution of the average target operating acceleration in the non-congested condition.

Aggressive drivers have higher target operating acceleration and braking than moderate drivers, who have higher target operating acceleration and braking than defensive drivers. It was expected that high average target operating acceleration significantly would increase the amount of emissions and fuel consumption. The standard deviation of observations is defined as Equation 6:

$$\sigma = E[X - \mu]$$

where the X is observations, μ is the mean of observations, E[X] is the expected value of observations (X), and σ is the standard deviation of observations (X).

The x-μ is called the deviation of an observation in statistics. The deviation of an observation means a distance between an observation and the mean of observations. Thus, to use standard deviation of average target operating acceleration as classification thresholds, distances between the average target operating accelerations of individual drivers and the mean of the average target operating accelerations of all drivers were calculated by Equations 7 and 8:

$$D_{acc,i} = A_i - A_{avg}$$

$$D_{brk,i} = B_{avg} - B_i$$

where $D_{acc,i}$ and $D_{brk,i}$ are the differences in average operating acceleration and braking, respectively, of the ith vehicle from the mean of average operating acceleration and braking of all vehicles. A_i and B_i are the average operating acceleration and braking, respectively, of the ith vehicle, and A_{avg} and B_{avg} are the mean of average operating acceleration and braking, respectively, of all vehicles under consideration. These distances are the differences of average target operating acceleration or braking.

Table 6 shows that mean acceleration was at its lowest and mean braking was at its highest in the second time period, but the differences between the highest and lowest values are rather small. However, the standard deviations for acceleration and braking were highest and lowest, respectively, in the first time period, which is a logical outcome. That is, the congestion conditions greatly influenced not the average acceleration and braking values but rather their variability. Therefore, the average operating target acceleration was used as the measure to classify drivers.

Table 6. Mean and standard deviation of operating acceleration and braking.

	First time period		**Second time period**		**Third time period**	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Acceleration	6.28	0.91	5.93	0.84	6.02	0.74
Braking	-6.31	0.88	-5.92	0.78	-5.93	0.74

Classification Results

Figure 21 shows the results of classification using the standard deviation of average target operating acceleration as the threshold. The average operating acceleration and braking values of individual vehicles in each class in the non-congestion condition were compared with the thresholds. This research defined aggressive drivers as ones who had the highest value of average operating acceleration relative to others. In contrast, defensive drivers had the relatively lowest values of average operating acceleration.

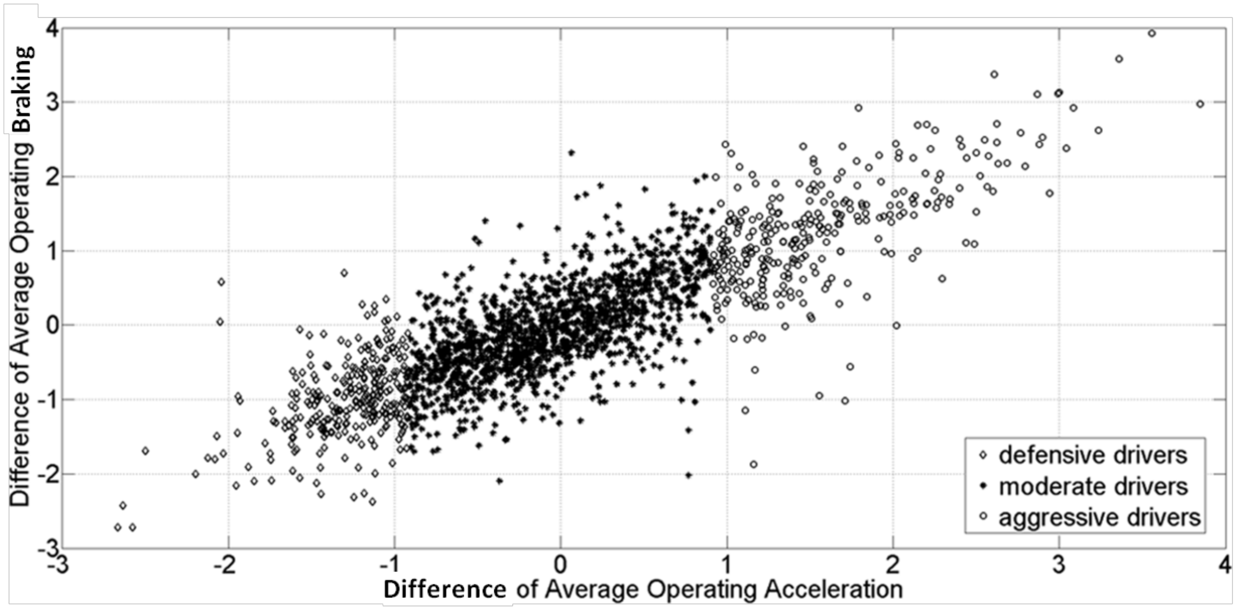


Figure 21. Classification of driving pattern based on acceleration in the non-congestion condition.

Table 7 shows that moderate drivers accounted for approximately 70% of all drivers, and defensive and aggressive drivers each accounted for roughly 15% of all drivers in each time period.

Table 7. Number of vehicles in each category.

	First time period	Second time period	Third time period	Total # of vehicles
Defensive	301	289	256	846
Moderate	1,448	1,275	1,291	4,014
Aggressive	303	272	243	818
Total # of vehicles	2,025	1,836	1,790	5,678

Evaluation of Emissions

Total fuel consumption and emissions for each vehicle in each class were evaluated using the VT-Micro model, and the results were compared. For input arguments to the VT-Micro model, it was assumed that all vehicles were of the same vehicle category, category 4: Light-duty vehicle, Model Year >= 1995, Engine Size < 3.2 liters, and Mileage < 83653 mile. Evaluation was conducted based on acceleration and speed profiles.

Evaluating the Contribution of Aggressive Drivers to Emissions and Fuel Consumption

Figure 22 compares the percentage of the number of vehicles in each driving group (aggressive, moderate, and defensive) to the percentage of emissions and fuel consumption of each group. In this figure, it was found that aggressive drivers constitute a small fraction of the vehicle fleet, but produce more percentage of HC, CO, and NO_x than the fraction. In the first time period, aggressive drivers constituted about 14. 77 % of entire vehicles, but produced about 21.52 %, 23.40%, and 20.40% of total amount of HC, CO, and NO_x, respectively. Similarly, in the second time period, aggressive drivers constituted about 14.81% of vehicles, but produced 19.08 %, 21.6%, and 19.79 % of total amount of HC, CO, and NO_x, respectively. In the third time period, aggressive drivers constituted about 13.58% of vehicles, but produced 16.01 %, 18.63%, and 17.69 % of total amount of HC, CO, and NO_x, respectively. In terms of fuel consumption, aggressive drivers tended to consume less fuel relatively. That is, aggressive driving did not influence the total fuel consumption pattern because the average vehicle speed was relatively low, ranging from 17 to 25 mph. Moderate drivers consumed slightly more fuel than others in congested condition. The defensive drivers produced slightly less emissions than the moderate drivers in all conditions. The inter-driver-category difference between emissions in the non-congestion condition was greater than that in the congestion condition.

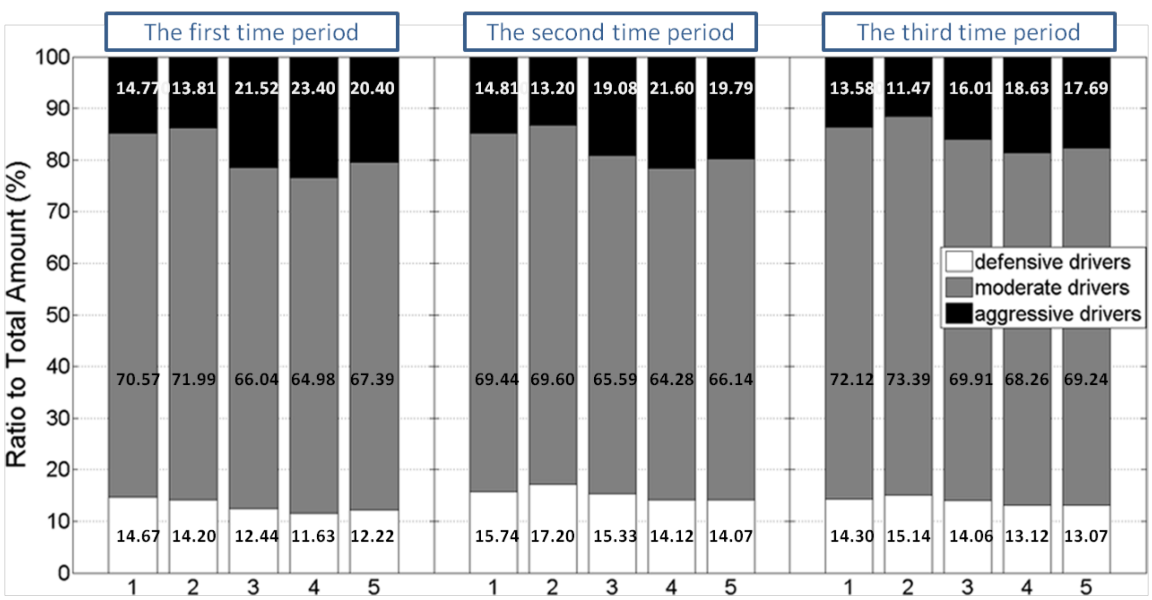


Figure 22. Results of fuel consumption and emissions in each condition.

Note: 1= number of vehicles, 2= fuel consumption, 3= hydrocarbons, 4= carbon monoxide, and 5= oxides of nitrogen.

Evaluating Environmental Impact of Driving Behavior Changes

Two alternative driving behavior scenarios were designed and then compared to the base scenario. The alternative scenarios were the following:

Alternative 1: All aggressive drivers changed their driving behaviors to those of moderate drivers.

Alternative 2: All aggressive and moderate drivers changed their driving behaviors to those of defensive drivers.

To estimate the changes in emissions and fuel consumption due to driving behavior changes, it was assumed that all drivers who changed their driving behavior had the average emission and fuel consumption values per vehicle of their new cluster.

Table 8 shows the results of comparing the two alternative scenarios with the base scenario. Based on these results, if aggressive drivers change their driving to moderate driving (Alternative 1), reductions to the amount of HC, CO and NO_x in the first time period are approximately 1,990.6 milligrams (7.7%), 6,7396 milligrams (9.8%), and 3,541.7 milligrams (6.3%), respectively. Reductions to the amount of HC, CO, and NO_x in the second time period are approximately 1,191.6 milligrams (5.0%), 4,7490 milligrams (7.9 %), and 2,844.6 milligrams (5.7%), respectively. Similarly, during the third time period, the reductions to the amount of HC, CO, NO_x are approximately 681.6 milligrams (2.9 %), 33,893 milligrams (5.8 %), and 2,279.7 milligrams (4.7 %), respectively.

If both aggressive and moderate drivers change their behavior to defensive driving (Alternative 2), the reductions to the amount of HC, CO, and NO_x in the first time period are approximately 3,912.3 milligrams (15.1%), 143,071 milligrams (20.8%), and 9,351.2 milligrams (16.7%), respectively. In the second time period, the reductions to the amount of HC, CO, and NO_x are approximately 677.5 milligrams (2.9%), 62,062.1 milligrams (10.3%), and 5,398.9 milligrams (10.8%), respectively. In the third time period, the reductions to the amount of HC, CO, and NO_x are approximately 387 milligrams (1.6%), 48,860 milligrams (8.3%), and 4,167.4 milligrams (8.6%), respectively.

Table 8. Environmental impacts of driving behavior changes according to alternative 1 and alternative 2

Time period	Emissions or fuel consumption	Base	Alternative 1			Alternative 2		
		Total	Total	Changed	%	Total	Changed	%
First time period	Fuel	136082ml	137797.1ml	1715.1ml	1.3%	131736.2	-4345.8ml	-3.2%
	HC	25866mg	23875.4mg	-1990.6mg	-7.7%	21953.7gm	-3912.3mg	-15.1%
	CO	686980mg	619584mg	-67396mg	-9.8%	543909gm	-143071mg	-20.8%
	NO_x	56131mg	52589.3mg	-3541.7mg	-6.3%	46779.8mg	-9351.2mg	-16.7%
Second Time period	Fuel	131991ml	134149ml	2158ml	1.6%	144298.2ml	12307.2ml	9.3%
	HC	23639mg	22447.4mg	-1191.6mg	-5.0%	22961.5mg	-677.5mg	-2.9%
	CO	601293mg	553803mg	-47490mg	-7.9%	539230.9mg	-62062.1mg	-10.3%
	NO_x	50028mg	47183.4mg	-2844.6mg	-5.7%	44629.1mg	-5398.9mg	-10.8%
Third time period	Fuel	141915ml	145243.1ml	3328.1ml	2.3%	150181.6ml	8266.6ml	5.8%
	HC	23654mg	22972.4mg	-681.6mg	-2.9%	23267mg	-387mg	-1.6%
	CO	587667mg	553774mg	-33893mg	-5.8%	538807mg	-48860mg	-8.3%
	NO_x	48731mg	46451.3mg	-2279.7mg	-4.7%	44563.6mg	-4167.4mg	-8.6%

Note: Alternative 1: Aggressive drivers change to moderate drivers. Alternative 2: Aggressive and moderate drivers change to defensive drivers.
Note: Fuel= fuel consumption, HC = hydrocarbons, CO = carbon monoxide, and NO_x = oxides of nitrogen.

Additional Classification

So far, the study of the environmental impact of acceleration changes has found that aggressive drivers greatly contributed to the total amount of emissions and fuel consumption. One question remains: whether or not emissions and fuel consumption increase in proportion to increases in aggressiveness. To determine this relationship, the number of classification thresholds increased from two to seven thresholds: μ-1.5σ, μ-σ, μ-0.5σ, μ, μ+0.5σ, μ+σ, and μ+1.5σ. Figure 23 shows the seven partitions on a normal distribution of average target operating acceleration for each time period. Figure 24 shows the results of the classification using seven thresholds for each time period with the additional classifications. The first cluster is the very defensive drivers group, the second cluster is the defensive drivers group, the third cluster is the slightly defensive drivers group, the fourth cluster is the defensively moderate drivers group, the fifth cluster is the aggressively moderate drivers group, the sixth cluster is the slightly aggressive drivers group, the seventh cluster is the aggressive drivers group, and the eighth cluster is the very aggressive drivers group. It was expected that the very aggressive drivers would produce more emissions of HC, CO, and NO_x, but that driving types would not influence fuel consumption, based on the former classification's results.

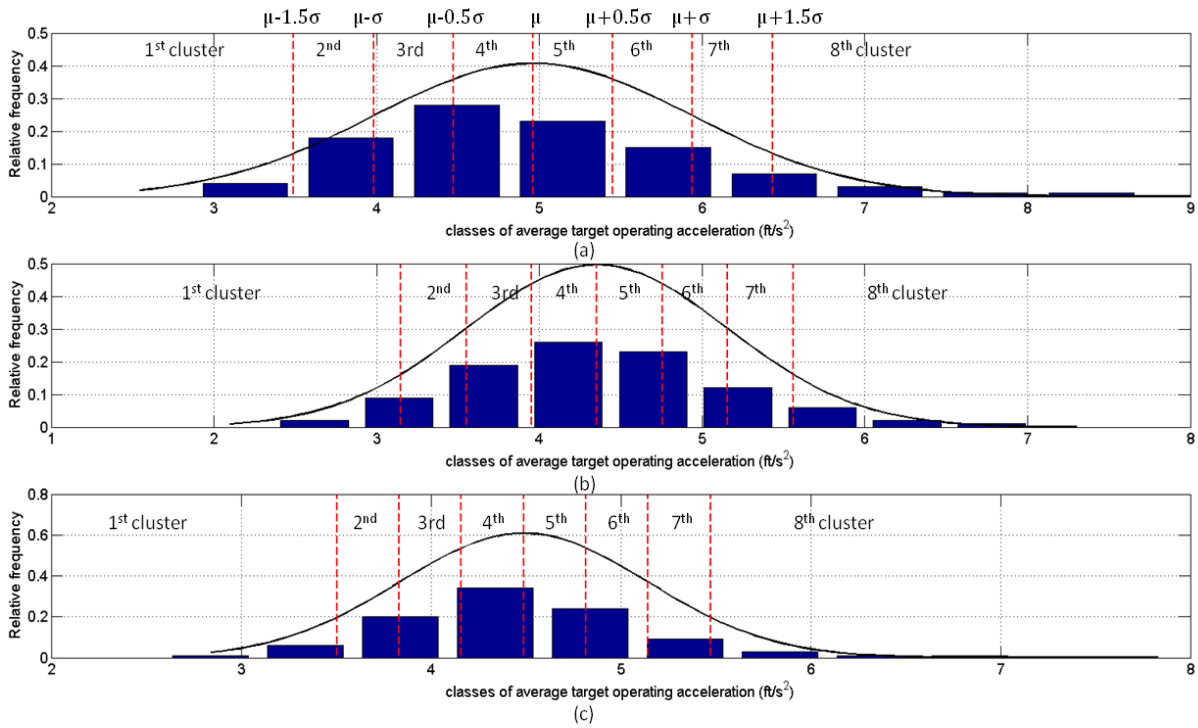


Figure 23. Distribution of the average target operating acceleration and partitions in each time period: (a) the first time period, (b) the second time period, and (c) the third time period.

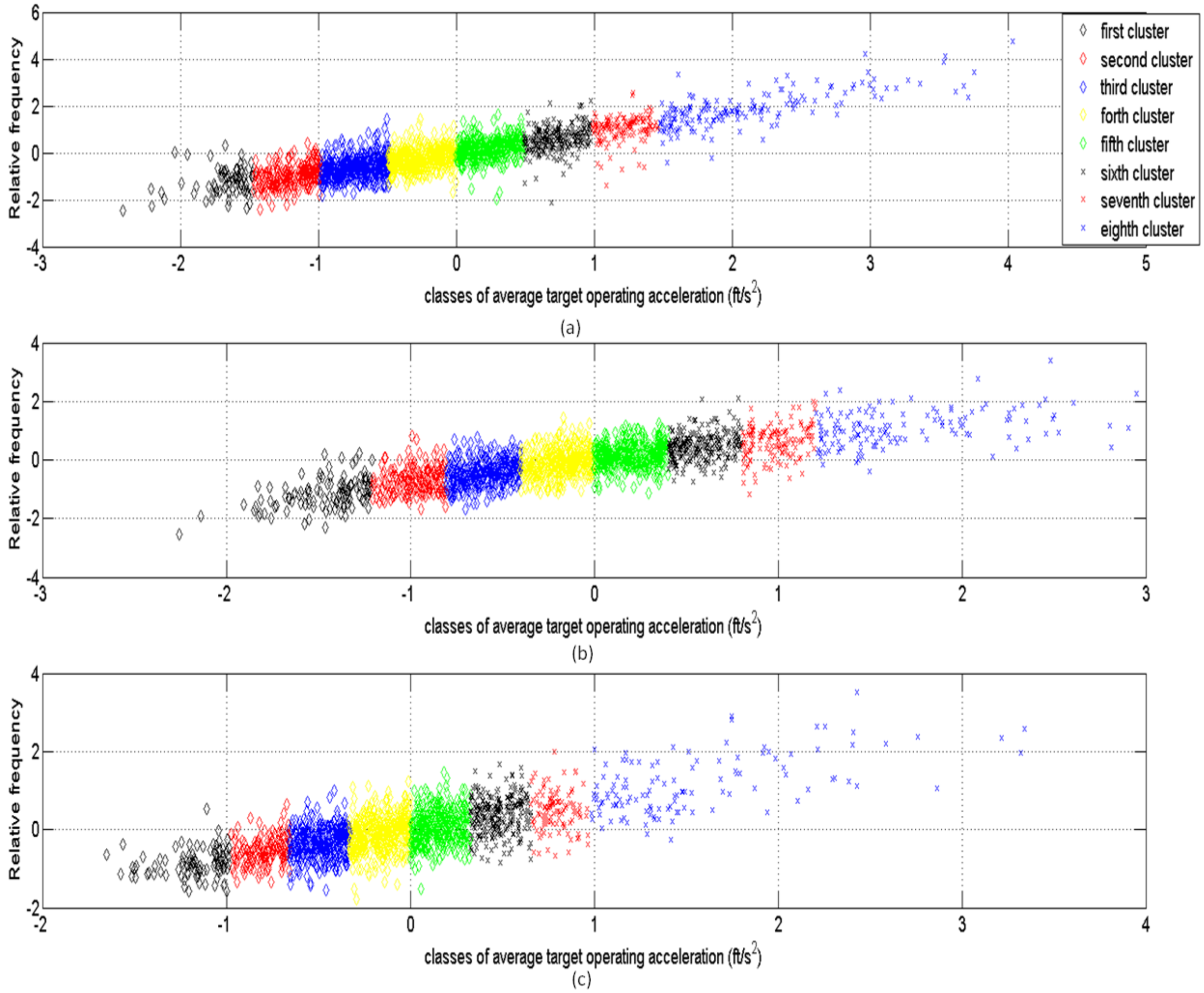


Figure 24. Results of classification using seven thresholds in each time period: (a) the first time period, (b) the second time period, and (c) the third time period.

Using VT-Micro, the total fuel consumption in mile-liters per vehicle was compared to emissions of HC, CO, and NO_x in mile-grams per vehicle.

Figure 25 shows that driving types were not related to fuel consumption in all time periods. In the first time period, the slightly aggressive drivers consumed the most fuel, and the very defensive drivers consumed the least fuel. However, in the second time period, the slightly defensive drivers consumed the most fuel, and the very aggressive drivers consumed the least fuel. In the third time period, the very defensive drivers consumed the most fuel, and the very aggressive drivers consumed the least fuel. This evidence indicates that the fuel consumption was not related to driving patterns based on the average target operating acceleration.

However, emissions of HC, CO, and NO_x were related to the average target operating acceleration. Figures 26 through 28 show that the very aggressive drivers produced significantly more emissions of HC, CO, and NO_x. The aggressive drivers also produced high emissions. Consequently, aggressive and very aggressive drivers could be assigned to the group of high emitters of HC, CO, and NO_x. The differences in emissions of HC, CO, NO_x among the other groups were not significant.

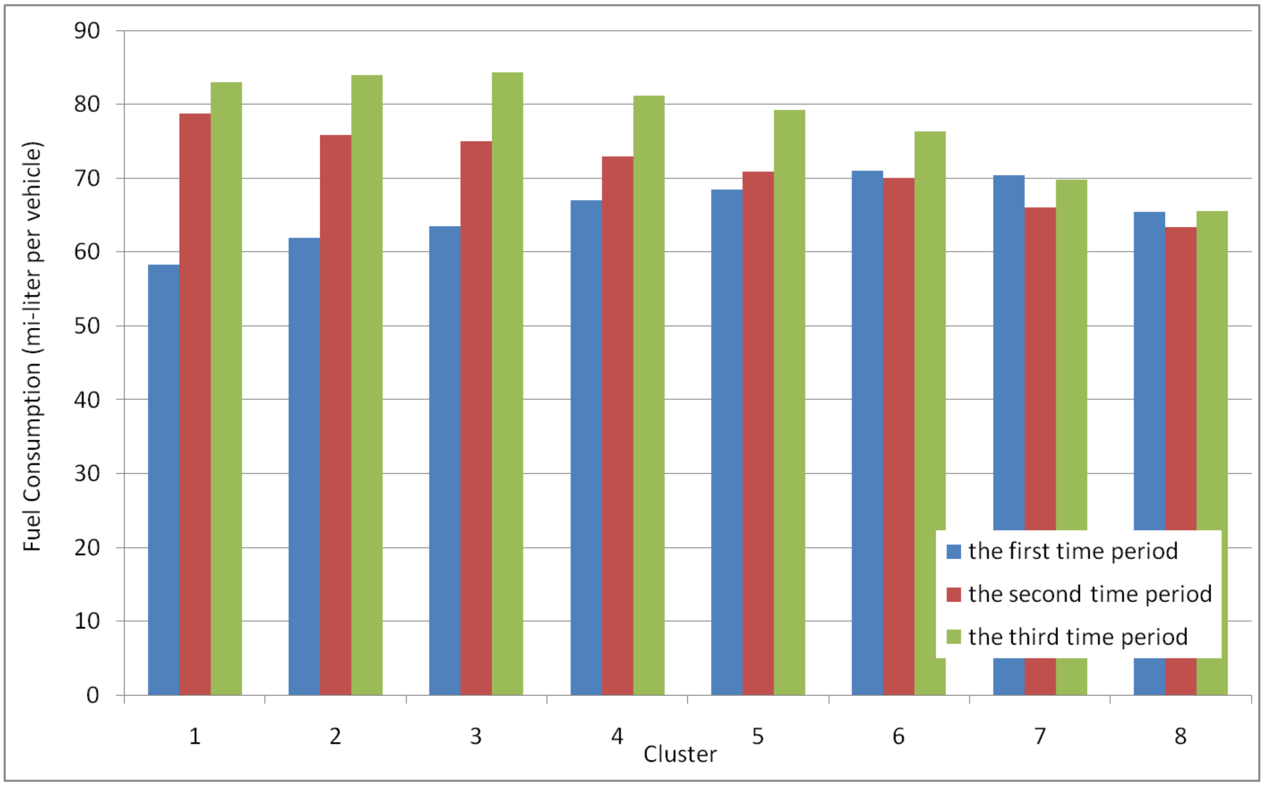


Figure 25. A comparative analysis of fuel consumption by cluster and time-period.

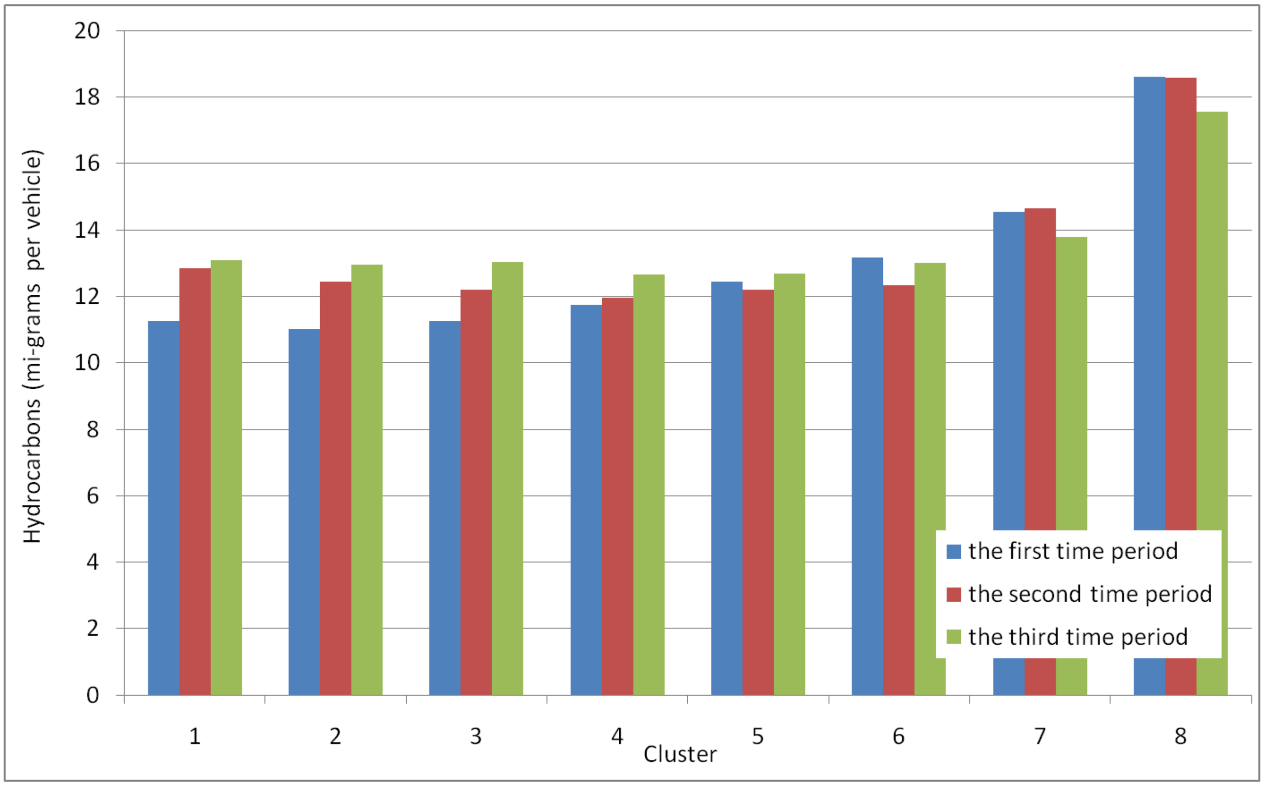


Figure 26. A comparative analysis of hydrocarbons (HC) by clusters.

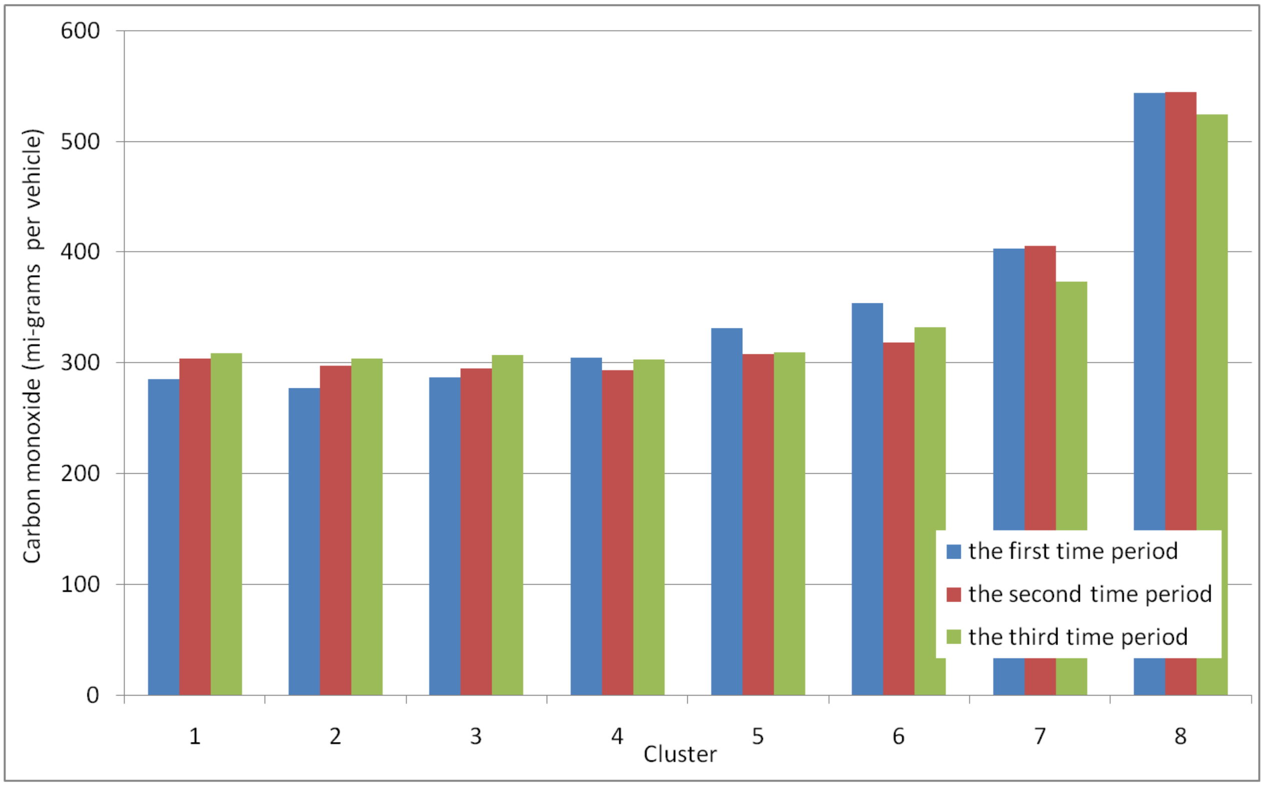


Figure 27. A comparative analysis of carbon monoxide (CO) by clusters.

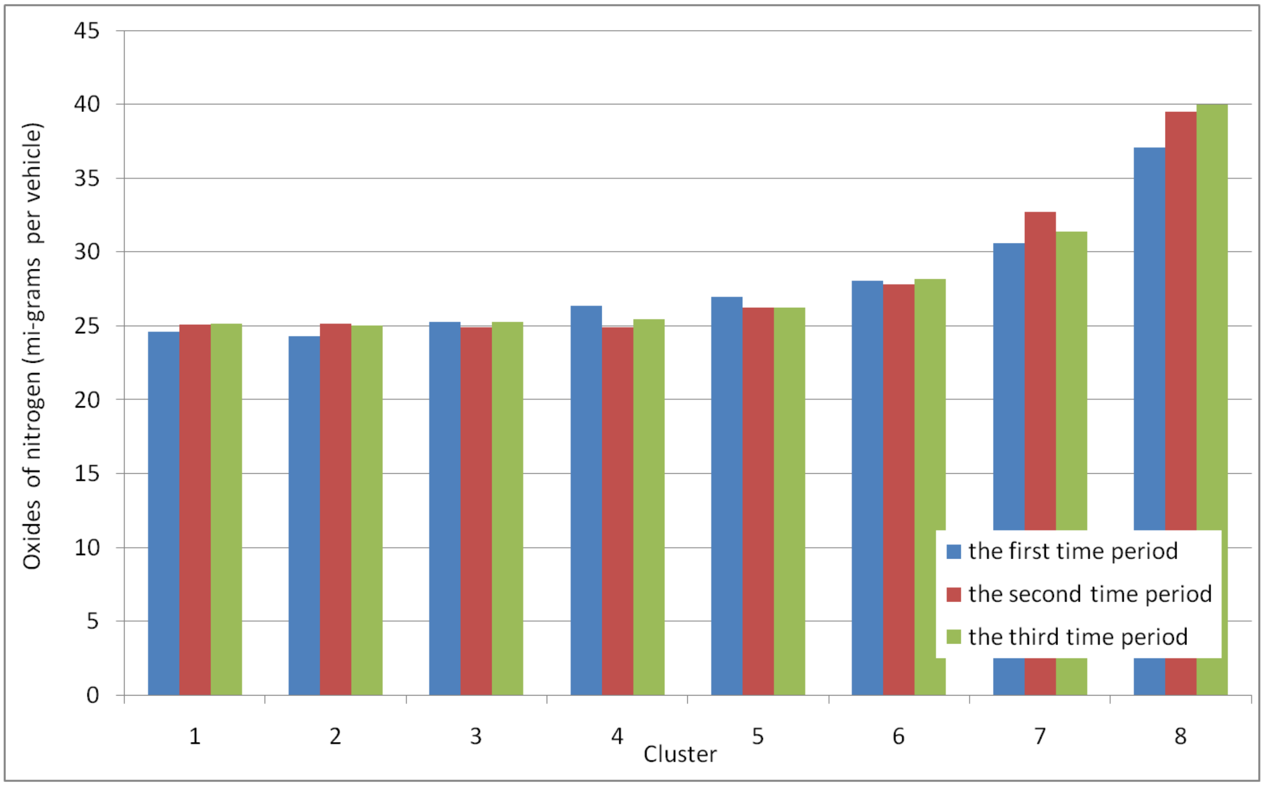


Figure 28. A comparative analysis of oxides of nitrogen by clusters.

Conclusions

This study estimated the influences of driving behaviors on emissions and fuel consumption. Drivers were classified into three groups based on their acceleration behavior. Their emissions and fuel consumption were evaluated using the VT-Micro model, using the drivers' velocity and acceleration profiles as input parameters.

Aggressive driving has traditionally been considered from a traffic safety point of view. However, this research considered it from the environmental point of view. Aggressive drivers used greater and more frequent acceleration than their moderate counterparts. Defensive drivers used less acceleration than their moderate counterparts. The aggressiveness of each driver is not related to his or her socio-economic or socio-culture characteristics because the available NGSIM data did report these characteristics.

This study focused on the instantaneous accelerating and braking behaviors of drivers to estimate the emissions and fuel consumption of vehicles under congestion and non-congestion traffic conditions. There are many other influencing factors that were not considered such as type of vehicle, engine characteristics, road condition, and weather. These are beyond the scope of this study and the availability of the data.

The results indicate that different driving behaviors among the three drivers' groups did not significantly influence fuel consumption because the average speed was low in all time periods. However, they did influence the emission of HC, CO, and NO_x, particularly in the non-congested condition. This is because the vehicles were classified according to average operating acceleration. Because headways between vehicles in the non-congestion condition were definitely greater than in the congestion condition, drivers in the non-congestion condition could vary their acceleration and braking more than those in the congestion condition. The speeds of vehicles were rarely different in all time periods because most vehicles were following preceding vehicles at similar speeds. The resulting conclusion is that speed was the primary factor influencing fuel consumption. But average operating acceleration and braking did influence the amount of HC, CO, and NO_x emissions. Therefore, if education or advertisement can change aggressive drivers to moderate or defensive drivers, emissions of HC, CO, and NO_x should decrease, particularly under non-congestion conditions. In non-congestion conditions, the amount of CO should decrease by 9.8% if aggressive drivers change to moderate drivers, and it should decrease by 20.8% if aggressive and moderate drivers change to defensive drivers. The reductions in emissions for the congestion conditions are still significant but are less than those in the non-congested condition.

In additional classification, I found that aggressive drivers based on the average target operating acceleration could be assigned to high emitters' group, but this classification using the average target operating acceleration was not enough to recognize low emitters, low consumer of fuel, and high consumer of fuel.

Future work will consider other algorithms and features to better classify the driving patterns of drivers. More detailed studies of the interaction among vehicles are also planned in order to better assess the changes in aggressiveness of following drivers due to the actions of preceding vehicles, such as lane-changing. Methods must be developed that would apply the obtained classified driving patterns to future micro-simulation logistics.

Chapter 6. Study Case 2: Development and Comparison of Driving and Environmental Impact Characteristics of Different Driver Types

This chapter examines individual driver behavior in terms of its environmental impact on overall fleet behavior. Trajectory data for individual drivers were extracted from the NGSIM database, the corresponding emissions and fuel consumptions outputs for each trajectory were calculated using the Comprehensive Modal Emissions Model (CMEM) software, and all the output was clustered by both driving data and environmental data. The clustering procedure produced distinct classes of drivers with significant environmental impacts.

Driving Data

The driving behavior of individual vehicles is obviously influenced by either traffic flow with congestion or without congestion (May, 1990). The fundamental characteristics of traffic flow can be considered as variables representing characteristics of individual driving behavior in the traffic flow. The fundamental traffic flow characteristics are flow, speed, and density. At the macroscopic analysis level, these characteristics have been observed as three corresponding variables: flow rates, average speeds, and density rates. At the microscopic analysis level, the corresponding variables are time headways, individual speeds, and distance headways. Table 9 summarizes these variables.

Table 9. Framework for the fundamental characteristics of traffic flow.

Traffic Characteristics	Microscopic Level (individual units)	Macroscopic Level (groups of units)
Flow	Time headway	Flow rates
Speed	Individual speeds	Average speeds
Density	Distance headway	Density rates

Source: Adolf D. May, Traffic Flow Fundamentals, 1990.

Individual Speed

Individual speeds are variations of location in unit time and are usually included in the trajectory data (May, 1990). The NGSIM trajectory data includes the instantaneous speed in feet per second recorded every 0.1 s. Under low-traffic flow conditions, the average speed could be related to the driver's desired speed, and the standard deviation could be related to stability of driving behavior. Under heavy-traffic flow conditions, the average individual speed would converge on the average speed of traffic flow, and the standard deviation would be close to zero.

Distance Headway (Spacing)

Distance headway, sometimes called as spacing, can be computed by taking the difference between the locations of subject vehicle and preceding vehicle (May, 1990). Distance headway in feet is included in the NGSIM trajectory data (Cambridge Systematics, 2005a; 2005b; 2005c). The distance headway is related to car-following theories such as Pipes' theory, Forbes' theory, and General Motors' theories. Because drivers base their own speed on relative speed and distance headway, the spacing can also be an important variable for characterizing driving behavior (Olson et al., 2010).

Relative Speed

Drivers can only know their own speed when they see the speedometer on their vehicle. Drivers could judge their own speed based on the difference between their own speed and that of the preceding vehicle (Olson et al., 2010). That difference is called the relative speed and is computed by Equation 9:

$$v_r = v_{subject} - v_{preceding}$$

Where v_r is the relative speed, $v_{subject}$ is speed of the subject vehicle, and $v_{preceding}$ is speed of the preceding vehicle.

The standard deviation of the relative speed could be related to stability of driving behavior, and the average relative speed could be related to sensitivity for speed of the preceding vehicle.

Time Headway

The time headway between vehicles is an important flow characteristic related to the safety, level of service, driver behavior, and capacity of transportation systems (May, 1990). In terms of the safety, the minimum time headway always represents safety in the event that the lead vehicle suddenly decelerates. Under very low-traffic flow conditions, the time headways vary greatly because all vehicles may be thought of as traveling independently of one another. Thus, the time headways can be considered random, and the random time headway distribution can be used for a microscopic analysis of time headways. This situation can be called the random headway state. Under heavy-traffic flow conditions where serious congestion has occurred, the time headway can be considered as almost constant because all vehicles are interacting with neighboring vehicles. This situation can be called the constant headway state, and the mean time headway can be calculated by Equation 10:

$$\bar{t} = \frac{3600}{V}$$

where $\bar{t}$ is mean time headway in seconds and V is average flow rates in vehicles per hour.

Under the absolutely constant headway state, the mean time headway can be constant. Equation 7, however, has been used to calculate the mean time headway using loop detector data collected at a fixed location. Conceptually, the time headway based on trajectory data can be calculated by Equation 11:

$$\mathrm{THW} = \frac{S}{v_r}$$

where THW is time headway based on trajectory data, S is distance headway (relative location), and v_r is instantaneous relative speed of the subject vehicle.

The distance headway can be measured simply by taking the distance from the front of preceding vehicle to the front of the subject vehicle, using the location data within the trajectory data. The instantaneous relative speed is computed by taking the difference between the speeds of host and preceding vehicles, using the data from the trajectory data.

Construction of Driving Data

Ten driving variables were selected from the NGSIM trajectory data, including the target operating acceleration, as shown in Table 10. The average target operating acceleration was derived from the mean of target accelerations of the accelerating process. This is because the average target acceleration of all processes represents only variation of speed during the entire observed travel period. For example, if a vehicle increased speed over the observed travel period, the average target acceleration of all processes probably has a positive value. As a result, the groups had to be separated into two categories: average target accelerations of the accelerating process and average target accelerations of the braking process. Because other processes have target operating acceleration of 0, they were not considered. An average target operating acceleration diagram was plotted, showing accelerating processes versus braking processes. Figure 29 shows the linear relationship between the average target operating accelerations of accelerating processes and braking processes. Knowing this relationship enabled the selection of one of the processes as a variables for classification. The average target operating accelerations of the accelerating process was selected because braking mainly depends on the movements of preceding vehicles.

Only 1,940 of the available 2,052 vehicles were selected because some vehicles did not have any preceding vehicle nor the associated data such as spacing, time headway, and relative speed. In addition, the study was only concerned with automobiles, so all vehicles whose data indicated that they were buses, trucks, or motorcycles were excluded.

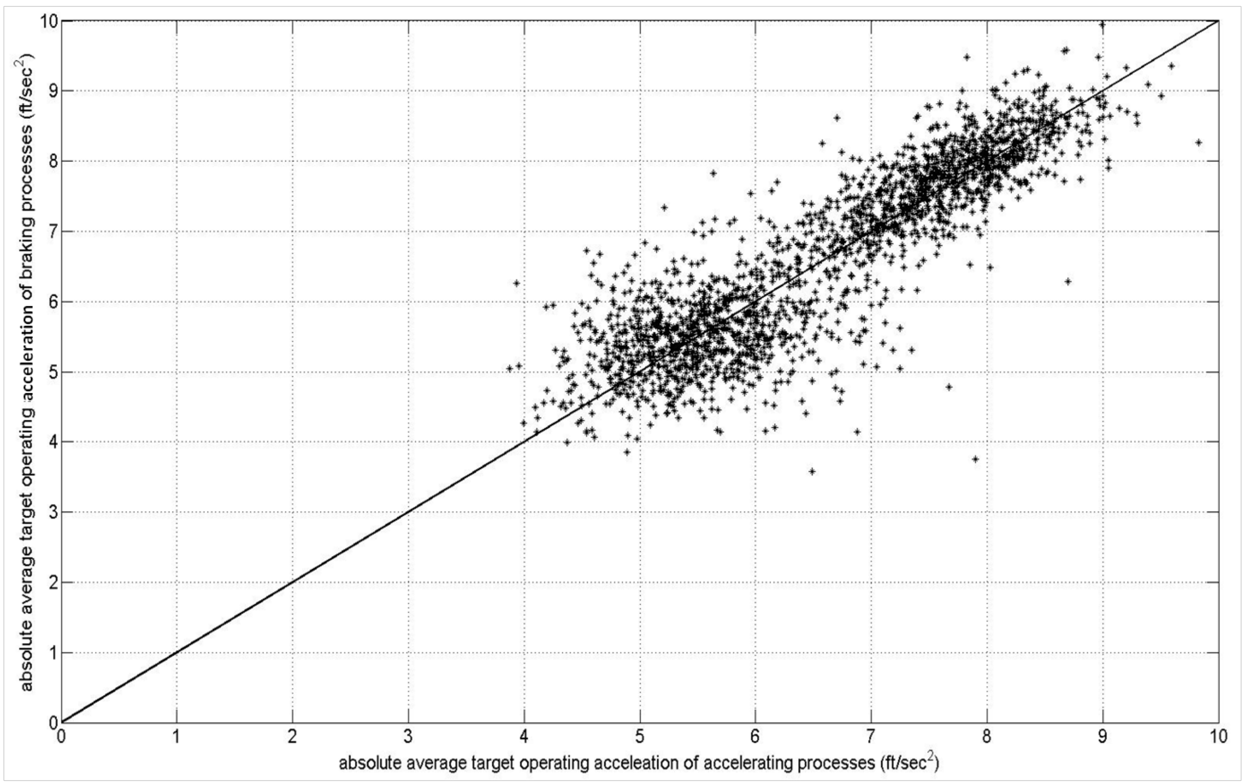


Figure 29. Absolute average target operating accelerations of accelerating process versus braking process.

Descriptive Statistics of Driving Data

Table 10 shows the results of descriptive statistics on 10 selected variables. Some variables such as acceleration and deceleration rates, were removed because the difference of the rates between individual vehicles was not significant. In Chapter 7, the average target operating acceleration will be used to classify drivers into three groups. All variables will be used for driving clustering and environmental clustering in Chapter 8.

Table 10. Selected variables of trajectory data for classification and their descriptive statistics.

Variables	min	max	median	mean	std	skewness	kurtosis	N
avg. speed in ft/s	13.16	78.56	24.26	29.16	13.63	1.73	5.24	1940
std. dev. of speed in ft/s	2.12	23.88	7.81	8.03	2.54	0.76	5.13	1940
avg. target operating acceleration in ft/s^2	3.87	9.83	6.68	6.61	1.22	0.01	1.87	1940
std. dev. of target operating acceleration in ft/s^2	2.95	5.00	4.04	4.04	0.26	0.01	3.71	1940
avg. spacing in ft	26.51	1256.14	62.52	75.58	54.64	8.75	144.21	1940
std. dev. of spacing in ft	3.68	284.43	18.42	22.53	17.07	5.13	60.54	1940
avg. time headway in s	0.88	2077.58	3.39	105.47	262.41	3.86	19.94	1940
std. dev. of time headway in s	0.05	4055.40	1.35	507.12	828.09	1.89	6.13	1940
avg. relative speed in ft/s	-17.82	24.72	0.08	0.12	2.02	1.45	34.22	1940
std. dev. of relative speed in ft/s	2.19	16.95	5.34	5.62	1.71	1.77	8.83	1940

Overview of Emissions and Fuel Consumption Models

State-of-the-art energy and emissions models are categorized as either macroscopic or microscopic (Rakha et al., 2003). Macroscopic models use average aggregate network parameters for estimation of network-wide emissions, and microscopic models estimate instantaneous fuel consumption and emissions aggregated to estimate network-wide measure of effectiveness (MOE).

Traditional Macroscopic Emissions Models

In North America, the Environmental Protection Angency's (EPA) MOBIL5 model and the California Air Resources Board's (CARB) EMFAC model have been mainly used (Rakha et al., 2003). Both models calculate activity-specific emissions rates using vehicle type, vehicle age, average speed, temperature, altitude, vehicle load, air conditioning usage, and vehicle operating mode. The vehicle activities include vehicle miles-traveled, number of trips, and vehicle-hours traveled, and are multiplied by the emission rates to estimate total emissions. MOBILE5 outputs estimates of three pollutants: hydrocarbons (HC), carbon monoxide (CO), and oxides of nitrogen (NO_x). EMFAC produces composite emission factors for these pollutants and particulate matter (PM). Both of models use average trip speeds as an input argument to select trip-specific emissions factors, and their factors are computed by testing vehicles through a limited number of driving cycles. The MOBIL5 model uses the Federal Test Procedure (FTP) cycle, commonly used for light-duty vehicle (LDV) testing, to derive baseline emission rates. This cycle is composed of three different phases: a cold start phase, a stabilized phase, and a hot start phase. The EMFAC uses the Unified Drive Cycle (LA92).

Microscopic Energy and Emissions Models

The microscopic energy and emissions models are derived from a relationship between dependent variables, such as fuel consumption and emission rates, and instantaneous measurements of explanatory variables, such as vehicle power, tractive effort, acceleration, and speed (Rakha et al., 2003). These models use second-by-second vehicle characteristics, traffic conditions, environmental conditions, and roadway condition as input arguments and estimate fuel consumption and emission rates. These models are sensitive to changes in vehicle acceleration behavior and can be utilized for the evaluation of operational-level transportation projects such as re-timing signals, modeling toll plazas, and modeling highway sections. The Comprehensive Modal Emissions Model (CMEM) is widely used and referenced in the literature review. The Virginia Tech microscopic energy and emissions model (VT-Micro) is an emerging model developed using instantaneous speed and acceleration levels as independent variables. This study includes two study cases using VT-Micro model (Chapter 7) and the CMEM model (Chapter 8).

Comprehensive Modal Emissions Model (CMEM)

The Comprehensive Modal Emission Model was developed to develop and verify a modal emissions model that accurately reflects light-duty vehicle (LDV) emissions. It was produced by the College of Engineering - Center for Environmental Research and Technology (CE-CERT) with researchers from the University of Michigan and Lawrence Berkeley National Laboratory at the University of California-Riverside (CMEM Users Guide). The CMEM is one of the power-demand-based emission models, and it predicts second-by-second tailpipe emissions and fuel consumption rates on the basis of vehicle and technology categories. The "comprehensive" part of the CMEM reflects the ability to predict emissions for wide variety of LDVs in various operating states: properly functioning, deteriorated, malfunctioning, and so on. The CMEM model was developed using second-by-second data of engine-out and tailpipe emissions of over 300 vehicles, including more than 30 tested high emitters, using three drive cycles: FTP, US06, and MEC. The CMEM has been enhanced every year. CMEM version 3.01 includes a Heavy-Duty Diesel (HDD) Modal Emissions Model for a heavy-duty diesel emissions and fuel consumption model. In late 2003, the CMEM was enhanced using new testing and calibration methodologies more compatible with the latest Portable Emissions Measurement System (PEMS) data set. In addition, the new methodology for determining categories for future fleet compositions was developed to use the modal emissions model for estimating an inventory for a vehicle fleet. The preliminary ammonia (NH_3) module was included in the CMEM based on a parallel vehicle emissions test and research program. In 2004, the CMEM was modified to include three new light-duty vehicle categories for extremely low-emitting vehicles with new technology 98% to 99% cleaner than catalyst-equipped vehicles produced in the 1980s, which were the basis of the California Air Resources Board's certification standards: Low Emitting Vehicle (LEV), Ultra Low Emitting Vehicle (ULEV), Super Ultra Low Emitting Vehicle (SULEV). In 2005, development of a particulate matter (PM) module for the CMEM was begun, focusing on second-by-second PM emissions measured from vehicles, including heavy-duty trucks.

To collect second-by-second emissions data for the CMEM, about 340 vehicles were tested. The vehicle and technology categories of the CMEM were chosen based on a vehicle's emission contribution, and the categories are divided into several subgroups on the basis of power-to-weight ratio and mileage. There are currently 36 CMEM vehicle and technology categories: 15 categories for normal emitting cars and low emitting cars, 8 categories for normal emitting light-duty diesel or gasoline-powered trucks, 8 categories for normal emitting heavy-duty diesel trucks, and 5 categories for high emitting light-duty vehicles.

Environmental Data

For this study, the NGSIM trajectory data was smoothed because the CMEM estimates emissions and fuel consumption based on second-by-second profile data of speed and acceleration. Second-by-second speeds were selected from original trajectory data based on deci-seconds, then four input files were made. The CMEM automatically computes a profile of acceleration based on the input profile of speed. Because the CMEM estimates emissions and fuel consumption data for each vehicle in grams per mile, it was possible to obtain the total amount of each emission and fuel consumption. However, only the unit amount of emissions and fuel consumption were used because the travel distances did not vary significantly. Because this study was concerned only with driving behavior, it was assumed that all vehicles were of the same type: category 11, tier 1, with mileage less than 50,000 miles, and with high power-to-weight ratio (power/weight).

Descriptive Statistics of Environmental Data

Table 11 shows descriptive statistics of emissions and fuel consumption. The distribution of CO_2 and fuel consumption was similar to normal distribution, but other distributions were not.

Table 11. Descriptive statistics of variables of emissions and fuel consumption.

grams/mile	min	max	median	mean	std	skewness	kurtosis	N
Hydrocarbons (HC)	0.01	1.50	0.04	0.15	0.21	2.22	8.73	1940
Carbon monoxide (CO)	0.19	179.21	1.85	16.18	24.59	2.18	8.66	1940
Oxides of nitrogen (NO_X)	0.15	1.22	0.32	0.36	0.13	1.83	7.86	1940
Carbon dioxide (CO_2)	231.18	787.79	487.42	494.79	89.22	0.35	3.08	1940
Fuel consumption	78.60	311.50	161.20	164.14	30.47	0.58	3.81	1940

Factor Analysis of Driving Data

To define factors, the study started from well-known driving variables such as time headway and spacing. Both of these variables are related to the preceding vehicle. Time headway, however, is also related to the vehicle's own speed, but spacing is not. It was assumed that the time headway was influenced by the desired speed, and drivers decrease the time headway when they want to go faster than the preceding vehicle. On the other hand, it was assumed that the spacing was perceived by drivers in terms of relative location of the preceding vehicle, and drivers had certain desired values of spacing. In Table 12, factor 1 is mainly related to average value and standard deviation of spacing, and factor 2 is related to time headway and average speed. Factor 2 might be related to speed and flow at the macroscopic level. Thus, factor 1 is the microscopic factor and factor 2 is the macroscopic factor. Note that loadings for factor 1 of average time headway and standard deviation of time headway are positive. This means that the increase of factor 1 increases the average time headway and the standard deviation of time headway. That is, vehicles that had higher values of factor 1 used longer time headway. Similarly, vehicles that had higher values of factor 2 used longer spacing. However, the standard deviation of spacing and time headway is used to measure the stability of spacing and time headway , and factor 1 and factor 2 decrease the stability of spacing and time-headway, respectively.

Table 12. Results of factor analysis of driving data.

Variables	Factor 1	Factor 2	Specific value
avg. speed	-0.2442	**0.6265**	0.4999
std. dev. of speed	**0.3971**	-0.0585	0.8316
avg. target operating acceleration	**0.4456**	**0.1096**	0.8047
std. dev. of target operating acceleration	**-0.1002**	**0.1331**	0.9681
avg. spacing	-0.0085	**0.8780**	0.2267
std. dev. of spacing	0.0688	**0.4999**	0.7561
avg. time headway	**0.9132**	-0.0043	0.1648
std. dev. of time headway	**0.9869**	-0.0575	0.0050
avg. relative speed	-0.0045	**0.1705**	0.9707
std. relative speed	0.1898	**0.4696**	0.7713

Factor Analysis of Environmental Data

Factor 1 was primarily related to HC, CO, and NO_x, and factor 2 was primarily related to CO_2 and fuel consumption. Because the vehicles in this study were traveling on the same road at the same time, aggressive driving could cause incomplete combustion inside the engine and consume more gasoline. Factor 1 was defined as incomplete combustion, and factor 2 as fuel consumption. In seen in Table 13, factor 1 mainly influenced the emitted amount of hydrocarbons, carbon monoxide, and oxides of nitrogen. This theoretically and practically agrees with existing knowledge that these gases increase when fuel is not burned completely. Factor 2 influenced only the amount of emitted carbon dioxide and fuel consumption, and this completely agrees with the knowledge that carbon dioxide is a product of complete combustion.

Table 13. Results of factor analysis of environmental data.

Variables	Factor 1	Factor 2	Specific value
Hydrocarbons (HC)	**1.0232**	-0.1078	0.0050
Carbon monoxide (CO)	**1.0219**	-0.1015	0.0050
Oxides of nitrogen (NO_x)	**0.8903**	0.2241	0.0459
Carbon dioxide (CO_2)	-0.2175	**1.0372**	0.0050
Fuel consumption	0.2168	**0.9164**	0.0050

Driving Clustering

Objects with various numbers of clusters were clustered into cases from 2 clusters to 10 clusters, and then the average silhouette values of each case were compared. As a result, three clusters were obtained as optimal solutions for k-means clustering, as shown in Figure 30.

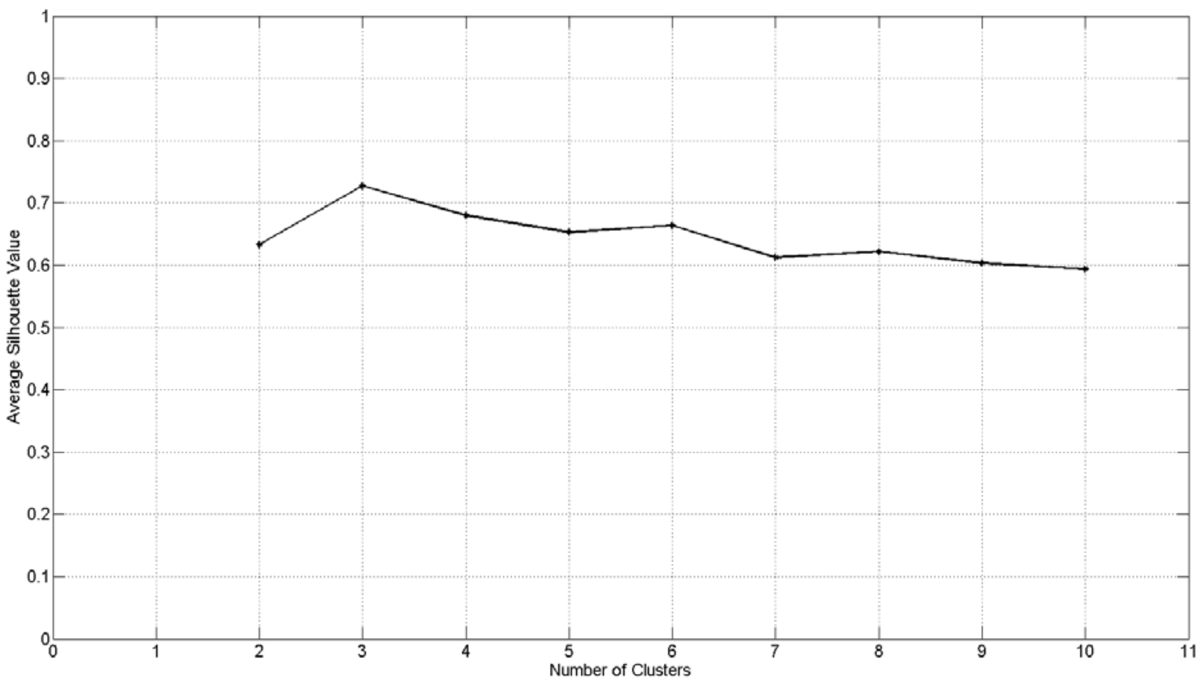


Figure 30. Comparison of average silhouette value corresponding to number of clusters for driving clustering.

A result of driving clustering with three clusters was obtained. In Figure 31, the first cluster included 195 vehicles, had 96 incorrectly assigned vehicles, and had a silhouette value of 0.026. These vehicles had a higher tailgating factor and can be classified as the speed-based drivers group. The second cluster included 294 vehicles and had only 24 vehicles with negative silhouette values. These vehicles have a high spacing factor and can be classified as the spacing-based drivers group. The third cluster included 1,451 vehicles, had no incorrectly assigned vehicles, and had lower values of both factors. They were classified as the moderate drivers group and constituted 1,451 of 1,940 total drivers.

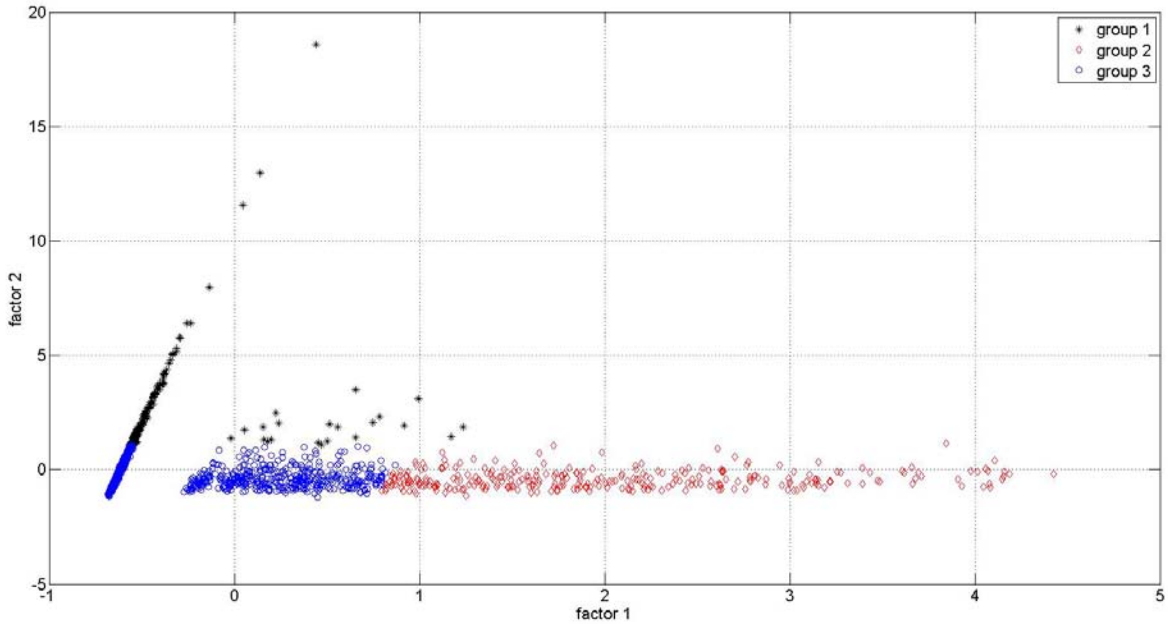


Figure 31. Results of driving clustering.

In Figure 31, vehicles in cluster 1 (black stars) have higher macroscopic factor values but lower microscopic factor values than the vehicles in the other clusters. The macroscopic factor is the primary factor of average speed and two variables of time headway, and the time headway is related to speed. Thus, cluster 1 was designated as the speed-based drivers group. Similarly, vehicles in cluster 2 (red diamonds) have higher microscopic factor values, a primary factor of two variables of spacing, and their cluster was designated as the spacing-based drivers group. Because cluster 3 includes most of the vehicles, it was designated as the moderate drivers' group. It was expected that both the speed-based and spacing-based drivers groups could be high emitters and high fuel consumers due to their driving behaviors.

Environmental Clustering

Similar to the driving clustering, two clusters were obtained as optimal solutions for clustering, as shown in Figure 32.

The first cluster of 560 vehicles had relatively higher values of factor 2 and had 193 incorrectly assigned vehicles. The second cluster had 1,492 vehicles, a relatively lower value of factor 2, and no incorrectly assigned vehicles. Certain behaviors cause high fuel consumption and incomplete combustion, and the result of the classification shows relative separation of two groups (see Figure 33). Cluster 1 was defined as the as high emitters group, and cluster 2 was defined as the moderate emitters group. Figure 33 shows that many vehicles were classified in the high emitters group due to the fuel consumption factor, and a small number of vehicles were classified in the high emitters group due to the incomplete combustion factor.

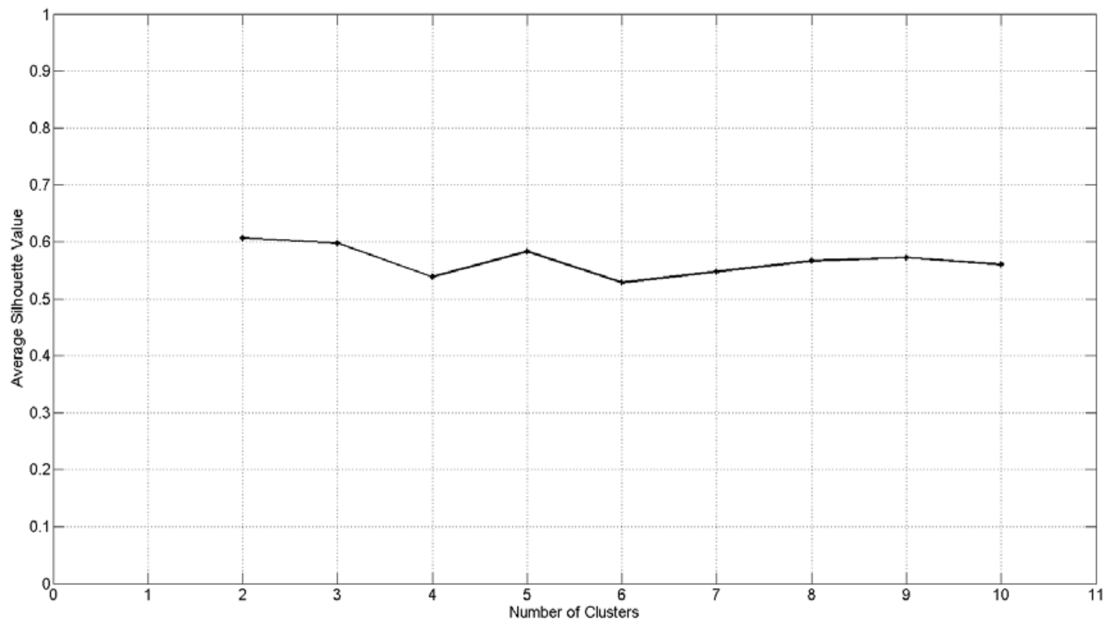


Figure 32. Comparative analysis of silhouette values corresponding to the number of clusters for environmental clustering.

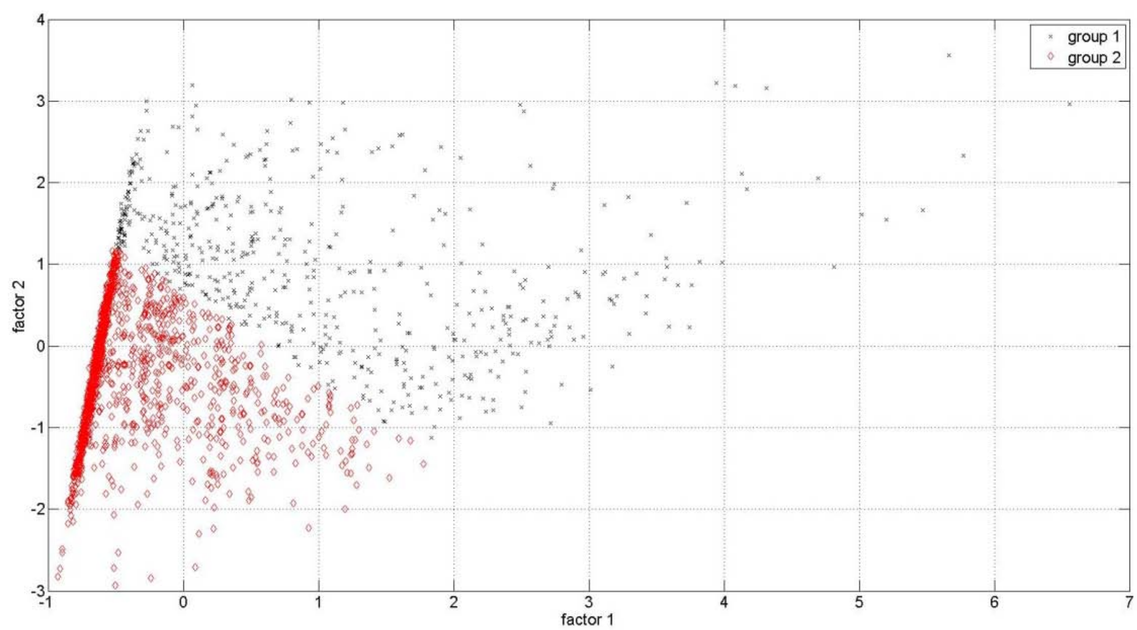


Figure 33. Results of environmental clustering.

Comparative Analysis

The results of each method were compared using two comparative analyses. The first comparative analysis compared the amount of emissions and fuel consumption corresponding to driving clustering and environmental clustering.

In the results for driving clustering, the speed-based drivers group, assigned to cluster 1, emitted a relatively large total amount of hydrocarbon, carbon monoxide, and oxides of nitrogen. The spacing-based drivers group, assigned to cluster 2, emitted a relatively larger amount of carbon dioxide and consumed more fuel. In the results for environmental clustering, the high emitters group (cluster 1) emitted significantly larger amount of hydrocarbons and carbon monoxide and emitted slightly larger amount of oxides of nitrogen and carbon dioxide. They also consumed slightly more fuel. See Table 14.

Table 14. Comparative analysis of emissions and fuel consumption for clustered vehicles.

Based data	Clusters	Number of vehicles		HC		CO		NOx		CO_2		Fuel	
		N	%	grams	%	grams	%	grams	%	grams	%	grams	%
Driving data	1	195	10.05	29.89	34.12	3,394.24	35.75	32.17	15.39	25,301.91	8.84	9,689.15	10.20
	2	294	15.15	8.21	9.37	850.38	8.96	32.99	15.79	52,792.08	18.45	17,071.44	17.97
	3	1,451	74.79	49.49	56.50	5,249.33	55.29	143.81	68.82	208,096.07	72.71	68,250.52	71.83
Emissions & fuel consumption	1	538	27.73	56.96	65.03	6,508.16	68.55	83.22	39.82	90,808.00	31.73	31,909.81	33.59
	2	1,402	72.27	30.62	34.97	2,985.80	31.45	125.75	60.18	195,382.07	68.27	63,101.30	66.41
Total vehicles		1,940	100	87.58	100	9,493.96	100	208.97	100	286,190.07	100	95,011.11	100

Another comparative analysis compared the number of clustered vehicles in the driving and the environmental clusterings based on the assumption that drivers in either the speed-based or the spacing-based drivers groups might be high emitters. As a result, 61.19% of vehicles were assigned to the moderate drivers' group and moderate emitters' group. Also, 13.61% of vehicles were assigned to the moderate drivers' group, but the high emitters group. Also, 8.61% of vehicles were assigned to the spacing-based drivers group and the high emitters group. Also, 5.52% of vehicles were assigned to the speed-based drivers group and the high emitters group. It was expected that vehicles assigned to the speed-based or spacing-based drivers group would also be assigned to the high emitters group, but 11.09% of the speed-based or spacing-based vehicles were not, as shown in Table 15.

Table 15. Comparative analysis of the number of clustered vehicles to each cluster.

		Driving clustering			Total
		speed-based drivers	spacing-based drivers	moderate drivers	
Environmental clustering	high emitters	5.52%	8.61%	13.61%	27.73%
	moderate emitters	4.54%	6.55%	61.19%	72.27%
Total		10.05%	15.15%	74.79%	100.00%

Consequently, about 25% of vehicles would be environmentally misclassified if they were categorized based on driving patterns, and about 75% would be assigned correctly. In particular, the vehicles in the speed-based drivers group of the driving clustering can be considered significant high emitters of hydrocarbons and carbon monoxide and light emitters of oxides of nitrogen.

Figure 34 shows the comparison between driving clustering and environmental clustering. Clustered drivers were mapped based on driving clustering in the environmental clustering domain. Figure 34(a) shows the result of environmental clustering in the environmental clustering domain, and Figure 34(b) shows the result of driving clustering in the environmental clustering domain. The figures show a relationship between driving clustering and environmental clustering. Each cluster's drivers, based on driving clustering, were mapped onto a certain area of the environmental clustering domain. The spacing-based drivers were mapped onto area with a high fuel consumption factor and a low incomplete combustion factor. Speed-based drivers had a relatively higher incomplete combustion factor than others who had the same fuel consumption factor. Moderate drivers had low values of both factors.

Many vehicles assigned to the speed-based drivers group were also assigned to the high emitters group due to a high incomplete combustion factor and a high fuel consumption factor. Many vehicles assigned to the speed-based drivers group were also assigned to the moderate emitters group due to low fuel consumption factors. That is, speed-based drivers tend to produce more emissions and consume more fuel than other drivers. Similarly, spacing-based drivers tend to produce less emissions but to consume more fuel than other drivers.

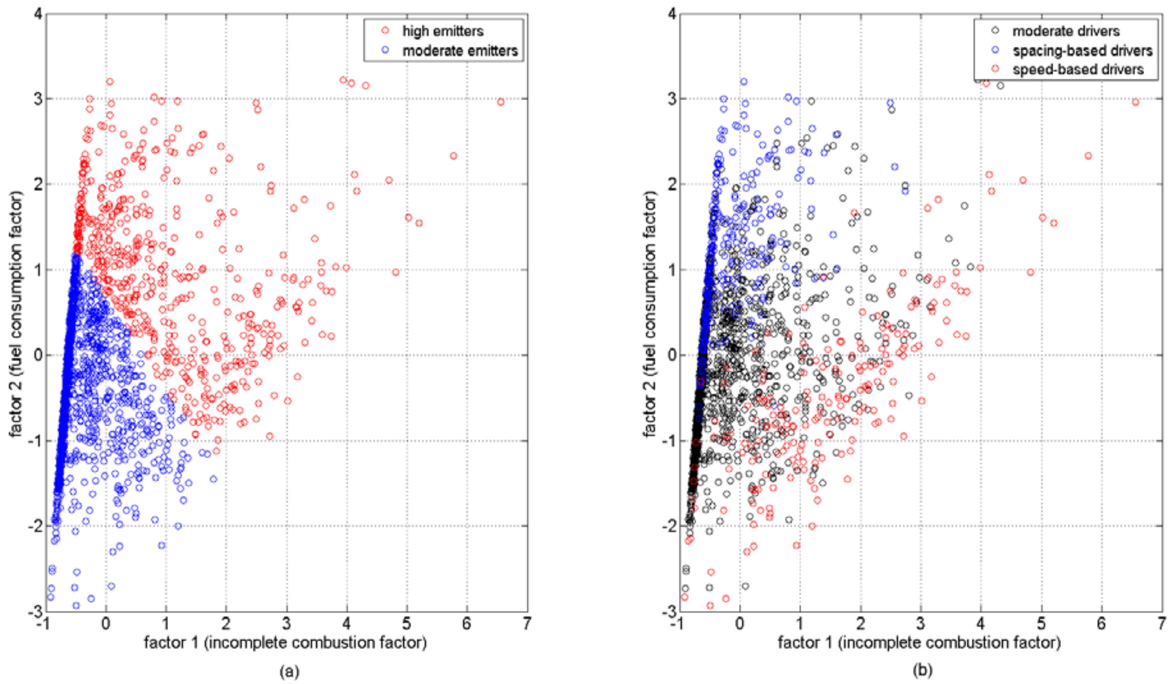


Figure 34. Comparative analysis between (a) clustered vehicles by environmental clustering and (b)) clustered vehicles by driving clustering on the environmental clustering domain.

Conclusion

This study determined two types of sets of variables for classification from NGSIM trajectory data. The first set consists of 10 driving variables such as average speed, standard deviation of speed, average target operating acceleration, standard deviation of target operating acceleration, average spacing, standard deviation of spacing, average time-headway, standard deviation of time-headway, average relative speed, and standard deviation of relative speed. The second set, estimated by the CMEM, consists of 5 environmental variables: fuel consumption and emitted hydrocarbons, carbon monoxide, oxides of nitrogen, and carbon dioxide. Factor analysis was used to reduce the number of variables in each set, and then the k-mean clustering algorithm was used to classify data. For the driving data, two factors were defined through the factor analysis: a spacing factor and a tailgating factor. Vehicles were classified into three clusters: the moderate drivers group, speed-based drivers group, and spacing-based drivers group. The moderate drivers group had lower value of factors 1 and 2 than the other groups, the spacing-based group had higher values of factor 1, and the speed-based group had higher values of factor 2. As a result, 195, 294, and 1,451 vehicles were assigned to clusters 1, 2, and 3, respectively.

The same analysis was then repeated using the environmental data calculated by the CMEM using the NGSIM trajectory data. Two factors were defined: the incomplete combustion factor and fuel consumption factor. Vehicles were classified into two clusters: the high emitters group and moderate emitters group. As a result, 538 and 1,402 vehicles were assigned to clusters 1 and 2, respectively. Finally, the results of each classification were compared to determine the difference between two methods.

About 75% of vehicles were similarly classified based on both driving and environmental data. In addition, the spacing-based drivers tended to tended to produce less emissions but consumed more fuel than other drivers. Similarly, the speed-based drivers tended to produce a large amount of emissions compared to other drivers who consumed a similar amount of fuel. Most moderate drivers consumed less fuel and produced less emissions.

The results support the conclusion that driving clustering is not enough to distinguish high emitters based on driving behavior. However, the study did find a relationship between driving clustering and environmental clustering. Future research should consider more driving variables to improve the driving clustering method. In addition, the use of higher number of factors could be investigated. Other hierarchical clustering algorithms, Gaussian Mixture Models, and the fuzzy c-means clustering algorithm could be considered as well.

Chapter 7. Conclusions

Summary of the Research

This study used the NGSIM trajectory data to conduct a microscopic analysis of driving behavior. It was expected that the results would show a significant difference between driving behavior of individual drivers, and that the driving behavior would greatly influence the amount of emissions and fuel consumption. This research aimed to classify drivers based on driving data and environmental data and to find a relationship between driving behaviors of drivers clustered by driving clustering and those clustered by environmental clustering. A new measure representing pedal-pushing behavior was proposed and designated as target operating acceleration of the five processes.

Study case 1 classified drivers into three groups based on pedal-pushing behavior: defensive, moderate, and aggressive drivers. Aggressive drivers were defined as those with a higher average target operating acceleration than other drivers, and defensive drivers were defined as those with a lower average target operating acceleration. The thresholds were set according to standard deviation of average target operating acceleration because the differences between driving behavior declined in inverse proportion to the congestion congested level.

Only the average target operating acceleration was used as a feature of each group. As a result, it was determined that aggressive drivers emitted more hydrocarbons (HC), carbon monoxide (CO), and oxides of nitrogen (NO_x). However, they consumed less fuel and emitted less carbon dioxide (CO_2) than other drivers. In addition, additional classifications were created with a greater number of clusters. This proposed approach determined that the heavily aggressive drivers tended to produce most emissions under the congestion condition.

Study case 2 dealt with two types of data: driving data and environmental data. The driving data included 10 variables, and the environmental data included 5 variables estimated by the CMEM. Factor analysis was conducted using two sets of variables, and then two factors were determined for each data set. A microscopic factor and a macroscopic factor were determined for driving data, and an incomplete combustion factor and a fuel consumption factor were determined for environmental data. The microscopic and macroscopic factors were used for driving clustering, and other factors were used for environmental clustering. Drivers were classified into three driving-based groups: speed-based, spacing-based, and moderate. Drivers were also classified into two environmental groups: high emitters and moderate emitters. Finally, driving clustering and environmental clustering were compared. The results showed that about 75% of drivers were similarly classified based on both driving and environmental data, but about 25% drivers were differently classified. It was also found that spacing-based drivers tended to produce less emissions than other drivers groups but consumed more fuel. The speed-based drivers produced relatively more emissions, but their fuel consumption was not known. Most moderate drivers consumed less fuel and produced less emissions.

Conclusions

The average target acceleration proposed in this study represents individual accelerating behavior. The classification using the new variable classified drivers into high emitters' group of emissions including hydrocarbon, oxides of nitrogen, and carbon monoxide, but the proposed approach did not classified drivers into high emitters for fuel consumption. The performance of the proposed approach did decrease under the congestion conditions because the variations in driving behavior decreased due to reduction of time-headway or spacing between vehicles. This phenomenon suggests that the potential to change driving behavior, as in eco-driving, is restricted by traffic conditions, and mitigation of traffic congestion could still be a primary solution for environmental and economical issues in transportation systems.

On the other hand, the evaluated environmental impact of driving behavior changes was significant, and the study verified that a small number of vehicles exhibiting certain driving behaviors has greatly contributed to the total amount of emissions. Even though the environmental impact of driving behavior changes is limited under congestion conditions, the results of this study show that we should still manage individual driving behavior.

This study found several significant factors by using factor analysis, and these were used for driving clustering and environmental clustering. This approach verified a correlation between driving behavior and environmental activities, and it raised the possibility that environmental contributions could be specified based on driving behavior.

In addition, both the moderate drivers group and the moderate emitters group exhibited the best overall driving behavior in both the driving clustering and environmental clustering. The driving behavior of these moderate groups could be similar to eco-driving because they produced the least emissions and used the least fuel of out of all the drivers groups. Therefore, it is recommend that all drivers should change their driving behavior to the driving behavior of the moderate drivers.

Future Work

This study used only five variables: speed, time-headway, spacing, relative speed, and a proposed average target operating acceleration instead of acceleration. All variables are related to driving behaviors because the NGSIM data includes only trajectory data and loop detector data. The upcoming IntelliDrive data would include other data such as weather data. In the future, the IntelliDrive data could be used to find more variables related to other factors in environmental impact or eco-driving.

This study used only NGSIM trajectory data, but the upcoming IntelliDrive data could be used to find more factors for driving data. The proposed approach should then be conducted once more using the new factors from the IntelliDrive data.

Today, eco-driving is given attention around the world. But because eco-driving involves various objectives, the definition and rules for eco-driving are still not clear. In the future, the proposed approach could be used to investigate methods of evaluating eco-driving.

References

1. Abdi, Herve. "Factor Rotations in Factor Analysis." *Encyclopedia of Social Sciences Research Methods*, Thousand Oaks, CA, 2003, pp. 1-8.
2. Ahn K., Rakha H., and Trani A. "VT-Micro framework for Modeling of High-Emitting Vehicles." *Transportation Research Record: Journal of the Transportation Research Board*, No. 1880, 2004, pp. 39-49.
3. Ando, Ryosuke, and Nishihori, Yasuhide. "How Does Driving Behavior Change When Following an Eco-Driving Car?" *Procedia Social and Behavioral Sciences*, Vol. 20, 2011, pp. 577-587.
4. Barth, Matthew, and Boriboonsomsin, Kanok. "Real-World Carbon Dioxide Impacts of Traffic Congestion." *Transportation Research Record: Journal of the Transportation Research Board*, No. 2058, Transportation Research Board of the National Academies, Washington, D.C. 2008, pp 163-171.
5. Boer, Erwin R., 1999, "Car following from the driver's perspective," *Transportation Research Part F, Vol. 2*, No. 4,201-206
6. Brockfeld, E., R. D. Kühne, and P. Wagner. "Calibration and Validation of Microscopic Models of Traffic Flow." *Transportation Research Record: Journal of the Transportation Research Board*, No. 1934, Transportation Research Board of the National Academies, Washington, D.C., 2005, pp. 179–187.
7. Cambridge Systematics, Inc. "NGSIM I-80 Data Analysis (4:00 p.m. to 4:15 p.m.)" *Summary Report*. Federal Highway Administration. 2005a. http://ngsim-community.org/. Accessed Jan. 21, 2012.
8. Cambridge Systematics, Inc. "NGSIM I-80 Data Analysis (5:00 p.m. to 5:15 p.m.)" *Summary Report*. Federal Highway Administration. 2005b. http://ngsim-community.org/. Accessed Jan. 21, 2012.
9. Cambridge Systematics, Inc. "NGSIM I-80 Data Analysis (5:15 p.m. to 5:30 p.m.)" *Summary Report*. Federal Highway Administration. 2005c. http://ngsim-community.org/. Accessed Jan. 21, 2012.

10. Casucci, M., M. Marchitto, and P. C. Cacciabue. "A Numerical Tool for Reproducing Driver Behaviour: Experiments and Predictive Simulations." *Applied Ergonomics*, Vol. 41, No. 2, 2010, pp. 198–210.
11. *Comprehensive Model Emission Model (CMEM) Users Guide version 3.01*. CMEM, University of California, Riverside Center for Environmental Research and Technology, 2006, pp. 1-56.
12. Cooper, C. David, and Alley, F.C. *Air Pollution Control: A Design Approach*. Waveland Press, Inc. 1996, pp. 515-549.
13. De Coster, J. *Overview of Factor Analysis*.1998. www.stat-help.com/notes.html Retrieved June, 30, 2012.
14. De Nervers, Noel. *Air Pollution Control Engineering, Second Edition*. Waveland Press, Inc. 2010, pp. 471-579.
15. ECC. "Communication from the Commission to the European Parliament and the Council." *European Commission Communication (ECC), COM (2009)* 713, 2009, pp.1-15
16. ECMT/IEA. "Making Cars More Fuel Efficient; Technology for Real Improvement on the Road." *OECD*, Paris, 2005, pp. 1-82.
17. El-Shawarby, Ihab, Ahn, Kyoungho, and Rakha, Hesham. "Comparative field evaluation of vehicle cruise speed and acceleration level impacts on hot stabilized emissions." *Transportation Research Part D*. Vol. 10, 2005, pp.13-30.
18. Everitt, Brian S., and Dunn, Graham. *Applied Multivariate Data Analysis*, *Second Edition*. Oxford University Press Inc., New York, 2001.
19. Fiat. *Eco-driving Uncovered: The benefits and challenges of eco-driving, based on the first study using real journey data*. Fiat, 2010, pp. 1-43. http://www.fiat.co.uk/uploadedFiles/Fiatcouk/Stand_Alone_Sites/EcoDrive2010/ECO-DRIVING_UNCOVERED_full_report_2010_UK.pdf / . Accessed Jun. 10, 2012.
20. FHWA. *Federal Highway Administration. Next Generation Simulation Program*. http://ngsim-community.org/ Accessed Jan.10, 2012.
21. Fulton, L., and Eads, G. *IEA/SMP Model Documentation and Reference Projection*. WBSCD, 2004.

22. Greenwood, I. D., Dunn, R. C., and Raine, R.R. “Estimating the effects of Traffic Congestion on Fuel Consumption and Vehicle Emissions Based on Acceleration Noise.” *Journal of Transportation Engineering*, ASCE, 2007, pp.96-104.
23. GTZ. “EcoDriving.” *Deutsche Gesellschaft fur Technische Zusammenarbeit (GTZ) GmbH*, 2005. Accessed Jul.13, 2012.
24. Hamdar, S. H., M. Treiber, H. S. Mahmassani, and A. Kesting. “Modeling Driver Behavior as Sequential Risk-Taking Task.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2088, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 208–217.
25. IEA. *Saving Oil and Reducing CO2 Emissions in Transport*. International Energy Agency, OECD, 2001, pp. 194.
26. IEA. *World Energy Outlook 2006*. International Energy Agency (IEA), Paris, 2006a, pp. 596.
27. IEA. *CO2 Emissions from Fuel Combustion 1971-2004. International Energy Agency (IEA)*, Paris, 2006b, pp. 596.
28. Johnson, Richard A., and Wichern, Dean W. *Applied Multivariate Statistical Analysis, Sixth Edition*. Pearson Education Inc., 2007.
29. Kamal, M. A. S., Mukai, M., Murata, J., and Kawabe, T. “On Board Eco-Driving System for Varying Road-Traffic Environments Using Model Predictive Control.” *2010 IEEE International Conference on Control Applications Part of 2010 IEEE Multi-Conference on System and Control*, Yokohama, Japan, 2010, pp.1636-1641.
30. Kim, Jiwon, and Mahmassani, Hani S. “Correlated Parameters in Driving Behavior Models.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2249, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 62–77.
31. Lee, Chanyoung, Saxena, Meeta, and Lin, Pei-Sung. “Aggressive Driving and Safety Campaigns: Lessons Learned from Better Driver Campaign.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2182, Transportation Research Board of the National Academies, Washington, D.C. 2010, pp 79-87.
32. Lownes, Nicholas E., and Machemehl, Randy B. “Sensitivity of Simulated Capacity to Modification of VISSIM Driver Behavior Parameters.” *Transportation Research Record:*

Journal of the Transportation Research Board, No. 1988, Transportation Research Board of the National Academies, Washington, D.C., 2006, pp. 102–110.

33. Ma, Xiaoliang, and Andréasson, Ingmar. "Statistical Analysis of Driver Behavior Data in Different Regimes of the Car-Following Stage." *Transportation Research Record: Journal of the Transportation Research Board,* No. 2018, Transportation Research Board of the National Academies, Washington, D.C., 2007, pp. 87–96.
34. Magana, V. Corcoba, and Organero, M. Munoz. "Artemisa: Using an Android Device as an Eco-Driving Assistant." C*yber Journals: Multidisciplinary Journals in Science and Technology*, *JMTC*, 2011, pp. 1-8.
35. Mensing, Felicitas, Trigui, Rochdi, and Bideaux, Eric. "Vehicle Trajectory Optimization for Application in Eco-Driving." *Vehicle Power and Propulsion Conference (VPPC)*, IEEE, 2011, pp. 1-6.
36. May, Adolf D. *Traffic Flow Fundamentals*. Prentice-Hall, Inc., 1990.
37. Mehmood, Arif, Saccomanno, Frank, and Hellinga, Bruce. "Application of System Dynamics in Car-Following Models." *Journal of Transportation Engineering*, Vol. 129, No. 6, 2003, pp. 625-634.
38. Metz, B. , Dacidson, O. R., Bosch, P. R., Dave, R., and Meyer, L.A. "Climate Change 2007: Working Group III: Mitigation of Climate Change." *Intergovernmental Panel on Climate Change*, Chapter 5, 2007.
39. Mizell, L. *Road Range: Causes and Dangers of Aggressive Driving: Hearing before the Subcommittee on Surface Transportation of the Committee on Transportation and Infrastructure*. House of Representatives, 105th Congress, 1st Session, 1997.
40. Miles, Donald E., and Gregory L. Johnson. "Aggressive Driving Behaviors: are There Psychological and Attitudinal Predictors?" *Transportation Research Part F*, Vol. 6, 2003, pp. 147-161.
41. Murphey, Yi Lu, Milton, Robert, and Kiliaris, Leonidas. "Driver's Style Classification Using Jerk Analysis." *IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems 2009 (CIVVS'09)*, 2009, pp.2-28.
42. National Highway Traffic Safety Administration. *Strategies for Aggressive Driver Enforcement*.

http://www.nhtsa.gov/people/injury/enforce/aggressdrivers/aggenforce/index.html Accessed Aug. 2, 2012.

43. Olson, Paul L., Dewar, Robert, and Farber, Eugene. *Forensic Aspects of Driver Perception and Response, Third Edition*, Lawyers & Judges Publishing Co., Inc., 2010.
44. Panwai, Sakda, and Dia Hussein. "Comparative Evaluation of Microscopic Car-Following Behavior." *IEEE Transaction on Intelligent Transportation Systems*, Vol. 6, No. 3, 2005, pp. 314-325.
45. Pataki, G. "Governor Pataki Announces Legislation to Fight Road Rage." *New York State Government Press Release*, 1998. www.state.ny.us/governor/press/feb1998.html Accessed July 23, 1999.
46. Pueboobpaphan, R., and B. van Arem. "Driver and Vehicle Characteristics and Platoon and Traffic Flow Stability: Understanding the Relationship for Design and Assessment of Cooperative Adaptive Cruise Control." *Transportation Research Record: Journal of the Transportation Research Board*, No. 2189, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 88–97.
47. Punzo,Vincenzo, Borzacchiello, Maria Teresa, and Ciuffo, Biagio. "Estimation of Vehicle Trajectories from Observed Discrete Positions and Next-Generation Simulation Program (NGSIM) Data." *88th TRB 2009 Annual Meeting CD-ROM*.
48. Rakha, Hesham; Ahn, Kyoungho; Trani, Antonio. "Comparison of MOBILE5a, MOBILE6, VT-MICRO, and CMEM Models for Estimating Hot-Stabilized Light-Duty Gasoline Vehicle Emissions." *Canadian Journal of Civil Engineering*, Vol. 30, Issue 6, 2003, pp.1010-1021.
49. Rakha H. and Ahn K. INTEGRATION Modeling Framework for Estimating Mobile Source Emissions. Journal of Transportation Engineering , Vol. 130, No. 2, 2004, pp. 183-193.
50. Rakha, Hesham A., Ahn, Kyoungho, Moran, Kevin, Saerens, Bart, and Bulck, Eric Van den. "Virginia Tech Comprehensive Power-Based Fuel Consumption Model: Model Development and Testing." *Transportation Research Part D*, Vol. 16, 2011, pp. 492-503.
51. Rayner, J.N. *An Introduction to Spectral Analysis*. London, Pion. 1971.
52. Rigolli, Marco, Williams , Quentin, Gooding , Mark J., and Brady, Michael. "Driver Behavioural Classification from Trajectory Data." *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, 2005, pp. 889- 894.

53. Rong, Jian, Mao, Kejun, and Ma, Jianming. “Effects of Individual Differences on Driving Behavior and Traffic Flow Characteristics.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2248, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 1–9.
54. Saboohi, Y., and Farzaneh, H. “Model for Developing an Eco-Driving Strategy of a Passenger Vehicle Based on the Least Fuel Consumption.” *Applied Energy*, Vol. 86, 2009, pp. 1925-1932.
55. Sadabadi, Kaveh F., and Haghani, Ali. “Real-Time Solution of Velocity-Based First-Order Continuum Traffic Model with Finite Element Method.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2260, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 24–41.
56. Scora, George, and Barth, Matthew. *Comprehensive Model Emission Model (CMEM) Users Guide version 3.01*. University of California, Riverside Center for Environmental Research and Technology, 2006, pp. 1-56.
57. Sivak, Michael, and Schoettle, Brandon. *Eco-Driving: Strategic, Tactical, and Operational Decisions of The Driver That Improve Vehicle Fuel Economy*. The University of Michigan Transportation Research Institute, 2011, pp.1-19.
58. Sultan, B., Brackstone, M., and McDonald, M. “Drivers’ Use of Deceleration and Acceleration Information in Car-Following Process.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 1883, TRB, National Research Council, Washington, D.C., 2004, pp. 31–39.
59. Talebpour, Alireza, Mahmassani, Hani S., and Hamdar, Samer H. “Multiregime Sequential Risk-Taking Model of Car-Following Behavior: Specification, Calibration, and Sensitivity Analysis.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2260, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 60–66.
60. The IntelliDrive Michigan Testbed. *Documentation of On-Board Equipment Files from the IntelliDrive Michigan Testbed*, 2005, http://datacapture.noblis.org/ Accessed Oct. 15, 2011.
61. Theodoridis, Segios, and Koutroumbas, Konstantinos. *Pattern Recognition*. Elsevier Inc., 2009.

62. Thiemann, Christian, Treiber, Martin, and Kesting, Arne. “Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2088, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 90–101.
63. Vanlaar, Ward, Simpson, Herb, and Robertson, Robyn. “Aggressive Driving: A Survey of Attitudes, Opinions and Behaviors.” *Journal of Safety Research*, Vol. 39. 2008, pp. 375–381.
64. Wada, Takahiro, Yoshimura, Koki, Doi, Shun-ichi, Youhata, Hironori, and Tomiyama Koichi. “Proposal of An Eco-Driving Assist System Adaptive to Driver’s Skill.” *14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011, pp.1880-1885.
65. Wang, Jianqiang, Meng Lu, and Li, Keqiang. “Characterization of Longitudinal Driving Behavior by Measurable Parameters.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2185, Transportation Research Board of the National Academies, Washington, D.C., 2010, pp. 15-23.
66. Winsum, Wim van. “The Human Element in Car Following Models.” *Transportation Research Part F*, Vol. 2, No. 4, 1999, pp. 207-211
67. Yang, Hong, and Ozbay, Kaan. “Estimation of Traffic Conflict Risk for Merging Vehicles on Highway Merge Section.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2236, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 58–65.
68. Yeo, Hwasoo, Skabardonis, Alexander, Halkias, John, Colyar, James, and Alexiadis, Vassili. “Oversaturated Freeway Flow Algorithm for Use in Next Generation Simulation.” *Transportation Research Record: Journal of the Transportation Research Board*, No. 2088, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 68–79.

Appendix A. Source Codes for Dissertation

All source codes were developed using MATLAB.

A - 1. Importing NGSIM Trajectory data to Variables

```
function [DB varargout]=importingNGSIMData(varargin)
%% importingData
% import Trajectory data of file for dessertation
%
%% syntax
%   DB=importData(sourceType);
%   [DB informDB]=importData(sourceType);
%   DB=importData(sourceType, 'PropertyName',PropertyValue);
%   [DB informDB]=importData(sourceType, 'PropertyName',PropertyValue);
%
%% description

%% Recording Basic information of importing
Time=clock;
madeby='Heejin Jung';
disp('1. Importing Process is executing...');
disp(' ');
disp(' ');
disp(['Imported by ', madeby]);
disp(' ');
disp(['Importing time: ', datestr(Time)]);
disp(' ');

%%  Setting Default Properties of File Format
disp(' ');
disp('2. Setting default Properties of File Format');

 properties.name={'fileterSpec'...
     ;'dialogTitle'...
     ;'format'...
     ;'header'...
     ;'delimiter'...
     ;'headerLines'...
     ;'source'...
     ;'type'...
     ;'collectedBy'...
     ;'sourceType'};

 properties.value=   {{'*.*'}...
     ;'Pick a Trajectory File of NGSIM'...
     ;'%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f'...
     ;{'veh_ID', 'Frame_ID', 'Total_Frames', 'Global_Time'...
     ,'Local_X','Local_Y','Global_X', 'Global_Y'...
     ,'Vehicle_Length','Vehicle _Width','Vehicle_Class'...
     ,'Vehicle_Velocity','Vehicle_Acceleration'...
     ,'Lane_Identification','Preceding_Vehicle'...
     ,'Following_Vehicle','Spacing','Headway'}...
     ;'whiteSpace'...
     ;0  ...
     ;'Next Generation SIMulation'...
```

```
    ;'trajectory data'...
    ;'Heejin Jung'...
    ;'NGI80tj'};

disp(' ');
disp('>>>> Completed Setting Default Properties of File Format');

disp(' ');disp(' ');disp(' ');

%% Reading File
filterSpec  = getProperty(properties,'fileterSpec');
fformat     = getProperty(properties,'format');
DialogTitle = getProperty(properties,'dialogTitle');
delimiter   = getProperty(properties,'delimiter');
HeaderLines = getProperty(properties,'headerLines');

disp('????? Please, pick a file ?????');
disp('...');

[filename, pathname]=uigetfile(filterSpec,DialogTitle);

disp('***** Accepted your asking ****');
disp(' ');

% openning file
disp('2.2. Calling the file.');
disp('...');

fid=fopen([pathname, filename]);

% reading file
disp(' ');
disp('2.3. Reading the file.');
disp('...');

if strcmpi(delimiter,'whiteSpace')
   rawDB=textscan(fid,fformat, 'HeaderLines', HeaderLines);

else
   rawDB=textscan(fid,fformat, 'delimiter', delimiter, 'HeaderLines', HeaderLines);

end

% closing file
disp('    Complete reading the file');
disp(' ');



fclose all;

%% making output arguements
DB.information.madeby=madeby;
DB.information.Time=datestr(Time);
DB.information.file=[pathname,filename];
DB.head=getProperty(properties,'header');
```

```
DB.data=cell2mat(rawDB);
function properties=setProperty(properties, varargin)

N=numel(varargin);

n=evenOrOdd(N);


while ~isempty(varargin)
    % confirm that # of elements of varargin is even.
    if n
        %% find property
        indx1=strcmpi(properties.name, varargin{1});
        indx2=find(indx1);

        if ~isempty(indx2)
            %% existing value
            if iscell(properties.value)
                propertyValueOld=properties.value{indx2};

                %% confirming class of value
                cOldVal=class(propertyValueOld);

                if isa(varargin{2},cOldVal)
                    %% change property value for new one
                    properties.value{indx2}=varargin{2};

                else
                    %% error in class of property value
                    disp('XXX Incorrected class of preperty value XXX');
                    disp('');
                    disp([' The name of property is ', varargin{1}, '.']);
                    disp([' The orignal class of property value must be ',cOldVal, '.']);
                    disp([' The class of your value is ', class(varargin{2}), '.']);
                    disp('');
                    disp([' The value of ',varargin{1}, ' was not changed.']);

                end
                varargin(1:2)=[];

            elseif isa(properties.value, 'double')
                if isa(varargin{2},'double')
                    s1=size(properties.value(indx2,:));
                    s2=size(varargin{2});
                    if s1(1)==s2(1) && s1(2)==s2(2)
                        properties.value(indx2,:)=varargin{2};
                         varargin(1:2)=[];

                    else
                        disp('dismatch the size of properties value');
                        break;
                    end

                else
                    disp('dismatch the class of properties value');
                    break;
```

```
            end

        else
            disp('the class of properties value is not available');
            break;
        end


    else
        %% error in name of property
        disp('XXX Incorrected preperty name XXX');
        disp('');
        disp('property could not be found!!');
        disp(' The list of properties are the following:');
        char(properties.name)
        disp('');
        disp([' The name of your property is ', varargin{1}, '.']);
        disp('');
        disp([' It defaulted!! .']);

        varargin=[];

    end



else
    varargin=[];
    %% error in  pair of arguements
    disp('XXX Incorrected pair of input arguments XXX');
    disp('');
    disp('too many arguement!!');
    disp('');
    disp([' It defaulted!! .']);

end

end

%% subfunction
function indx=evenOrOdd(n)
n=n/2;
n=n-floor(n);
indx=n==0;
```

A - 2. setProperty.m

```
function properties=setProperty(properties, varargin)
 N=numel(varargin);
 n=evenOrOdd(N);
 while ~isempty(varargin)
   % confirm that # of elements of varargin is even.
   if n
       %% find property
       indx1=strcmpi(properties.name, varargin{1});
       indx2=find(indx1);

       if ~isempty(indx2)
           %% existing value
           if iscell(properties.value)
               propertyValueOld=properties.value{indx2};

               %% confirming class of value
               cOldVal=class(propertyValueOld);

               if isa(varargin{2},cOldVal)
                   %% change property value for new one
                   properties.value{indx2}=varargin{2};

               else
                   %% error in class of property value
                   disp('XXX Incorrected class of preperty value XXX');
                   disp('');
                   disp([' The name of property is ', varargin{1}, '.']);
                   disp([' The orignal class of property value must be ',cOldVal, '.']);
                   disp([' The class of your value is ', class(varargin{2}), '.']);
                   disp('');
                   disp([' The value of ',varargin{1}, ' was not changed.']);

               end
               varargin(1:2)=[];

           elseif isa(properties.value, 'double')
               if isa(varargin{2},'double')
                   s1=size(properties.value(indx2,:));
                   s2=size(varargin{2});
                   if s1(1)==s2(1) && s1(2)==s2(2)
                       properties.value(indx2,:)=varargin{2};
                        varargin(1:2)=[];

                   else
                       disp('dismatch the size of properties value');
                       break;
                   end

               else
                   disp('dismatch the class of properties value');
                   break;
```

```
        end

      else
        disp('the class of properties value is not available');
        break;
      end


    else
      %% error in name of property
      disp('XXX Incorrected preperty name XXX');
      disp('');
      disp('property could not be found!!');
      disp(' The list of properties are the following:');
      char(properties.name)
      disp('');
      disp([' The name of your property is ', varargin{1}, '.']);
      disp('');
      disp([' It defaulted!! .']);

      varargin=[];

    end




  else
    varargin=[];
    %% error in  pair of arguements
    disp('XXX Incorrected pair of input arguments XXX');
    disp('');
    disp('too many arguement!!');
    disp('');
    disp([' It defaulted!! .']);

  end

end

%% subfunction
function indx=evenOrOdd(n)
n=n/2;
n=n-floor(n);
indx=n==0;
```

A - 3. getProperties.m

```
function varargout=getProperty(properties,varargin)
%N=numel(varargin)
% find property
k=1;
while ~isempty(varargin)
   %% find property
   indx1=strcmpi(properties.name, varargin{1});
   indx2=find(indx1);

   if ~isempty(indx2)
      varargout{k}=properties.value{indx2};
      varargin(1)=[];

   else
      disp('XXX Incorrected preperty name XXX');
      disp('');
      disp('property could not be found!!');
      disp(' The list of properties are the following:');
      char(properties.name)
      disp('');
      disp([' The name of your property is ', varargin{1}, '.']);
      disp('');
      disp([' The blank array was returned as the value of the property ',varargin{1}, '.']);

      varargout{k}=[];
      varargin(1)=[];

   end

end
```

A - 4. converting raw data to numeric data

```
function [nDB varargout]=convertingDB(rawDB,informDB, varargin)
%% Converting Data into numeric Databse and Information variable
%% properties
disp('1. setting properties');

properties.name={'preFuncs'...
            ;'funcs'...
            ;'postFuncs'};

properties.value={{'CVRstrTime2num(DB, ''time'', ''::'' )'...
              ;'end'}...
              ...
             ;{'CVRconvertVehID(DB,B,I,''OBE_ID'')'...
             ;'CVRconvertTime(DB,B,I, ''time'')'...
             ;'CVRconvertSpendT(DB,B,I, ''secs'')'...
             ;'CVRconvertSpeed(DB,B,I, ''speed'')'...
             ;'CVRconvertGPS(DB,B,I, {''long'',''lat''})'...
             ;'CVRconvertL(DB,B,I, {''x(ft)'',''y(ft)''})'...
```

```
            ;'CVRconvertDir(DB,B,I, ''dir'')'...
            ;'end'}...
            ...
            ;{'end'}};

if strcmpi(informDB.general.sourceType,'NGI80tj')
   disp(['$ Configuring Properties for ',informDB.general.source]);

   properties=setProperty( properties...
                  ,'preFuncs',{'end'}...
                  ...
               ,'funcs',{'NGSIMconvertVehID(DB,B,I,{''veh_ID'';''Total_Frames'';''Vehicle_Length'';''Vehicle
_Width'';''Vehicle_Class''})'...% 1
                         ;'NGSIMconvertTime(DB,B,I,''Frame_ID'')'...% 2
                         ;'NGSIMconvertSpendT(DB,B,I,''Frame_ID'')'...% 3
                         ;'NGSIMconvertSpeed(DB,B,I,''Vehicle_Velocity'')'...% 4
                         ;'NGSIMconvertAccel(DB,B,I,''Vehicle_Acceleration'')'...% 5
                         ;'NGSIMconvertGPS(DB,B,I,{''Global_X'';''Global_Y''})'...% 6
                         ;'NGSIMconvertL(DB,B,I,{''Local_X'';''Local_Y''})'...% 7
                         ;'NGSIMconvertLink(DB,B,I,''Lane_Identification'')'...% 8
                         ;'NGSIMconvertflwVeh(DB,B,I,''Following_Vehicle'')'...% 9
                         ;'NGSIMconvertLdVeh(DB,B,I,''Preceding_Vehicle'')'...% 10
                         ;'NGSIMconvertHW(DB,B,I, {''Spacing'';''Headway''})'...%11
                         ;'end'}...
                  ...
                  ,'postFuncs',{'end'}...
                  );

      disp(['$ Executing conversion for ',informDB.general.source]);

else
      disp('$ Executing conversion for Conneted Vehicle Research (default) >>>');

end

%% Configure properties
disp('$ Additional Congiguration of Properties');

if ~isempty(varargin)
   try
      while ~isempty(varargin)
         propertyName=varargin{1};
         propertyValue=varargin{2};

         varargin(1)=[];
         varargin(2)=[];

         Properties=setProperty(Properties,propertyName, propertyValue);

         disp(['$ Executing conversion for modified',informDB.general.source,' >>>']);

      end
```

```
    catch
        disp('too many arguements');
        disp('failed configuration');

        if strcmpi(informDB.general.sourceType,'NGI80tj')
            disp(['$ Executing conversion for unmodified',informDB.general.source,' >>>']);

        else
            disp('$ Executing conversion for Conneted Vehicle Research (default) >>>');

        end
    end

end

%% Initialization
disp('2. Initializing...');

DB=rawDB;
B=[];
I.tag=informDB.general.sourceType(1:end-2);
I.span=[155855,161537];

%% PreFunction: time- str->numeric of D (optional process)
disp('3. Pre-Process');

preFNum=1;
funcs=getProperty(properties, 'preFuncs');
func=funcs{preFNum};

if strcmpi(func,'end')
    disp('$ There is no pre-process');

end

while ~strcmpi(func,'end')
    disp(['$ ',num2str(preFNum), 'th pre-process:']);

    DB.data=eval(func);

    preFNum=preFNum+1;
    funcs=getProperty(properties, 'preFuncs');
    func=funcs{preFNum};

end



%% regular processes
disp('4. Regular Process');

prcssNum=1;
```

```matlab
funcs=getProperty(properties, 'funcs');
func=funcs{prcssNum};

if strcmpi(func,'end')
   disp('$ There is no regural process');

end

while ~strcmpi(func,'end')
   disp(['$ ',num2str(prcssNum), 'th process:']);

   if ~strcmpi(func,'none')
      [I,B]=eval(func);

   end

   prcssNum=prcssNum+1;
   funcs=getProperty(properties, 'funcs');
   func=funcs{prcssNum};

end

%% post Function: (optional process)
disp('5. Post-Process');

postFNum=1;
funcs=getProperty(properties, 'postFuncs');
func=funcs{postFNum};

if strcmpi(func,'end')
   disp('$ There is no post-process');

end

while ~strcmpi(func,'end')
   disp(['$ ',num2str(postFNum), 'th pre-process:']);

   DB=eval(func);

   postFNum=postFNum+1;
   funcs=getProperty(properties, 'postFuncs');
   func=funcs{postFNum};

end

informDB.data=I;
nDB=DB;
varargout{1}=informDB;
varargout{2}=B;
```

A - 5. Separating numeric Data into individual vehicle data and individual relative data

```
function [sepDB, DB,relativeDB]=NGSIMSeparatedDB(numericDB, informDB)

getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

colVehID=getIndx(informDB.data.vehID.listHeadings, 'numeric_vehicle ID');

vehIDs=informDB.data.vehID.list{colVehID};

colVehID=getIndx(informDB.data.vehID.listHeadings, 'string_vehicle ID');

dataHeadings.vehID=informDB.data.vehID.list;

s=size(numericDB);

database=zeros(s(1),42);
relDB=zeros(s(1),14);
stdRate=0;

s=size( informDB.data.vehID.list);
vehicles=cell(s(1),s(2)+3);

headingsList=[ informDB.data.vehID.listHeadings,{'start lane', 'end lane', 'numLC'}];

vehicles(1:end-3)= informDB.data.vehID.list;
vehicles{end-2}=zeros(size(vehicles{1}));
vehicles{end-1}=vehicles{end-2};
vehicles{end}=vehicles{end-2};

%numCol=length(vehicles);

%removeIndx=11;

%for m=1:numCol
%    vehicles{m}(removeIndx:end)=[];
%end

vehIDs=vehicles{colVehID};
numVeh=length(vehIDs);
data=cell(2,numVeh);
k=1;

while ~isempty(vehIDs)
    vehIDs(1)=[];

    %% first: Colecting subDatabase for a specific vehicle
    %indice=findIndxofVehID(informDB, vehID);
    [subDB, indxDB]=NGSIMCollectingSubDB(numericDB,informDB, k);
    % subDB=NGSIMSepRemove(subDB);
```

```
    %% second: Allocating travel time record
    [subDB, vehicles]=NGSIMSepSecs(subDB,headingsList,vehicles,k);

    %% third: Recording number of lane-changing into the table of vehicles
    [subDB, vehicles]=NGSIMSepLC(subDB,vehicles,k);

    %% forth: Adding the data of following vehicle into the subdatabase
    subDB=NGSIMSepFV(numericDB, subDB);

    %% fifth: Adding the data of leading vehicle into the subdatabase
    subDB=NGSIMSepLV(numericDB, subDB);

    %% sixth: Adding distance headway
    subDB=NGSIMSepHW(subDB);

    %% seventh: Computing data of relative speed, and adding them into the subdatabase
    subDB=NGSIMSepRS(subDB);
    s=size(subDB.data);

    %% containing the subdatabase into the entire database
    database(indxDB,1:s(2))=subDB.data;

    %% reconstructing relative database using subdatabase
    rDB=NGSIMRelativeData(subDB);

    %% containing the relative database into the entire relative database
    relDB(indxDB,:)=rDB.data;
    %%
    data{1,k}=subDB.data;
    data{2,k}=rDB.data;

    rateProcess=round(k/numVeh*100);

    if rateProcess>=stdRate
        disp(['>>> The rate of process: ', num2str(rateProcess),' %. ']);

        stdRate=stdRate+1;

    end

    k=k+1;

end

dataHeadings.subdata=subDB.dataHead;
dataHeadings.reldata=rDB.head;
dataHeadings.vehicles=headingsList;

sepDB.head=dataHeadings;
DB.head=dataHeadings.subdata;
relativeDB.head=dataHeadings.reldata;
DB.vehicleHead=dataHeadings.vehicles;
```

```
sepDB.data=data;
DB.data=database;
DB.vehicles=vehicles;
relativeDB.data=relDB;
```

A - 6. Collecting subDatabase for a specific vehicle

```
function [subDB, indxDB]=NGSIMCollectingSubDB(DB,I, index)

%% Head
getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

% Head End

%% Body
if ~isempty(index)
   %% data setup
   indxDBHead= getIndx(DB.head,'V_index');

   if ~isempty(indxDBHead)
      indxDB=find(DB.data(:,indxDBHead)==index);

      if ~isempty(indxDB)
         data=DB.data(indxDB,:);

      else
         data=[];

      end

   else
      data='error';
      indxDB=[];
   end

else
   headingsList='none';
   vehicles=[];
   headingsData='none';
   data=[];
   indxDB=[];
end

subDB.dataHead=DB.head;
subDB.data=data;

% Body End
```

A - 7. Allocating travel time record

```
 function [subDB, vehicles]=NGSIMSepSecs(subDB,headingsList,vehicles,k)
```

```
indx1=find(strcmpi(subDB.dataHead,'sec (sec)'));

if ~isempty(indx1)
   %% fixing local time record
   secs1=round(subDB.data(:,indx1)*10)/10;

   secs2=round((secs1-secs1(1))*10)/10;


   %% confirming total travel time
   indx2=find(strcmpi(headingsList,'travel time (sec)'));

   if ~isempty(indx2)
      totalTravelTime=vehicles{indx2}(k);

   else
      totalTravelTime=[];

   end


   if ~isempty(totalTravelTime)

      %% update time record secs
      if totalTravelTime==secs2(end)
         subDB.data(:,indx1)=secs2;

         delSecs=[secs2(2:end)-secs2(1:end-1);0];

         subDB.data=[subDB.data,delSecs];
         subDB.dataHead=[subDB.dataHead,{'delT'}];

        % disp(' successfully fixing secs');

      else
         disp(' unsuccessfully fixinf secs');

      end

   else
      disp(' We did not have total travel time of this vehicle!');
      vehicles{indx2}(k)=secs2(end);

   end

else
   disp(' This database does not have the data of time record as secs~')

end
```

A - 8. Recording number of lane-changing into the table of vehicles

```
function [subDB, vehicles]=NGSIMSepLC(subDB,vehicles,index)

indx=find(strcmpi(subDB.dataHead,'Link'));

if ~isempty(indx)
   Links=subDB.data(:,indx);

   detectingLC=Links(2:end)~=Links(1:end-1);
   LC=[0; detectingLC];

   subDB.data=[subDB.data,LC];
   subDB.dataHead=[subDB.dataHead,{'LC'}];

   numLC=sum(LC);

   vehicles{end-2}(index)=Links(1);
   vehicles{end-1}(index)=Links(end);
   vehicles{end}(index)=numLC;

else
   disp(' there is not Link col.');

end
```

A - 9. adding the data of following vehicle into the subdatabase

```
function subDB=NGSIMSepFV(DB, subDB)
%% Head
getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

%% BODY
colFollower=getIndx(subDB.dataHead,'follower (numeric)');
colTimes=getIndx(subDB.dataHead,'time (hhmmss)');

colVID=getIndx(DB.head,'numeric_vehicle ID');
colTime=getIndx(DB.head,'time (hhmmss)');
colClass=getIndx(DB.head,'class');
colLeng=getIndx(DB.head,'vehicle length (ft)');
colSpeed=getIndx(DB.head,'speed (ft/s)');
colAccel=getIndx(DB.head,'accel (ft/s-sqr)');

a=~isempty(colTime)*~isempty(colClass)*~isempty(colLeng)*~isempty(colSpeed)*~isempty(colAccel)*~is
empty(colFollower);

if a
   followerID=subDB.data(:, colFollower); % calling folowing vehicle IDs
   Times=subDB.data(:, colTimes);
   numSets=numel(followerID);
   dataOfFlwVeh=[Times, followerID, zeros(numSets,4)];

   numFlw=sum(followerID>0);

   indice=find(followerID);
```

```
    while ~isempty( indice)
        indx=indice(1);
        indice(1)=[];

        vehid=followerID(indx);
        time=Times(indx);

        indx1=find(DB.data(:,colVID)==vehid);

        if ~isempty(indx1)
            dataBase=DB.data(indx1,:);

            indx2=find(dataBase(:,colTime)==time);

            if ~isempty(indx2)
            dataSet=dataBase(indx2,[colClass, colLeng, colSpeed, colAccel]);

            dataOfFlwVeh(indx,[3:end])=dataSet;

            end

        end

    end
    subDB.dataHead=[subDB.dataHead,{'flwVeh class', 'flwVeh length (ft)', 'flwVeh speed (ft/s)', 'flwVeh
accel (ft/s-sqr)'}];

    subDB.data=[ subDB.data, dataOfFlwVeh(:, [3: end])];

end
```

A - 10. Adding the data of leading vehicle into the subdatabase

```
function subDB=NGSIMSepLV(DB, subDB)
%% Head
getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

%% BODY
colLeader=getIndx(subDB.dataHead,'leader (numeric)');
colTimes=getIndx(subDB.dataHead,'time (hhmmss)');

colVID=getIndx(DB.head,'numeric_vehicle ID');
colTime=getIndx(DB.head,'time (hhmmss)');
colClass=getIndx(DB.head,'class');
colLeng=getIndx(DB.head,'vehicle length (ft)');
colSpeed=getIndx(DB.head,'speed (ft/s)');
colAccel=getIndx(DB.head,'accel (ft/s-sqr)');

a=~isempty(colTime)*~isempty(colClass)*~isempty(colLeng)*~isempty(colSpeed)*~isempty(colAccel)*~is
empty(colLeader);
```

```
if a
   leaderID=subDB.data(:, colLeader); % calling preceding vehicle IDs
   Times=subDB.data(:, colTimes);
   numSets=numel(leaderID);
   dataOfLdVeh=[Times, leaderID, zeros(numSets,4)];

   numLd=sum(leaderID>0);

   indice=find(leaderID);

   while ~isempty( indice)
      indx=indice(1);
      indice(1)=[];

      vehid=leaderID(indx);
      time=Times(indx);

      indx1=find(DB.data(:,colVID)==vehid);

      if ~isempty(indx1)
         dataBase=DB.data(indx1,:);

         indx2=find(dataBase(:,colTime)==time);

         if ~isempty(indx2)
         dataSet=dataBase(indx2,[colClass, colLeng, colSpeed, colAccel]);

         dataOfLdVeh(indx,[3:end])=dataSet;

         else
            disp('error2');

         end

      else
         disp('error1');

      end

   end
   subDB.dataHead=[subDB.dataHead,{'ldVeh class', 'ldVeh length (ft)', 'ldVeh speed (ft/s)', 'ldVeh accel
(ft/s-sqr)'}];

    subDB.data=[ subDB.data, dataOfLdVeh(:, [3: end])];

else
      disp('error0');

end
```

A - 11. adding distance headway

```
function subDB=NGSIMSepHW(subDB)
```

```
getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

colIndxLd=getIndx(subDB.dataHead,'leader (numeric)');
colIndxSP=getIndx(subDB.dataHead,'spacing (ft)');
%colIndxLdLeng=getIndx(subDB.dataHead,'ldVeh length (ft)');
%colIndxHVLeng=getIndx(subDB.dataHead,'vehicle length (ft)');

a=~isempty(colIndxSP)*~isempty(colIndxLd);%*~isempty(colIndxLdLeng)*~isempty(colIndxHVLeng);

if a
   ldVID=subDB.data(:, colIndxLd);
   indx=find(ldVID>0);

   distance=subDB.data(:, colIndxSP);
   %leadVehLeng=subDB.data(:, colIndxLdLeng);
   %hostVehLeng=subDB.data(:,colIndxHVLeng);

   spacing=NaN*ones(size(distance));

   if ~isempty(indx)
      spacing(indx)=distance(indx);%-0.5*(leadVehLeng(indx)+hostVehLeng(indx));

   end

   subDB.dataHead=[subDB.dataHead,{'relative distance (ft)'}];

   subDB.data=[ subDB.data, spacing];

end
```

A - 12. Computing data of relative speed, and adding them into the subdatabase

```
function subDB=NGSIMSepRS(subDB)

getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));

colIndxLd=getIndx(subDB.dataHead,'leader (numeric)');
indxHVSP=getIndx(subDB.dataHead, 'speed (ft/s)');
indxLVSP=getIndx(subDB.dataHead, 'ldVeh speed (ft/s)');

a=~isempty(indxHVSP)*~isempty(indxLVSP)*~isempty(colIndxLd);

if a
   ldVID=subDB.data(:, colIndxLd);
   indx=find(ldVID>0);

   hostSpeed=subDB.data(:, indxHVSP);
   leadSpeed=subDB.data(:, indxLVSP);

   rSpeed=NaN*ones(size(hostSpeed));
```

```
    if ~isempty(indx)
        rSpeed(indx)= hostSpeed(indx)- leadSpeed(indx);

    end

    subDB.dataHead=[subDB.dataHead,{'relative speed (ft/s)'}];

    subDB.data=[ subDB.data, rSpeed];

end
```

A - 13. reconstructing relative database using subdatabase

```
function rDB=NGSIMRelativeData(subDB, varargin)
properties.name={'column name', 'order'};
properties.value={...
                       {'numeric_Vehicle ID'...
                        ;'class'...
                        ;'time (hhmmss)'...
                        ;'sec (sec)'...
                        ;'speed (ft/s)'...
                        ;'accel (ft/s-sqr)'...
                        ;'Link'...
                        ;'leader (numeric)'...
                        ; 'LC'...
                        ;'ldVeh class'...
                        ;'ldVeh speed (ft/s)'...
                        ;'ldVeh accel (ft/s-sqr)'...
                        ;'relative distance (ft)'...
                        ;'relative speed (ft/s)'}...
                        ,1:14};

getIndx=@ (DBheads, heading) find(strcmpi(DBheads, heading));


if ~isempty(varargin)
    setVal=varargin;

    while ~isempty(setVal)
        item=setVal{1};
        setVal(1)=[];

        if ~isempty(setVal);
            val=setVal{1};
            setVal(1)=[];
        else
            disp('wrong arguments');
            item=[];
            val=[];
        end

        if ~isempty(item)
            properties=setProperty(properties, item, val);
```

```
        end

    end

end


colNames=getProperty(properties, 'column name');
colOrder=getProperty(properties, 'order');

numCol=length(colOrder);



colIndx=zeros(1,numCol);
colHead=cell(1,numCol);

for k=1:numCol
    indx=getIndx(subDB.dataHead,colNames{colOrder(k)});

    if ~isempty(indx)
        colHead(k)=colNames(colOrder(k));
        colIndx(k)=indx;

    else
        colHead(end)=[];
        colIndx(end)=[];

    end


end

rDB.head=colHead;
rDB.data=subDB.data(:,colIndx);
```

A - 14. Microscopic Analysis: Operating Acceleration

```
function MICROsepDB=microAnaly01(sepDB)
%%
% To counting acceleration and brake for each vehicle

vehicles=sepDB.data(1,:);

numVehs=size(vehicles,2);

microAnalysis=cell(7,numVehs);

for k=1: numVehs
    accels=vehicles{k}(:,15);
```

```
    states=~isnan(vehicles{k}(:,42));
    [count, time1, time2 ,FAcount, FAtime1, FAtime2 , oprtAccel]=countingAccelBrake(accels, states);

    microAnalysis{1,k}=count;% 3
    microAnalysis{2,k}=time1;%4
    microAnalysis{3,k}=time2;%5
    microAnalysis{4,k}=FAcount;%6
    microAnalysis{5,k}=FAtime1;%7
    microAnalysis{6,k}=FAtime2;%8
    microAnalysis{7,k}=[oprtAccel, accels, states];%9

end

MICROsepDB.data=[sepDB.data; microAnalysis];
MICROsepDB.head.subDB=sepDB.head.subdata;
MICROsepDB.head.relativeDB=sepDB.head.reldata;
MICROsepDB.head.number={'# of acceleration', '# of recovery acceleration', '# of recovery brake', '# of brake'};
MICROsepDB.head.secs={'secs of acceleration', 'secs of recovery acceleration', 'secs of recovery brake', 'secs of brake'};
MICROsepDB.head.ratio={'ratio of acceleration', 'ratio of recovery acceleration', 'ratio of recovery brake', 'ratio of brake'};

MICROsepDB.head.FAnumber={'# of acceleration in FA', '# of recovery acceleration in FA', '# of recovery brake in FA', '# of brake in FA'};
MICROsepDB.head.FAsecs={'secs of acceleration in FA', 'secs of recovery acceleration in FA', 'secs of recovery brake in FA', 'secs of brake in FA'};
MICROsepDB.head.FAratio={'ratio of acceleration in FA', 'ratio of recovery acceleration in FA', 'ratio of recovery brake in FA', 'ratio of brake in FA'};

MICROsepDB.head.data={'operating acceleration', 'acceleration', 'states of FA or not'};
```

A - 15. Counting Accelerating and Braking

```
function [count, time1, time2, FAcount, FAtime1, FAtime2 , accels]=countingAccelBrake(accels, states)
%%
% To counting # aond time of brake and acceleration

%% Syntax
%  [count, time1, time2, FAcount, FAtime1, FAtime2 , accels]=countingAccelBrake(accels, states);
%
%% Inputs
% accels is accelerations in ft/sec. it is vectors
% states is index number that indicates car following or not
% if it is free agent, state is 1, unless 0.
% states is vector
%
%% outputs
% count is a vector that is number of [acceleration, recovery of
% acceleration, recovery brake and brake].
% time1 is a vector of total time of [acceleration, recovery of acceleration,
% recovery brake and brake] in seconds.
% time2 is a vector of ratio of time of [acceleration, recovery of
% acceleration, recovery brake and brake].
```

```
% FA~ means data for only data in Free agent.


%% initialization
oprtAccel=[];
preAccel=[];

totalTime=numel(states);
totalFATime=numel(find(states));

countAccel=0;
countRAccel=0;
countRBrake=0;
countBrake=0;

timeAccel=0;
timeRAccel=0;
timeRBrake=0;
timeBrake=0;

FAstate=0;
FAcountAccel=0;
FAcountRAccel=0;
FAcountRBrake=0;
FAcountBrake=0;

FAtimeAccel=0;
FAtimeRAccel=0;
FAtimeRBrake=0;
FAtimeBrake=0;

startIndx=0;
endIndx=0;

FAstartIndx=0;
FAendIndx=0;

% state=0 is noncontroled, 1 is acceleration and -1 is brake
state=0;
numData=size(accels,1);

critical=0.5*[-1,1];
indxAZeroPlus=accels<critical(2);
indxAZeroMinus=accels>critical(1);
indxZero=find(indxAZeroPlus.*indxAZeroMinus);

if ~isempty(indxZero)
   accels(indxZero)=0;

end

for k=1:numData
   if isempty(oprtAccel)
```

```
    oprtAccel=accels(k);

else
    switch state
        case 2 % Accelation
            if accels(k)< preAccel % stop Acceleration
                endIndx=k-1;
                indce=startIndx:endIndx;
                accels(indce)=oprtAccel;
                countAccel=countAccel+1;
                timeAccel=timeAccel+(endIndx-startIndx);

                if FAstate==1
                    FAcountAccel=FAcountAccel+1;

                    if states(k)==1
                        FAendIndx=k-1;
                        FAtimeAccel=FAtimeAccel+(FAendIndx-FAstartIndx);

                    end

                end


                if accels(k)<0 %start Brake
                    startIndx=k;
                    oprtAccel=accels(k);
                    state=-2;
                    FAstate=states(k);

                    if FAstate==1
                        FAstartIndx=k;

                    end


                else % start recover acceleration
                    startIndx=k;
                    oprtAccel=0;
                    state=1;
                    FAstate=states(k);

                    if FAstate==1
                        FAstartIndx=k;

                    end

                end

            else %% keep acceleration
                oprtAccel=accels(k);

                if FAstate==1 && states(k)==0
```

```
            FAcountAccel=FAcountAccel+1;
            FAstate=states(k);
            FAendIndx=k-1;
            FAtimeAccel=FAtimeAccel+(FAendIndx-FAstartIndx);

        end

        if FAstate==0 && states(k)==1
            FAstate=states(k);
            FAstartIndx=k;

        end

    end

case 1 % recover acceleration
    if accels(k)>preAccel
        endIndx=k-1;
        indce=startIndx:endIndx;
        accels(indce)=oprtAccel;
        countRAccel=countRAccel+1;
        timeRAccel=timeRAccel+(endIndx-startIndx);

        if FAstate==1
            FAcountRAccel=FAcountRAccel+1;

            if states(k)==1
                FAendIndx=k-1;
                FAtimeRAccel=FAtimeRAccel+(FAendIndx-FAstartIndx);

            end

        end

        if accels(k)<0
            startIndx=k;
            oprtAccel=accels(k);
            state=-2;
            FAstate=states(k);

            if FAstate==1
                FAstartIndx=k;

            end

        else
            startIndx=k;
            oprtAccel=accels(k);
            state=2;
            FAstate=states(k);

            if FAstate==1
                FAstartIndx=k;
```

```
    end

  end
else
  if accels(k)<0 % start Brake
    endIndx=k-1;
    indce=startIndx:endIndx;
    accels(indce)=oprtAccel;
    countRAccel=countRAccel+1;
    timeRAccel=timeRAccel+(endIndx-startIndx);

    if FAstate==1
      FAcountRAccel=FAcountRAccel+1;

      if states(k)==1
        FAendIndx=k-1;
        FAtimeRAccel=FAtimeRAccel+(FAendIndx-FAstartIndx);

      end

    end

    startIndx=k;
    oprtAccel=accels(k);
    state=-2;
    FAstate=states(k);

    if FAstate==1
      FAstartIndx=k;

    end

  else % keep recovering accel
    oprtAccel=0;

    if FAstate==1 && states(k)==0 % end of FA
      FAcountRAccel=FAcountRAccel+1;
      FAstate=states(k);
      FAendIndx=k-1;
      FAtimeRAccel=FAtimeRAccel+(FAendIndx-FAstartIndx);

    end

    if FAstate==0 && states(k)==1 % start of FA
      FAstate=states(k);
      FAstartIndx=k;

    end

  end

end
```

```
case -2 %brake
   if accels(k)> preAccel % stop brake
      endIndx=k-1;
      indce=startIndx:endIndx;
      accels(indce)=oprtAccel;
      countBrake=countBrake+1;
      timeBrake=timeBrake+(endIndx-startIndx);

      if FAstate==1
         FAcountBrake=FAcountBrake+1;

         if states(k)==1
            FAendIndx=k-1;
            FAtimeBrake=FAtimeBrake+(FAendIndx-FAstartIndx);

         end

      end

      if accels(k)>0 %start Acceleration
         startIndx=k;
         oprtAccel=accels(k);
         state=2;
         FAstate=states(k);

         if FAstate==1
            FAstartIndx=k;

         end

      else % start recover brake
         startIndx=k;
         oprtAccel=0;
         state=-1;
         FAstate=states(k);

         if FAstate==1
            FAstartIndx=k;

         end

      end

   else % keep braking
      oprtAccel=accels(k);

      if FAstate==1 && states(k)==0 % end of FA
         FAcountBrake=FAcountBrake+1;
         FAstate=states(k);
         FAendIndx=k-1;
         FAtimeBrake=FAtimeBrake+(FAendIndx-FAstartIndx);
```

```
        end

        if FAstate==0 && states(k)==1 % start of FA
           FAstate=states(k);
           FAstartIndx=k;

        end

     end

  case -1 %recover brake
     if accels(k)<preAccel %stop recover brake
        endIndx=k-1;
        indce=startIndx:endIndx;
        accels(indce)=oprtAccel;
        countRBrake=countRBrake+1;
        timeRBrake=timeRBrake+(endIndx-startIndx);

        if FAstate==1
           FAcountRBrake=FAcountRBrake+1;

           if states(k)==1
              FAendIndx=k-1;
              FAtimeRBrake=FAtimeRBrake+(FAendIndx-FAstartIndx);

           end

        end

        if accels(k)>0 % start acceleration
           startIndx=k;
           oprtAccel=accels(k);
           state=2;
           FAstate=states(k);

           if FAstate==1
              FAstartIndx=k;

           end

        else
           startIndx=k;
           oprtAccel=accels(k);
           state=-2;
           FAstate=states(k);

           if FAstate==1
              FAstartIndx=k;

           end

        end
     else
```

```
        if accels(k)>0 % start Acceleration
            endIndx=k-1;
            indce=startIndx:endIndx;
            accels(indce)=oprtAccel;
            countRBrake=countRBrake+1;
            timeRBrake=timeRBrake+(endIndx-startIndx);

            if FAstate==1
                FAcountRBrake=FAcountRBrake+1;

                if states(k)==1
                    FAendIndx=k-1;
                    FAtimeRBrake=FAtimeRBrake+(FAendIndx-FAstartIndx);

                end

            end

            startIndx=k;
            oprtAccel=accels(k);
            state=2;
            FAstate=states(k);

            if FAstate==1
                FAstartIndx=k;

            end

        else % keep recovering brake
            oprtAccel=0;

            if FAstate==1 && states(k)==0 % end of FA
                FAcountRBrake=FAcountRBrake+1;
                FAstate=states(k);
                FAendIndx=k-1;
                FAtimeRBrake=FAtimeRBrake+(FAendIndx-FAstartIndx);

            end

            if FAstate==0 && states(k)==1 % start FA
                FAstate=states(k);
                FAstartIndx=k;

            end

        end

    end

otherwise% non-controled
    if accels(k)>preAccel %start acceleration
        startIndx=k;
        oprtAccel=accels(k);
```

```
                state=2;
                FAstate=states(k);

                if FAstate==1
                    FAstartIndx=k;

                end

            else if accels(k)<preAccel % start brake
                    startIndx=k;
                    oprtAccel=accels(k);
                    state=-2;
                    FAstate=states(k);

                    if FAstate==1
                        FAstartIndx=k;

                    end

                end
            end

        end

    end
    preAccel=accels(k);

end

%% construction of outputs
count=[countAccel countRAccel countRBrake countBrake];
time1=[timeAccel timeRAccel timeRBrake timeBrake]/10;
time2=round([timeAccel timeRAccel timeRBrake timeBrake]/totalTime*10000)/100;

FAcount=[FAcountAccel FAcountRAccel FAcountRBrake FAcountBrake];
FAtime1=[FAtimeAccel FAtimeRAccel FAtimeRBrake FAtimeBrake]/10;
FAtime2=round([FAtimeAccel FAtimeRAccel FAtimeRBrake FAtimeBrake]/totalFATime*10000)/100;
```

A - 16. batch file for microscopic Analysis

```
%% 1. Histogram
%A. Distribution of the average operating acceleration
close all

figure('name','Distribution of the average operating acceleration');

h=H.data;
h1a=plotingHistogramBar(h(:,3),' average operating acceleration','ft/s^2');

%B. Distribution of the average operating brake
figure('name','Distribution of the average operating brake');
h1b=plotingHistogramBar(h(:,6),' average operating brake','ft/s^2');
```

```
%% 2. Comparison of all situation and FA
MICROstatisticalAnalysis=MICROstatisticalAnalysis01;
% A. Comparison of all situation and free agent situation
figure('name','Comparison of all situation and free agent situation ');
h2a=classCompareAccBrk(MICROstatisticalAnalysis,0);

% B. Relationship between acceleration and brake
figure('name','Relationship between acceleration and brake ');
h2b=classCompareAccBrk(MICROstatisticalAnalysis,1);

%% 3. Classifying driving pattern
% A. based on average operating acceleration
figure('name','Classification based on average operating acceleration ');
MICROstatisticalAnalysis=classifyingDrivingPattern02(MICROstatisticalAnalysis,0);

% B. based on average operating brake
figure('name','Classification based on average operating brake ');
MICROstatisticalAnalysis=classifyingDrivingPattern02(MICROstatisticalAnalysis,1);

% C. based on average operating acceleration and brake
figure('name','Classification based on average operating acceleration and brake ');
MICROstatisticalAnalysis=classifyingDrivingPattern02(MICROstatisticalAnalysis,2);



%% 4. average number of acceleration and brake
analysis02;
```

A - 17. Plotting Histogram

```
function h=plotingHistogramBar02(H, titles, units)

X=H{1,1}(1,:);
Y=H{1,1}(2,:);
N=sum(Y);
y=round(Y/N*100)/100;

mu=H{2,1}(1);
sigma=H{2,1}(2);
sigma1=mu-sigma;
sigma2=mu+sigma;

%yLimit=ceil(max(Y)/100)*100;
yLimit=1;

h=bar(X,y);

ylim([0,yLimit]);
```

```
title(titles, 'fontsize', 16);
xlabel([' classes (', units, ')'],'fontsize', 14);
ylabel(' relative frequency','fontsize', 14);

line([mu,mu], [0,yLimit],'linewidth', 2, 'color', 'r');

line([sigma1,sigma1], [0,yLimit],'linewidth', 2, 'color', 'b');
line([sigma2,sigma2], [0,yLimit],'linewidth', 2, 'color', 'b');

text(mu, yLimit, num2str(mu),'VerticalAlignment','top', 'fontsize',12);

text(sigma1, yLimit, num2str(sigma1),'VerticalAlignment','top', 'fontsize',12);
text(sigma2, yLimit, num2str(sigma2),'VerticalAlignment','top', 'fontsize',12);

set(gca, 'XTick', round(X*100)/100, 'fontsize',12);

%%normal distribution curve
x=linspace(min(X), max(X));
points=pdf('norm', x, mu,sigma);

line(x, points, 'color','k', 'linewidth', 2);

grid on
```

A - 18. Analysis Emission based on classification of driving pattern 1

```
[emissions, totalAmounts, profiles]=EmissionsAnalysisVTM(MICROstatisticalAnalysis01);

 totalVehs=size(totalAmounts(7,:),2);
 totalFuelConsumption=sum(totalAmounts(7,:));
 totalHC=sum(totalAmounts(8,:));
 totalCO=sum(totalAmounts(9,:));
 totalNOx=sum(totalAmounts(10,:));
 totalCO2=sum(totalAmounts(11,:));

 m=1;
 indxDef=find(totalAmounts(m,:)==-1);
 indxMed=find(totalAmounts(m,:)==0);
 indxAgg=find(totalAmounts(m,:)==1);

 if ~isempty(indxDef)
    totalVehsDef=numel(indxDef)/totalVehs*100;
    totalFuelConsumptionDef=sum(totalAmounts(7,indxDef))/ totalFuelConsumption*100;
    totalHCDef=sum(totalAmounts(8,indxDef))/totalHC*100;
    totalCODef=sum(totalAmounts(9,indxDef))/totalCO*100;
    totalNOxDef=sum(totalAmounts(10,indxDef))/totalNOx*100;
    totalCO2Def=sum(totalAmounts(11,indxDef))/totalCO2*100;

 else
    totalVehsDef=0;
    totalFuelConsumptionDef=0;
    totalHCDef=0;
```

```
    totalCODef=0;
    totalNOxDef=0;
    totalCO2Def=0;

end

 if ~isempty(indxMed)
    totalVehsMed=numel(indxMed)/totalVehs*100;
    totalFuelConsumptionMed=sum(totalAmounts(7,indxMed))/ totalFuelConsumption*100;
    totalHCMed=sum(totalAmounts(8,indxMed))/totalHC*100;
    totalCOMed=sum(totalAmounts(9,indxMed))/totalCO*100;
    totalNOxMed=sum(totalAmounts(10,indxMed))/totalNOx*100;
    totalCO2Med=sum(totalAmounts(11,indxMed))/totalCO2*100;

 else
     totalVehsMed=0;
     totalFuelConsumptionMed=0;
     totalHCMed=0;
     totalCOMed=0;
     totalNOxMed=0;
     totalCO2Med=0;

 end


 if ~isempty(indxAgg)
     totalVehsAgg=numel(indxAgg)/totalVehs*100;
     totalFuelConsumptionAgg=sum(totalAmounts(7,indxAgg))/ totalFuelConsumption*100;
     totalHCAgg=sum(totalAmounts(8,indxAgg))/totalHC*100;
     totalCOAgg=sum(totalAmounts(9,indxAgg))/totalCO*100;
     totalNOxAgg=sum(totalAmounts(10,indxAgg))/totalNOx*100;
     totalCO2Agg=sum(totalAmounts(11,indxAgg))/totalCO2*100;

 else
     totalVehsAgg=0;
     totalFuelConsumptionAgg=0;
     totalHCAgg=0;
     totalCOAgg=0;
     totalNOxAgg=0;
     totalCO2Agg=0;

 end

 Y=[ totalVehsDef, totalVehsMed, totalVehsAgg;...
     totalFuelConsumptionDef,totalFuelConsumptionMed,totalFuelConsumptionAgg;...
     totalHCDef,totalHCMed, totalHCAgg;...
     totalCODef, totalCOMed,totalCOAgg;...
     totalNOxDef,totalNOxMed,totalNOxAgg];%...
     %totalCO2Def,totalCO2Med,totalCO2Agg];
 Y1=[numel(indxDef), numel(indxMed), numel(indxAgg);...
     sum(totalAmounts(7,indxDef)), sum(totalAmounts(7,indxMed)), sum(totalAmounts(7,indxAgg));...
     sum(totalAmounts(8,indxDef)), sum(totalAmounts(8,indxMed)), sum(totalAmounts(8,indxAgg));...
     sum(totalAmounts(9,indxDef)), sum(totalAmounts(9,indxMed)), sum(totalAmounts(9,indxAgg));...
```

```
    sum(totalAmounts(10,indxDef)), sum(totalAmounts(10,indxMed)),
sum(totalAmounts(10,indxAgg))];%...
    %sum(totalAmounts(11,indxDef)), sum(totalAmounts(11,indxMed)), sum(totalAmounts(11,indxAgg))];
  Y3=[numel(indxDef), numel(indxMed), numel(indxAgg);...
    sum(totalAmounts(7,indxDef))/numel(indxDef), sum(totalAmounts(7,indxMed))/numel(indxMed),
sum(totalAmounts(7,indxAgg))/numel(indxAgg);...
    sum(totalAmounts(8,indxDef))/numel(indxDef), sum(totalAmounts(8,indxMed))/numel(indxMed),
sum(totalAmounts(8,indxAgg))/numel(indxAgg);...
    sum(totalAmounts(9,indxDef))/numel(indxDef), sum(totalAmounts(9,indxMed))/numel(indxMed),
sum(totalAmounts(9,indxAgg))/numel(indxAgg);...
    sum(totalAmounts(10,indxDef))/numel(indxDef), sum(totalAmounts(10,indxMed))/numel(indxMed),
sum(totalAmounts(10,indxAgg))/numel(indxAgg)];%...
    %sum(totalAmounts(11,indxDef))/numel(indxDef), sum(totalAmounts(11,indxMed))/numel(indxMed),
sum(totalAmounts(11,indxAgg))/numel(indxAgg)];


  bar(Y,'stacked');
  set(gca,'fontsize',24);
  %title('Contribution of groups to total fuel consumption and emissions','fontsize',16);
  set(gca,'XTickLabel',{'# of vehicles'; 'Fuel consumption'; 'HC'; 'CO'; 'NO_x'},'fontsize', 30);
  %xlabel('Items','fontsize',30);
  ylabel('Ratio to Total Amount (%)','fontsize',30);
  legend('defensive drivers', 'moderate drivers','aggressive drivers');
  grid on

function [emissions, varargout]=EmissionsAnalysisVTM(DB)

data=DB.data;
numVehs=size(data,2);
emissions=cell(1,numVehs);
totalEmission=zeros(5,numVehs);
profileEmission=cell(1,numVehs);

for k=1:numVehs
   accels=DB.data{2,k}(:,6); %ft/s^2
   speed=DB.data{2,k}(:,5); %ft/s
   %vehicleType=DB.data{2,k}(1,2);
   Speed=speed*3600/3281; %km/h
   Acceleration=accels*3600/3281; %km/h-s
   VehicleType=4; %LDV3: Model Year>=1995, Engine Size<3.2 liters, and Mileage<83653

   emission =VTMicroExecutor(Speed, Acceleration, VehicleType);
   emissions{2,k}=emission;

   if nargout>=2
      totalEmission(1,k)=emission.totalAmount.Fuel;
      totalEmission(2,k)=emission.totalAmount.HC;
      totalEmission(3,k)=emission.totalAmount.CO;
      totalEmission(4,k)=emission.totalAmount.NOx;
      totalEmission(5,k)=emission.totalAmount.CO2;

   end

   if nargout>=3
```

```
        profileEmission{1,k}=[DB.data{2,k}(:,4), speed, accels, emission.profile.Fuel, emission.profile.HC,
emission.profile.CO, emission.profile.CO2];

    end


end


if nargout>=2
    varargout{1}=[cell2mat(DB.data(11:16,:));totalEmission];

end

if nargout>=3
    varargout{2}=profileEmission;

end


function emissions=VTMicroExecutor(Speed, Acceleration, VehicleType)
%% VTMicroExecutor.m
% Execute VTMicro.p function
%% Syntax
% Emission = VTMicro(Speed,Acceleration,VehicleType)
%
%%  -Input
%   Speed: Speed vector (km/h)
%   Acceleration: Acceleration vector (km/h/s)
%   Vehicle Type   1:ORNL 2:LDV1 3:LDV2 4:LDV3 5:LDV4 6:LDV5 7:LDT1 8:LDT2
%                  9:HE1 10:HE2 11:HE3 12:HE4
%
%%  -Output
%   Emssion.Fuel (Liter/second) Emission.HC  (grams/second)
%   Emission.CO  (grams/second) Emission.NOx (grams/second)
%   Emission.CO2 (grams/second)

Emission = VTMicro(Speed,Acceleration,VehicleType);

emissions.Units.profile.Fuel='Liter/deci-second';
emissions.Units.profile.HC='grams/deci-second';
emissions.Units.profile.CO='grams/deci-second';
emissions.Units.profile.NOx='grams/deci-second';
emissions.Units.profile.CO2='grams/deci-second';
emissions.Units.totalAmount.Fuel='mili-Liter';
emissions.Units.totalAmount.HC='mili-grams';
emissions.Units.totalAmount.CO='mili-grams';
emissions.Units.totalAmount.NOx='mili-grams';
emissions.Units.totalAmount.CO2='mili-grams';

emissions.profile.Fuel=Emission.Fuel/10;
emissions.profile.HC=Emission.HC/10;
emissions.profile.CO=Emission.CO/10;
emissions.profile.NOx=Emission.NOx/10;
```

```
emissions.profile.CO2=Emission.CO2/10;

emissions.totalAmount.Fuel=round(sum(emissions.profile.Fuel)*1000);
emissions.totalAmount.HC=round(sum(emissions.profile.HC)*1000);
emissions.totalAmount.CO=round(sum(emissions.profile.CO)*1000);
emissions.totalAmount.NOx=round(sum(emissions.profile.NOx)*1000);
emissions.totalAmount.CO2=round(sum(emissions.profile.CO2)*1000);
```

A - 19. percentile analysis for emissions

```
function [emissionsProfile, percentiles, percentilePercent]=percentileAnalysis01(emissions)

numVeh=size(emissions,2);

emissionsProfile=zeros(6, numVeh);

for k=1:numVeh
   emissionsProfile(1,k)=k;
   emissionsProfile(2,k)=emissions{2,k}.totalAmount.Fuel;
   emissionsProfile(3,k)=emissions{2,k}.totalAmount.HC;
   emissionsProfile(4,k)=emissions{2,k}.totalAmount.CO;
   emissionsProfile(5,k)=emissions{2,k}.totalAmount.NOx;
   emissionsProfile(6,k)=emissions{2,k}.totalAmount.CO2;

end

[Fuel, indxFuel]=sort(emissionsProfile(2,:),2);
totalFuel=sum(Fuel,2);

[HC, indxHC]=sort(emissionsProfile(3,:),2);
totalHC=sum(HC,2);

[CO, indxCO]=sort(emissionsProfile(4,:),2);
totalCO=sum(CO,2);

[NOx, indxNOx]=sort(emissionsProfile(5,:),2);
totalNOx=sum(NOx,2);

[CO2, indxCO2]=sort(emissionsProfile(6,:),2);
totalCO2=sum(CO2);

rankingFuel=emissionsProfile(:,indxFuel);
rankingHC=emissionsProfile(:,indxHC);
rankingCO=emissionsProfile(:,indxCO);
rankingNOx=emissionsProfile(:,indxNOx);
rankingCO2=emissionsProfile(:,indxCO2);

pm=20;

percentiles=zeros(6,pm);
percentilePercent=zeros(5,pm);
k=1;
```

```
for m=[5:5:95, 99]
    indx=round(m/100*numVeh+1/2);

    percentiles(1,k)=indx;
    percentiles(2,k)=rankingFuel(2,indx);
    percentiles(3,k)=rankingHC(3,indx);
    percentiles(4,k)=rankingCO(4,indx);
    percentiles(5,k)=rankingNOx(5,indx);
    percentiles(6,k)=rankingCO2(6,indx);
    k=k+1;

end

percentilePercent(1,:)=round(10000*percentiles(2,:)/totalFuel)/100;
percentilePercent(2,:)=round(10000*percentiles(3,:)/totalHC)/100;
percentilePercent(3,:)=round(10000*percentiles(4,:)/totalCO)/100;
percentilePercent(4,:)=round(10000*percentiles(5,:)/totalNOx)/100;
percentilePercent(5,:)=round(10000*percentiles(6,:)/totalCO2)/100;

h=bar(percentilePercent');
xlabel('percentile','fontsize',24);
set(gca,'xTick',0:20,'xTickLabel',{'';'5th';'10th'; '15th'; '20th'; '25th';
'30th';'35th';'40th';'45th';'50th';'55th';'60th'; '65th'; '70th'; '75th'; '80th';'85th';'90th';'95th';'99th'}, 'fontsize',18);
ylabel('percentage of fuel consumption and emissions (%)', 'fontsize',24);
legend({'Fuel';'HC';'CO';'NOx';'CO2'}, 'fontsize',24);
colormap copper;
grid on
```

A - 20. Evaluating Emissions and Fuel consumption using VT-Micro Model

```
function [emissions, varargout]=EmissionsAnalysisVTM002(DB)

data=DB.data;
numVehs=size(data,2);
emissions=cell(1,numVehs);
totalEmission=zeros(5,numVehs);
profileEmission=cell(1,numVehs);

for k=1:numVehs
    accels=DB.data{2,k}(:,6); %ft/s^2
    speed=DB.data{2,k}(:,5); %ft/s
    %vehicleType=DB.data{2,k}(1,2);
    Speed=speed*3600/3281; %km/h
    Acceleration=accels*3600/3281; %km/h-s
    VehicleType=4; %LDV3: Model Year>=1995, Engine Size<3.2 liters, and Mileage<83653

    emission =VTMicroExecutor(Speed, Acceleration, VehicleType);
    emissions{2,k}=emission;

    if nargout>=2
        totalEmission(1,k)=emission.totalAmount.Fuel;
        totalEmission(2,k)=emission.totalAmount.HC;
        totalEmission(3,k)=emission.totalAmount.CO;
```

```
        totalEmission(4,k)=emission.totalAmount.NOx;
        totalEmission(5,k)=emission.totalAmount.CO2;

    end

    if nargout>=3
        profileEmission{1,k}=[DB.data{2,k}(:,4), speed, accels, emission.profile.Fuel, emission.profile.HC,
emission.profile.CO, emission.profile.CO2];

    end


end


if nargout>=2
    varargout{1}=[DB.X';totalEmission];

end

if nargout>=3
    varargout{2}=profileEmission;

end
```

A - 21. Computing Average Target Operating Acceleration

```
function report=MicroscopicAnalysisPaper01(DB)

TrjDataSet=DB.data;

numVehs=size(TrjDataSet,2);

X=zeros(numVehs,2);

for k=1:numVehs
    %trjData01=TrjDataSet{2,k};
    toaData01=TrjDataSet{9,k};
    data=toaData01;
    numSets=size(toaData01,1);
    tOA=zeros(numSets,6);
    tOB=zeros(numSets,6);
    a=0;
    b=0;

   while ~isempty(data)

       %% calculating average Target Operating Acceleration
       indx01=find(data(:,1)~=0);

        if ~isempty(indx01)
            indx02=indx01(1);
```

```
        if indx02>=2
            data(1:indx02-1,:)=[];

        end

        toa=data(1,1); % initiating process
        strtAcc=data(1,2);

        indx03=data(:,1)==toa; % indicating process

        indx04=find(indx03==0); % indicating end of process

        lengthProcess=indx04(1)-1; % ending process

        indx05=find(data(:,2)==toa); % indicating reaching point

        reachingTime=indx05(1);

        if toa>0 % accelerating process
            a=a+1;
            indx06=find(tOA(:,1)==0);
            % [indxNo., taget operating acceleration, langth of
            % process, reaching time in sec, Acc/Dec rates
            tOA(indx06(1),:)=[a, toa, strtAcc, lengthProcess/10, reachingTime/10, (toa-
strtAcc)/reachingTime*10];

            data(1:lengthProcess,:)=[];

        else %breaking process
            b=b+1;
            indx06=find(tOB(:,1)==0);
            % [indxNo., taget operating acceleration, langth of
            % process, reaching time in sec, Acc/Dec rates
            tOB(indx06(1),:)=[b, toa, strtAcc, lengthProcess/10, reachingTime/10, (toa-
strtAcc)/reachingTime*10];

            data(1:lengthProcess,:)=[];

        end

    else
        data=[];

    end

  end

  indx07=find(tOA(:,1)==0);
  if ~isempty(indx07)
      tOA(indx07,:)=[];
```

```
   end
   indx07=find(tOB(:,1)==0);
   if ~isempty(indx07)
       tOB(indx07,:)=[];

   end

   avgTOA=mean(tOA(:,2));
   avgTOB=mean(tOB(:,2));

   X(k,:)=[avgTOA,avgTOB];

   report.heads.Table01={'process ID', 'start acceleration', 'target operating acceleration', 'langth of
process in sec', 'reaching time in sec', 'Acc/Dec rates'};
   report.Table01.targetOA{1,k}=tOA;
   report.Table01.targetOB{1,k}=tOB;

end

report.X=X;
```

A - 22. making vehicle definition file for CMEM

```
function [count, vehDef]=structingVehDefFile4CMEM(vehIDs,projectName, iterationNumber,
folder_name)

%% creating Activity  file
% initilization
extension='.def';

if iterationNumber<10
    filename=[projectName, '_00',num2str(iterationNumber), extension];

else if iterationNumber<100
        filename=[projectName, '_0',num2str(iterationNumber), extension];

    else
        filename=[projectName, '_',num2str(iterationNumber), extension];

    end
end

fid=fopen([folder_name,'\',filename],'w');

fprintf(fid, '# Vehicle Definition File for %s \n', projectName);
fprintf(fid, '# vehicle id, category, Soak Time, Specific Humidity \n');
numVeh=size(vehIDs,1);

categories=11*ones(numVeh,1);
soakTime=zeros(numVeh,1);
sH=75*ones(numVeh,1);

vehDef=[vehIDs,categories, soakTime,sH];
```

```
count=fprintf(fid, '%u\t%u\t%u\t%u \n',vehDef');

fclose(fid);
```

A - 23. Making Control file for the Batch Model of CMEM

```
%% creating control file
function folder_name=structingCotrolFile4CMEM(projectName, iterationNumber)
% initilization
fclose all
%projectName='NGSIM';
extension='.ctb';
%iterationNumber=1;

if iterationNumber<10
   filename=[projectName, '_00',num2str(iterationNumber), extension];

else if iterationNumber<100
       filename=[projectName, '_0',num2str(iterationNumber), extension];

   else
       filename=[projectName, '_',num2str(iterationNumber), extension];

   end
end

folder_name = uigetdir('C:\Users\Heejin\Documents\MATLAB\MicroScopic
Analysis\papers\paper2\Data\Emissions');

fid=fopen([folder_name,'\',filename],'w');

fprintf(fid, '# Control File for %s \n', projectName);

fprintf(fid, 'IN_UNITS = ENGLISH\n');

fprintf(fid, 'OUT_UNITS = ENGLISH\n');

fclose(fid);
```

A - 24. Making Activity File for CMEM

```
function [count, vehIDs]=structingActivityFile4CMEM(indiDB2,projectName, iterationNumber,
folder_name)
%% creating Activity  file
% initilization
extension='.atb';

if iterationNumber<10
   filename=[projectName, '_00',num2str(iterationNumber), extension];
```

```
else if iterationNumber<100
        filename=[projectName, '_0',num2str(iterationNumber), extension];

    else
        filename=[projectName, '_',num2str(iterationNumber), extension];

    end
end

ACTIVITIES=[];

numVeh=size(indiDB2.data,2);

for k=1:numVeh
    vehicle=indiDB2.data{1,k};
    activities=profileSpeed(vehicle, 1);
    ACTIVITIES=[ACTIVITIES;activities];

end

% vehicle IDs sort for CMEM
[sortedID, idnxSort]=sort(ACTIVITIES(:,2));

ACTIVITIES=ACTIVITIES(idnxSort,:);

% time sort for CMEM
[sortedTime, idnxSort]=sort(ACTIVITIES(:,1));

ACTIVITIES=ACTIVITIES(idnxSort,:);



fid=fopen([folder_name,'\',filename],'w');

fprintf(fid, '# Activity File for %s \n', projectName);
fprintf(fid, '# time, vehid, velocity, {acceleration}, {grade}, {secondary load}\n');

count=fprintf(fid, '%u,%u,%4.10f,,0,0 \n',ACTIVITIES');

vehIDs=unique(ACTIVITIES(:,2));

clear sortedID sortedTime idnxSort
fclose(fid);
```

A - 25. Importing Data from Batch Model Summary File of CMEM

```
function [C1, C2]=importingBMSummaryFile4CMEM(numVeh)

[fileName, pathName, FilterIndx]=uigetfile('*.smb','Select the Batch Model Summary File of CMEM');
filename=[pathName,fileName];
```

```
fid=fopen(filename, 'r');
mode=1;
while mode
   C1=textscan(fid,'%s', 1);

   if strcmpi(C1{1,1}{1,1}, '(miles)')
      mode=0;
      disp(C1{1,1}{1,1});

   end

end

C1=textscan(fid,'%s %u %s %4.2f', numVeh);

mode=1;
while mode
   C2=textscan(fid,'%s', 1);

   if strcmpi(C2{1,1}{1,1}, 'CO2')
      mode=0;
      disp(C2{1,1}{1,1});
   end

end

C2=textscan(fid,'%u  %u  %6.3f  %6.3f  %6.3f  %6.3f  %6.3f', numVeh);

fclose(fid);

clear ans fid fileName filename mode pathName
```

A - 26. Smoothing Algorithm for Recognizing Processes Algorithm

```
function [smthAccel, delAccel]=smoothingAlgrth4RcgnzdPrss00(accel, threshold1, threshold2)
%% Syntax
%       [smoothedAccel, delAccel]=smoothingAlgrth4RcgnzdPrss00(accel)
%
%% Arguements
%      INPUTS:
%         accel: a n-by-1colunm vector including acceleration of
%         vehicle in ft/sec^2.
%         threshold1: criterion for smoothing of small values of
%         acceleration near zero.
%         threshold2: criterion for smoothin of small values of variation
%         of acceleration near zero.
%
%      OUTPUTS:
%         smoothedAccel: a n-by-1 column vector including smoothed
%         acceleration in feet/sec^2.
%         delAccel: a n-by-1 column vector including smoothed variation
%         of acceleration in feet/sec^2.
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% smoothing acceleration
indx1=accel<=threshold1;
indx2=accel>=-threshold1;

indx=find(indx1.*indx2);

smthAccel=accel;

if ~isempty(indx)
    smthAccel(indx)=0;

end

%% Variation of acceleration
delAcc=smthAccel(2:end)-smthAccel(1:end-1);
delAccel=[0;delAcc];

%% smothing delAccel
indx1=delAccel<=threshold2;
indx2=delAccel>=-threshold2;
indx=find(indx1.*indx2);

if ~isempty(indx)
    delAccel(indx)=0;

end
```

A - 27. regime analysis: closing and opening regimes

```
function [vehicle, matrix, eventTable] =separatingAnalysis04(vehicle, matrix, error, minDurEv)

%% Data construction
%vehicleID=(indiDB4.matrix([1 2], n))'; % columnIndx ; vehicle ID

%error=0.3;
%minDurEv=[1, 0.4]; %seconds
%% initialization
indxCF=find(~isnan(vehicle(:,23)));

if ~isempty(indxCF)
    t=vehicle(indxCF,5);
    %vH=vehicle(:,8);
    %aH=vehicle(:,9);

    %vP=vehicle(:,21);
    %aP=vehicle(:,22);

    %dx=vehicle(:,23);
    dv=vehicle(indxCF,25);
```

```
%% separated opening and closing regions
indxOpening=find(dv<=-error);
indxClosing=find(dv>=error);
indxStable=find((dv>-error).*(dv<error));

%% counting events
indxEvents=NaN*ones(size(t));

if ~isempty(indxClosing) %closing
   indxEvents(indxClosing)=-1;

end

if ~isempty(indxOpening)%opening
   indxEvents(indxOpening)=1;

end

if ~isempty(indxStable)% stable
   indxEvents(indxStable)=0;

end

indx01=[1;indxEvents(1:end-1)~=indxEvents(2:end)];
indx02=[indxEvents(1:end-1)~=indxEvents(2:end);1];
indx03=find(indx01); %start
indx04=find(indx02); %end
duration=indx04-indx03+1;

if ~isempty(indx03)
   typeEvents=indxEvents(indx03);

end

indx05=typeEvents==0;
indx06=duration<(minDurEv(1)*10);

indx07=find(indx05.*indx06);

if ~isempty(indx07)
   if indx07(1)>1
      for k=(indx07)'
         indx04(k-1)=indx04(k);
         duration(k-1)=duration(k-1)+duration(k);

      end
      indx03(indx07)=[];
      indx04(indx07)=[];
      duration(indx07)=[];
      typeEvents(indx07)=[];
```

```
    else
        indx07(1)=[];
        for k=(indx07)'
            indx04(k-1)=indx04(k);
            duration(k-1)=duration(k-1)+duration(k);

        end
        indx03(indx07)=[];
        indx04(indx07)=[];
        duration(indx07)=[];
        typeEvents(indx07)=[];

    end

end

indx08A=typeEvents~=0;
indx08B=~isempty(typeEvents);
indx08=indx08A.*indx08B;
indx09=duration<(minDurEv(2)*10);

indx10=find(indx08.*indx09);

if ~isempty(indx10)
    if indx10(1)>1
        if indx04(indx10(end))<length(t)
            for k=(indx10)'
                indx04(k-1)=indx04(k);
                duration(k-1)=duration(k-1)+duration(k);


            end
            indx03(indx10)=[];
            indx04(indx10)=[];
            duration(indx10)=[];

            typeEvents(indx10)=[];

        else
            indx10(end)=[];
            for k=(indx10)'

                indx04(k-1)=indx04(k);
                duration(k-1)=duration(k-1)+duration(k);

            end
            indx03(indx10)=[];
            indx04(indx10)=[];
            duration(indx10)=[];
            typeEvents(indx10)=[];

        end
```

```
    else
        indx10(1)=[];
        if ~isempty(indx10)
            if indx04(indx10(end))<length(t)
                for k=(indx10)'
                    indx04(k-1)=indx04(k);
                    duration(k-1)=duration(k-1)+duration(k);

                end
                indx03(indx10)=[];
                indx04(indx10)=[];
                duration(indx10)=[];
                typeEvents(indx10)=[];

            else
                indx10(end)=[];
                for k=(indx10)'
                    indx04(k-1)=indx04(k);
                    duration(k-1)=duration(k-1)+duration(k);

                end
                indx03(indx10)=[];
                indx04(indx10)=[];
                duration(indx10)=[];
                typeEvents(indx10)=[];

            end

        end

    end

end

totalEvnts=size(typeEvents,1);
indxEvents=NaN*ones(size(t));

%% reconstruction
for k=1:totalEvnts
    indx11=(indx03(k):indx04(k))';
    indxEvents(indx11)=typeEvents(k);

end

eventTable=[(1:totalEvnts)',typeEvents, duration/10,indx03, indx04];
indx12=find(isnan(eventTable(:,2)));

if ~isempty(indx12)
    eventTable(indx12,:)=[];

end
```

```
    totalEvnts=size(eventTable,1);
    matrix(23)=totalEvnts;
    vehicle(indxCF,63)=indxEvents;

else
    eventTable=[];
    matrix(23)=0;
    vehicle(:,63)=NaN*ones(size(vehicle(:,5),1),1);


end
```

A - 28. Recognizing process algorithm

```
function [processes, processPeakDataAccel, varargout]=recognizProcess4PA00(smthAccel, delAccel)
%% Syntax
%       [processes, processPeakDataAccel]=recognizProcess4PA00(smthAccel, delAccel)
%       [processes, processPeakDataAccel, indxingProcess]=recognizProcess4PA00(smthAccel, delAccel)
%
%% Arguements
%       INPUTS:
%           smoothedAccel: a n-by-1 column vector including smoothed
%           acceleration in feet/sec^2.
%           delAccel: a n-by-1 column vector including smoothed variation
%           of acceleration in feet/sec^2.
%
%       OUTPUTS:
%           processes:
%               a p-by-5 matrix including information of p processes of a vehicle. the p is number of processes
of a vehicle.
%               processes=[process identify number,type of process, length of process in # of frames, index of
start time of a processi frame id, index of end time of a process in fram id];
%                       types of processes: -2=braking process,
%                       -1=recoveryB that is a recovery process from
%                       braking process,
%                       0 = constant speed process,
%                       1=recoveryA that is a recovey process from
%                       accelerating process, and
%                       2=accelerating process.
%           processPeakDataAccel:
%               n-by-8 matrix: peak signal data
%                   column 1:   peak signal of start point of an
%                   accelerating process=1.
%                   column 2:   peak signal of start point of an braking
%                   process =1.
%                   column 3:   peak signal of start point of an recovery A
%                   process =1.
%                   column 4:   peak signal of start point of an recovery B
%                   process =1.
%                   column 5:   peak signal of start point of constant
%                   speed process =1.
%                   column 6:   peak signal of start point of an controlled
%                   process; 1=accelerating process and -1=braking process.
%                   column 7:   peak signal of start point of an controlled
%                   process; 1=recovery B process and -1=recovery A process.
```

```
%               column 8:   peak signal of start point of all
%               processes.
%          indxingProcess: a 1-by-5 cell variable including indices of each
%          processes: accereating process, braking process, recoveryA
%          process, recoveryB process and constant speed process.
%          (optional)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% recognizing processes
processIndx=zeros(size(delAccel));

% accelerating processes
indx1=smthAccel>0;
indx2=delAccel>=0;

accelerating=find(indx1.*indx2);

if ~isempty(accelerating)
   processIndx(accelerating)=2;

end

% recovery A processes
indx1=smthAccel>=0;
indx2=delAccel<0;

recoveryA=find(indx1.*indx2);

if ~isempty(recoveryA)
   processIndx(recoveryA)=1;

end

% braking processes
indx1=smthAccel<0;
indx2=delAccel<=0;

braking=find(indx1.*indx2);

if ~isempty(braking)
   processIndx(braking)=-2;

end

% recovery B processes
indx1=smthAccel<=0;
indx2=delAccel>0;

recoveryB=find(indx1.*indx2);

if ~isempty(recoveryB)
   processIndx(recoveryB)=-1;
```

```
end

% constant speed processes
indx1=smthAccel==0;
indx2=delAccel==0;

constSpeed=find(indx1.*indx2);

if ~isempty(constSpeed)
    processIndx(constSpeed)=0;

end

%% set processes variable
indx1=processIndx(1:end-1)~=processIndx(2:end);
indx2=[1;indx1];

% number of processes
numProcess=sum(indx2);

% first column of processes
processID=(1:numProcess)';

% forth column of processes
startIndx=find(indx2);

% third column of processes
lengthProcess=[startIndx(2:end);length(delAccel)+1]-startIndx;

% fifth column of processes
endIndx=startIndx+lengthProcess-1;

% second column of processes
processType=processIndx(startIndx);

% set processes variable
processes=[processID,processType,lengthProcess,startIndx,endIndx];

%% set processPeakDataAccel
processPeakDataAccel=zeros(size(delAccel,1),1);

peakIndx01=indx2.*processIndx;

indxAccelerating=find(peakIndx01==2);
indxBrakting=find(peakIndx01==-2);
indxRecoveryA=find(peakIndx01==1);
indxRecoveryB=find(peakIndx01==-1);
indxConstSpeed=find(peakIndx01==-1);

if ~isempty(indxAccelerating) % accelerating processes
    processPeakDataAccel(indxAccelerating,1)=1;
```

```
end

if ~isempty(indxBrakting) % breaking processes
   processPeakDataAccel(indxBrakting,2)=1;

end

if ~isempty(indxRecoveryA) % recovery A processes
   processPeakDataAccel(indxRecoveryA,3)=1;

end

if ~isempty(indxRecoveryB) % recovery A processes
   processPeakDataAccel(indxRecoveryB,4)=1;

end

if ~isempty(indxConstSpeed) % recovery A processes
   processPeakDataAccel(indxConstSpeed,5)=1;

end

%% controlled processes: 1=accelerating, -1= braking
processPeakDataAccel(:,6)=processPeakDataAccel(:,1)-processPeakDataAccel(:,2);

%% non-controlled processes: 1=recovery B, -1=recoveryA
processPeakDataAccel(:,7)=processPeakDataAccel(:,4)-processPeakDataAccel(:,3);

%% all processes
processPeakDataAccel(:,8)=sum(processPeakDataAccel,2);


%% set process index
if nargout>2
   varargout{1} =processIndx;

end




%% set indxingProcess
if nargout>3
   indxingProcess=cell(1,5);
   indxingProcess{1}=accelerating;
   indxingProcess{2}=braking;
   indxingProcess{3}=recoveryA;
   indxingProcess{4}=recoveryB;
   indxingProcess{5}=constSpeed;

   varargout{2} =indxingProcess;
```

```
end


function [vehicle2, processes, avgOprAccels, counting,  indxingProcess]=processAnalysis00(vehicle)
%% Syntax
%       [processes, operatingAcccel, processPeakDataAccel, reporting,
indxingProcess]=processAnalysis00(vehicle)
%
%% Arguements
%      INPUTS:
%          vehicle: a n-by-36 matrix including vehicle information from
%          indiDB.
%
%      OUTPUTS:
%          vehicle2: a n-by-48 matrix including vehicle information for
%          indiDB2.
%          processes:
%              a p-by-5 matrix including information of p processes of a vehicle. the p is number of processes
of a vehicle.
%              processes=[process identify number,type of process, length of process in # of frames, index of
start time of a processi frame id, index of end time of a process in fram id];
%                    types of processes: -2=braking process,
%                    -1=recoveryB that is a recovery process from
%                    braking process,
%                    0 = constant speed process,
%                    1=recoveryA that is a recovey process from
%                    accelerating process, and
%                    2=accelerating process.
%          avgOprAccels:
%              1-by-2 vector including average operating acceleration in
%              feet/sec^2.
%              =[average operating accelerating, average operating
%              braking];
%         counting:
%              3-by-8 vector including number of processes
%                 rows:
%                    row1: total number of processes.
%                    row2: average number of processes per minutes.
%                    row3: average number of processes per minutes.
%                 columns:
%                    column1: accelerating processes.
%                    column2: braking processes.
%                    column3: recovery A processes.
%                    column4: recover B processes.
%                    column5: constant speed processes.
%                    column6: controlled processes.
%                    column7: recovery processes.
%                    column8: all processes
%          indxingProcess: a 1-by-5 cell variable including indices of each
%          processes: accereating process, braking process, recoveryA
%          process, recoveryB process and constant speed process.
%          (optional)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% initialization
```

```
accel=vehicle(:,9);

%% smoothing
[smthAccel, delAccel]=smoothingAlgrth4RcgnzdPrss00(accel, 0.3, 0.3);

%% recognizing processes
[processes, processPeakDataAccel, processIndx, indxingProcess]=recognizProcess4PA00(smthAccel,
delAccel);

%% set operating Acceleration
[processes, operatingAccel]=determineOprtAccel(smthAccel, processes);

%% set average operating accelrations
avgOprAccels=avgOprtAccel4PA(operatingAccel, processPeakDataAccel);

%% set counting
counting=countingProcesses4PA(processPeakDataAccel);

vehicle2.data=[vehicle, smthAccel, delAccel,processIndx, processPeakDataAccel,operatingAccel];
vehicle2.heads={'smoothed Acceleration'; 'variation of acceleration'; 'process type'; 'peak signal of start
point of an accelerating process';...
    'peak signal of start point of an braking process'; 'peak signal of start point of an recovery A process';...
    'peak signal of start point of an recovery B process'; 'peak signal of start point of constant speed
process';...
    'peak signal of start point of an controlled accelerating process and -1=braking process'; ...
    'peak signal of start point of an controlled process; 1=recovery B process and -1=recovery A process';...
    'peak signal of start point of all processes'; 'operating acclerations '};
```

A - 29. K-means Clustering Algorithm

```
function [clusterIndx, C, sumd, D, s, numVehs]=kMeansClustering(X, k, varargin)
%% clustering analysis function

%% inputs
% X=input data
% k= number of clusters

%% outputs
%%% clusterIndx:    n-by-1 vector including cluster indices of each point.
%%% C :     k-by-p matrix including the k cluster centroid locations
%%% sumd:   1-by-k vector including the within-cluster sums of
%%%             point-to-centroid
%%% D :     n-by-k matrix including distances from each point to every
%%% centroid.
%%% s:      n-by-1 vector including silhouette values for each point
%%% numVehs: the clustering analysis table including 3 rows and k+1 columns:
%%% row 1: # of vehicles of each cluster and entire data
%%% row 2: # of vehicles with negative silhouette value
%%% row 3: average silhouette value
%%% column k+1: total value

%% initialization
optionNames={'silhouettePlot';...
    'distance';...
```

```
    'replicates';...
    'display';...
    'emptyaction';...
    'startOption'};
optionValues={'off';...
    'sqEuclidean';...
    30;...
    'off';...
    'error';...
    'sample'};

if nargin>2
    if ~rem(nargin-2,2)
        m=(nargin-2)/2;

        for q=1:m
            parameters=varargin{2*q-1};

            indx=find(strcmpi(optionNames,parameters));

            if ~isempty(indx)
                optionValues{indx}=varargin{2*q};

            end

        end

    else
        disp('You must assign options'' value with option'' name!');

    end

end



%% Clustering
[clusterIndx, C, sumd, D]=kmeans(X,k, 'distance',optionValues{2},'replicates', optionValues{3}, 'display', optionValues{4},...
    'emptyaction', optionValues{5}, 'start', optionValues{6});


%% calculate Silhouette value
if strcmpi(optionValues{2},'off')
    s=silhouette(X,clusterIndx);

else
    figure;
    [s, h]=silhouette(X,clusterIndx);
    set(get(gca,'children'),'FaceColor', [.8 .8 1]);
    xlabel('Silouette Value','fontsize', 16);
    ylabel('Cluster ID', 'fontsize', 16);
```

```
end

grid on

numVehs=NaN*ones(3,k+1); % 1st row: num Vehicles; 2nd row: num vehicles having negative silhouettte
value
numVehs(1,k+1)=numel(s);
numVehs(3,k+1)=nanmean(s);

for m=1:k
    indx01=find(clusterIndx==m);
    if ~isempty(indx01)
        numVehs(1,m)=numel(indx01);
        indx02=find(s(indx01)<0);

        if ~isempty(indx02)
            numVehs(2,m)=numel(indx02);

        else
            numVehs(2,m)=0;

        end

        numVehs(3,m)=nanmean(s(indx01));

    else
        numVehs(1,m)=0;
        numVehs(2,m)=0;
        numVehs(3,m)=0;

    end

end

numVehs(2,k+1)=sum(numVehs(2,1:k),2);

function [inputs, varargout]=clusteringVehiclesInputs003(DB)

data=DB.matrix;
dataBase=DB.data;

numVeh=size(DB.matrix,2);

dataSets=zeros(numVeh,13);
subdata=zeros(numVeh,8);

for k=1:numVeh
    %% 1. Avg. Speed
    speed=dataBase{1,k}(:,8);

    dataSets(k,1)=nanmean(speed);
```

```
    %% 2. std. Speed
    dataSets(k,2)=nanstd(speed);

    %% 3. avg target operating acceleration
    toAccTable=dataBase{2,k};
    indx01=find(toAccTable(:,2)~=2); % find process as no accelerating

    if ~isempty(indx01)
        toAccTable(indx01,:)=[];

    end

    toAcc=toAccTable(:,6);
    dataSets(k,3)=nanmean(toAcc);

    %dataSets(k,3)=dataBase{3,k}(1);

    %% 4. std target operating Acceleration
    dataSets(k,4)=nanstd(toAcc);

    %% 5. number of lane-changing
    laneIDs=dataBase{1,k}(:,10);
    indx01=laneIDs(1:end-1)~=laneIDs(2:end);
    dataSets(k,5)=sum(indx01);

    %% 10. average time headway
    timeHW=dataBase{1,k}(:,24);

    dataSets(k,10)=nanmean(timeHW);

    %% 11. std. time headway
    dataSets(k,11)= nanstd(timeHW);

    %% fixing matrix for time headway
    subdata(k,1:4)=[nanmean(timeHW), nanstd(timeHW), nanmax(timeHW), nanmin(timeHW)];

    %% 12. average relative speed
    relSp=dataBase{1,k}(:,25);

    dataSets(k,12)=nanmean(relSp);
    %% 13. std relative speed
    dataSets(k,13)=nanstd(relSp);

    %% fixing matrix for relative speed
    subdata(k,5:8)=[nanmean(relSp), nanstd(relSp), nanmax(relSp), nanmin(relSp)];

end

%% 3.  others
% 6. vehicle type, 7. rate of car-following in time, 8. avg. spacing
% 9. std. spacing,
dataSets(:,6:9)=(data(6:9,:))';
```

```
if nargout>1
    DB.matrix(5,:)=(dataSets(:,3))';
    DB.matrix(12:19,:)=(subdata)';

    varargout{1}=DB;

end

inputs.head={'avg. speed in ft/s';...
    'std. speed in ft/s';...
    'avg. target operating acceleration in ft/s^2';...
    'std. target operating acceleration in ft/s^2';...
    'number of lane-changing';...
    'vehicle type';...
    'rate of car-following in time';...
    'avg. spacing in ft';...
    'std. spacing in ft';...
    'avg. time headway in sec.';...
    'std. time headway in sec.';...
    'avg. relative speed in ft/s';...
    'std. relative speed in ft/s'};
inputs.data=dataSets;

function [inputs, varargout]=clusteringVehiclesInputs001(DB)

data=DB.matrix;
dataBase=DB.data;

numVeh=size(DB.matrix,2);

dataSets=zeros(numVeh,8);
subdata=zeros(numVeh,8);

for k=1:numVeh
    %% 1. Avg. Speed
    speed=dataBase{1,k}(:,8);

    dataSets(k,1)=nanmean(speed);

    %% 2. avg target operating acceleration
    dataSets(k,2)=dataBase{3,k}(1);

    %% 3. number of lane-changing
    laneIDs=dataBase{1,k}(:,10);
    indx01=laneIDs(1:end-1)~=laneIDs(2:end);
    dataSets(k,3)=sum(indx01);

    %% 4. average time headway
    timeHW=dataBase{1,k}(:,24);

    dataSets(k,7)=nanmean(timeHW);
    subdata(k,1:4)=[nanmean(timeHW), nanstd(timeHW), nanmax(timeHW), nanmin(timeHW)];
```

```
    %% 8. average relative speed
    relSp=dataBase{1,k}(:,25);

    dataSets(k,8)=nanmean(relSp);
    subdata(k,5:8)=[nanmean(relSp), nanstd(relSp), nanmax(relSp), nanmin(relSp)];

end

%% 3.  others
dataSets(:,4:6)=(data([6 7 8 ],:))';

inputs.head={'avg. speed in ft/s';...
    'avg. target operating acceleration in ft/s^2';...
    'number of lane-changing';...
    'vehicle type';...
    'rate of car-following in time';...
    'avg. spacing in ft';...
    'avg. time headway in sec.';...
    'avg. relative speed in ft/s';};
inputs.data=dataSets;

if nargout>1
    DB.matrix(5,:)=(dataSets(:,3))';
    DB.matrix(12:19,:)=(subdata)';

    varargout{1}=DB;

end

function [inputs, varargout]=clusteringVehiclesInputs002(DB)

data=DB.matrix;
dataBase=DB.data;

numVeh=size(DB.matrix,2);

dataSets=zeros(numVeh,13);
subdata=zeros(numVeh,8);

for k=1:numVeh
    %% 1. Avg. Speed
    speed=dataBase{1,k}(:,8);

    dataSets(k,1)=nanmean(speed);

    %% 2. std. Speed
    dataSets(k,2)=nanstd(speed);

    %% 3. avg target operating acceleration
    toAccTable=dataBase{2,k};
    indx01=find(toAccTable(:,2)~=2); % find process as no accelerating
```

```
    if ~isempty(indx01)
        toAccTable(indx01,:)=[];

    end

    toAcc=toAccTable(:,6);
    dataSets(k,3)=nanmean(toAcc);

    %dataSets(k,3)=dataBase{3,k}(1);

    %% 4. std target operating Acceleration
    dataSets(k,4)=nanstd(toAcc);

    %% 5. number of lane-changing
    laneIDs=dataBase{1,k}(:,10);
    indx01=laneIDs(1:end-1)~=laneIDs(2:end);
    dataSets(k,5)=sum(indx01);

    %% 10. average time headway
    timeHW=dataBase{1,k}(:,24);

    dataSets(k,10)=nanmean(timeHW);

    %% 11. average time headway
    dataSets(k,11)= nanstd(timeHW);

    %% fixing matrix for time headway
    subdata(k,1:4)=[nanmean(timeHW), nanstd(timeHW), nanmax(timeHW), nanmin(timeHW)];

    %% 12. average relative speed
    relSp=dataBase{1,k}(:,25);

    dataSets(k,12)=nanmean(relSp);
    %% 13. std relative speed
    dataSets(k,13)=nanstd(relSp);

    %% fixing matrix for relative speed
    subdata(k,5:8)=[nanmean(relSp), nanstd(relSp), nanmax(relSp), nanmin(relSp)];

end

%% 3.  others
% 6. vehicle type, 7. rate ofcar-following in time, 8. avg. spacing
% 9. std. spacing,
dataSets(:,6:9)=(data(6:9,:))';

inputs.head={'avg. speed in ft/s';...
    'std. speed in ft/s';...
    'avg. target operating acceleration in ft/s^2';...
    'std. target operating acceleration in ft/s^2';...
    'number of lane-changing';...
    'vehicle type';...
```

```
    'rate of car-following in time';...
    'avg. spacing in ft';...
    'std. spacing in ft';...
    'avg. time headway in sec.';...
    'std. time headway in sec.';...
    'avg. relative speed in ft/s';...
    'std. relative speed in ft/s'};
inputs.data=dataSets;

if nargout>1
    DB.matrix(5,:)=(dataSets(:,3))';
    DB.matrix(12:19,:)=(subdata)';

    varargout{1}=DB;

end
```

A - 30. Factor Analysis

```
%% initialization
Data=indiDB7.matrix([25 26 27 29 28],:);
m=2;
rotate='promax';
scores='wls';
variablesName=[];

% creating a n-by-d matrix of n observatins of d variables
X=Data';

% Factor Analysis
[loadings, psi, T, stats, F]=factoran(X,m, 'scores', scores, 'rotate', rotate);


% ploting each stock using its factor loadings as coordinates
if isempty(variablesName)
    variablesName=num2str((1:size(loadings,1))');

end
figure;
biplot(loadings, 'varlabels', variablesName);
axis square
view(155, 27);

reportingFactorAnalysis.projectName='Factor Analysis of Emissions and Fuel Consumption';
reportingFactorAnalysis.methods.Xtype='raw';

if strcmpi(reportingFactorAnalysis.methods.Xtype,'raw')
    reportingFactorAnalysis.methods.scores=scores;

else
     reportingFactorAnalysis.methods.scores='none';
```

```
end

reportingFactorAnalysis.methods.rotate=rotate;

reportingFactorAnalysis.variables.heads=variablesName;
reportingFactorAnalysis.variables.data=X;

reportingFactorAnalysis.factors.N=m;

reportingFactorAnalysis.results.Communality=loadings;
reportingFactorAnalysis.results.SpecificVar=psi;
reportingFactorAnalysis.results.rotationMatrix=T;
reportingFactorAnalysis.results.MaxLogLikehoodVal=stats.loglike;
reportingFactorAnalysis.results.ErrorDOF=stats.dfe;

try
   reportingFactorAnalysis.results.NullHypothesis.Chi_SquaredStatistic=stats.chisq;
   reportingFactorAnalysis.results.NullHypothesis.RightTailSigLevel=stats.p;

catch
   reportingFactorAnalysis.results.NullHypothesis.Chi_SquaredStatistic=[];
   reportingFactorAnalysis.results.NullHypothesis.RightTailSigLevel=[];

end

reportingFactorAnalysis.results.predictionOfCommonFactor=F;


x=F(:,1);
y=F(:,2);

figure;
plot(x,y,'*');

xlabel('factor 1');
ylabel('factor 2');
grid on


reportingFAforEmiss01=reportingFactorAnalysis;

clear Data m rotate variablesName loadings psi T stats F a b s scores x y X
```

A - 31. determining target operating accelerations

```
function [processes, varargout]=determineOprtAccel(smthAccel, processes)
%% Syntax
%      processes2=determineOprtAccel(smthAccel, processes1)
%      [processes2, operatingAccel]=determineOprtAccel(smthAccel, processes1)
%
%% Arguements
%      INPUTS:
%          smoothedAccel: a n-by-1 column vector including smoothed
```

```
%          acceleration in feet/sec^2.
%
%          processes1:
%              a p-by-5 matrix including information of p processes of a vehicle. the p is number of processes
of a vehicle.
%              processes1=[process identify number,type of process, length of process in # of frames, index
of start time of a processi frame id, index of end time of a process in fram id];
%                      types of processes: -2=braking process,
%                      -1=recoveryB that is a recovery process from
%                      braking process,
%                      0 = constant speed process,
%                      1=recoveryA that is a recovey process from
%                      accelerating process, and
%                      2=accelerating process.
%
%      OUTPUTS:
%          processes2:
%              a p-by-6 matrix including information of p processes of a vehicle. the p is number of processes
of a vehicle.
%              processes=[process identify number,type of process, length of process in # of frames, index of
start time of a processi frame id, index of end time of a process in fram id, operating accelerations];
%                      types of processes: -2=braking process,
%                      -1=recoveryB that is a recovery process from
%                      braking process,
%                      0 = constant speed process,
%                      1=recoveryA that is a recovey process from
%                      accelerating process, and
%                      2=accelerating process.
%          operatingAccel:
%              a n-by-1 column vector including operating accleration in
%              feet/sec^2. (optional)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

operatingAcc=zeros(size(processes,1),1);

if nargout==2
   operatingAccel=zeros(size(smthAccel));

end

processes1=processes;

k=1;
while ~isempty(processes)
   process=processes(1,:);
   processes(1,:)=[];

   % operating acceleration of accelerating processes
   if process(2)==2
      operatingAcc(k)=max(smthAccel(process(4):process(5)));

   end
```

```
    % operating acceleration of braking processes
    if process(2)==-2
        operatingAcc(k)=min(smthAccel(process(4):process(5)));

    end

    if nargout==2
        operatingAccel(process(4):process(5))=operatingAcc(k);

    end

    k=k+1;

end

processes=[processes1,operatingAcc];

if nargout==2
    varargout{1}=operatingAccel;

end
```

A - 32. counting processes

```
 function counting=countingProcesses4PA(processPeakDataAccel, varargin)
%% Syntax
%       counting=countingProcesses4PA(processPeakDataAccel)
%       counting=countingProcesses4PA(processPeakDataAccel, indx)
%
%% Arguements
%      INPUTS:
%          processPeakDataAccel:
%              n-by-8 matrix: peak signal data
%                  column 1:   peak signal of start point of an
%                  accelerating process=1.
%                  column 2:   peak signal of start point of an braking
%                  process =1.
%                  column 3:   peak signal of start point of an recovery A
%                  process =1.
%                  column 4:   peak signal of start point of an recovery B
%                  process =1.
%                  column 5:   peak signal of start point of constant
%                  speed process =1.
%                  column 6:   peak signal of start point of an controlled
%                  process; 1=accelerating process and -1=braking process.
%                  column 7:   peak signal of start point of an controlled
%                  process; 1=recovery B process and -1=recovery A process.
%                  column 8:   peak signal of start point of all
%                  processes.
%          indx:
%              m-by-1 vector including index indicating regions that you
%              want to analyze. (optional)
%
%      OUTPUTS:
```

```
%        counting:
%            3-by-8 vector including number of processes
%               rows:
%                  row1: total number of processes.
%                  row2: average number of processes per minutes.
%                  row3: average number of processes per minutes.
%               columns:
%                  column1: accelerating processes.
%                  column2: braking processes.
%                  column3: recovery A processes.
%                  column4: recover B processes.
%                  column5: constant speed processes.
%                  column6: controlled processes.
%                  column7: recovery processes.
%                  column8: all processes
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if nargin==1
   %% initialization
   totalTime=size(processPeakDataAccel,1)/10;

   %% counting processes
   counting1=sum(abs(processPeakDataAccel),1);

else

   indx=varargin{1};
   if ~isempty(indx)
      totalTime=size(indx,1)/10;

      counting1=sum(processPeakDataAccel,1);

   else
      counting1=zeros(1,8);

   end


end

counting2=counting1/totalTime*60; %# of processes/min
counting3=counting1/totalTime*3600; %# of processes/ hour

counting=[counting1;counting2; counting3];
```