



ROYAL INSTITUTE
OF TECHNOLOGY

Driver Modeling based on Computational Intelligence Approaches

– Exploration and Modeling of Driver-following Data collected using
an Instrumented Vehicle

BY

XIAOLIANG MA

DISSERTATION

in simulation modeling of transport systems
in partial fulfillment of the requirements for
the degree of *Doctor of Philosophy*

Royal Institute of Technology

Stockholm, Sweden 2006

Copyright © 2006 by Xiaoliang Ma

Driver Modeling based on Computational Intelligence Approaches

Center for Traffic Simulation Research
Department of Transport and Economics
Royal Institute of Technology (KTH)
Teknikringen 72
SE-100 44 Stockholm
Sweden

Tel. +46 8 7908426, Fax. +46 8 7908461

TRITA-TEC-PHD 06-004
ISSN 1653-4468
ISBN 13: 978-91-85539-11-6
ISBN 10: 91-85539-11-2

Preface

This thesis is concerned with modeling of driver behavior based on data collected from real traffic using an advanced instrumented vehicle. In particular, the focus is on driver-following behavior (often called car-following in transport science) for microscopic simulation of road traffic systems. In addition, the modeling methodology developed can be applied for the design of human-centered control algorithms in adaptive cruise control (ACC) and other longitudinal active-safety technologies.

Driver behavior is a constant research topic in the modeling of traffic systems and Intelligent Transportation Systems (ITS), which could be traced back to the work of General Motor (GM) Co. in 1950's. In the early time, researchers were only interested in the development of driver models fulfilling basic physical properties and producing reasonable flow dynamics on a macroscopic level. With the booming interest on driver modeling on a microscopic level and needs in ITS developments, researchers now emphasize modeling using microscopic data acquired from real world. To follow this research trend, a methodological framework on car-following data acquisition, analysis and modeling has been developed step by step in this thesis, and the basic idea is to build a computational model for car-following behavior by exploration of collected data. To carry out the work, different techniques within the field of modern Artificial Intelligence (AI), namely Computational Intelligence (CI)¹, have been applied in the research subtasks e.g. information estimation, behavioral regime classification, regime model integration and model estimation. Therefore, a preliminary introduction of the CI methods being used in this thesis work is included in the text.

¹CI is an internationally well-established scientific field with its research core focused on fuzzy systems, neural and evolutionary computing and a large number of numerical methods borrowed from various scientific and engineering fields, all of which are essential for building individual models and complex systems in intelligent machine design and imitation of human level intelligence. There are many journals that provide in-depth research contributions and conferences where the most up-to-date research can be found. An introductory website can be http://en.wikipedia.org/wiki/Computational_Intelligence.

Following data collection on selected Swedish motorways, the behavioral information of drivers was estimated using Kalman Smoothing algorithm based on a state-space model of the multivariate data series. This preprocessing work canceled measurement noise and provided more accurate and consistent car-following data for further analysis and modeling. Spectrum analysis was first conducted on the smoothed data in order to investigate driver properties and to estimate their reaction delay. The estimated results were further applied in the model estimation procedure. From the car-following data, it was not difficult to find that the general driver behavior showed apparent nonlinearity and stage adaptivity. This prompted us to resort to clustering methods and to classify data into behavioral regimes and then analyze and model statistical relations between perceptual variables and the acceleration output in each regime. In order to integrate the models in all regimes, a fuzzy inference framework is developed as a variant of the Sugeno fuzzy inference system. A neural fuzzy system is formulated as an extension of the idea of the adaptive neural-fuzzy inference system (ANFIS) in order to provide the computational model with ability to learn from real data.

In addition to the model structure determination, an important issue in the driver modeling research is the estimation method of driver-following models using acquired data. Traditionally, it is widely applied to use the least square deviation between real and modeled acceleration outputs as the performance measure and then search the parameter space based on derivative-based methods or genetic algorithm (GA) using observed model inputs. Proposed in the thesis is a dynamic estimation approach based on general physical states of the driver-vehicle unit. Kalman-Filting based algorithms including Extended Kalman-Filter (EKF) and Unscented Kalman-Filter (UKF) have been utilized for the implementation of this model calibration methodology. Furthermore, the model estimation methods have been evaluated based on real data and compared with the traditional approach using gradient-based methods or naive GA algorithm. The research subject and result offer a promising perspective for the improvement of simulation authenticity.

Acknowledgements

This thesis summarizes the research work conducted, during the past four years, in the Center for Traffic Simulation Research (CTR) at KTH. It includes two projects commissioned by the Swedish Road Authority (SRA) and Swedish Innovation Systems Authority (VINNOVA): SIMLAB, concerned with improvement, innovation and integration of microsimulation models; and EVITA, aimed at prediction of the safety effect of Intelligent Speed Adaptation (ISA) using simulation methods.

I would like first and foremost to express my sincere gratitude to my supervisors, professors Karl-Lennart Bång and Ingmar Andréasson. Professor Bång gave me valuable advice on how to formulate research method in transport science, and his broad mind and warm heart helped me break through several thresholds in my Ph.D study. Professor Andréasson guided me into the research of simulation modeling in transport science, and encouraged me to study and apply knowledge from applied mathematics and computer science. I am very grateful to him for his high patience and constant encouragement, and being ready to help whenever I needed in the uneven road of my Ph.D research.

I particularly appreciate the faculties in the Dept. of Numerical Analysis and Computer Science (NADA) as well as the Dept. of Signal, Sensor and Systems (S3) for providing many wonderful courses in computer science, mathematical modeling and signal processing, which consistently consolidate my knowledge and give me inspirations in my own research topics. Among them, I specially thank Dr. Frank Hoffmann, (now professor in the Dept. of Computer Science, University of Dortmund) who initiated my interest in the topic of computational intelligence with his inspiring instruction during the courses of Machine Learning and Artificial Neural Networks.

Special thanks to Mark Dougherty, professor at Dept. of Computer Science at Dalarna University for being my doctoral dissertation opponent and to Pontus Matstoms, chief

researcher at the Division of Transport and User Behavior, Swedish National Road Transport Institute (VTI) for being my final seminar opponent.

“Tack så mycket” to all my colleagues in CTR and TOL. Dr. Leonid Engelson deserves my appreciation for many helpful discussions during the EVITA project and my teaching assistant work, from which I learned a lot; A big thank to my “officemates”, Wilco, Chiguma and Ryan for creating an international and active communication environment and for their friendship and kindness during the years of sharing our “big” office; many thanks to Katarina, Lennart and Brigitt for helping me figure out many practical issues in our institution; I also owe debt to Stefan, who helped me in my data collection using the instrumented car.

I owe many thanks to my friends all of whom make my leisure time here so enjoyable. In particular, I am indebted to Andy, who designed and drew the cover picture for me, very impressive creativity. Special thanks to all friends previously or currently in NADA and S3 for sharing study and fun time together.

Last but not least, I would deeply thank my mum, Yiwan, and my brother, Ren, for all their wishes, cares, supports and loves; without them, I could hardly pursue my doctoral degree and obtain the current achievement in my academic route. And finally, to Aihua, my dear wife, my very special thanks for all your understanding and love during these years in Sweden: together with you, my sky of the day is always bright, my sky of the night is always sparkling.

Xiaoliang Ma
Stockholm, Nov. 2006

List of papers

This thesis is mainly centered on the following selected research papers being produced within the Ph.D period:

- I. X. Ma and I. Andréasson, Behavior measurement, analysis and regime classification in car-following, *IEEE Transactions on Intelligent Transportation Systems*, accepted for publication in July, 2006;
- II. X. Ma and I. Andréasson, Driver reaction delay estimation from real data and its application in GM-type model evaluation, *Transportation Research Record - Journal of Transportation Research Board*, accepted for publication in March, 2006;
- III. X. Ma and I. Andréasson, Statistical analysis of driver behavioral data in different regimes of the car-following stage, *Transportation Research Record - Journal of Transportation Research Board*, under revision, 2006;
- IV. X. Ma and M. Jansson, A general Kalman-filter based model estimation method for car-following dynamics in traffic simulation, *Transportation Research C: Emerging Technology*, submitted for publication, 2006;
- V. X. Ma, A computational model for driver-following behavior based on a neural-fuzzy system, *Neurocomputing*, submit for publication, 2006;
- VI. X. Ma and I. Andréasson, Predicting the effect of various ISA penetration grades on pedestrian safety by simulation, *Accident Analysis & Prevention*, Vol 37:6, 1162-1169, 2005.

Contents

1	Introduction	1
1.1	Intelligent transportation systems	1
1.1.1	Adaptive cruise control	2
1.1.2	Intelligent speed adaption	3
1.1.3	Automated highway systems	3
1.2	Simulation models of traffic systems	3
1.3	Driver behavior	5
1.3.1	Tactical driver modeling	6
1.3.2	Review of car following models	6
1.4	Identification of research needs	13
1.5	Objective	14
1.6	Research contributions	15
1.7	Thesis outline	16
2	Methodological review	17
2.1	Least square and numerical optimization	18
2.1.1	Least square methods	18
2.1.2	Derivative-based optimization	21
2.1.3	Evolutionary-based optimization	27
2.2	Kalman filtering method	28
2.2.1	Linear systems and state space models	29
2.2.2	Kalman filter algorithms	30
2.3	Artificial neural networks	33
2.3.1	Multilayer feed-forward neural networks	34

2.3.2	Radial basis function neural network	37
2.3.3	Innovative training algorithms	40
2.4	Fuzzy sets and inference principles	41
2.4.1	Fuzzy sets	41
2.4.2	Fuzzy relations and reasoning	42
2.4.3	Fuzzy modeling	44
2.5	Clustering techniques	47
2.5.1	Bayesian approaches	47
2.5.2	Fuzzy clustering	48
2.6	Summary	50
3	Research summary	51
3.1	Measurement on driver behavior	51
3.2	Information estimation	52
3.3	Driving regime analysis	56
3.4	Model estimation scheme	59
3.5	Computational model based on a neural-fuzzy system	65
3.6	Safety study on ISA	68
3.7	Synthesis and future perspectives	71
	Bibliography	75
	Papers	83

Chapter 1

Introduction

Since the appearance of the steam-powered automobile in the late eighteenth century, humans have benefited from the convenience and advantages it brought to us. Moreover, the invention with its later developments promoted constant expansion of modern industries and became a milestone in human civilization. With the technical advances of road and railroad vehicles and the later development of airplanes, modern transportation industry was formed and has become an indispensable part of global economy. Undoubtedly, due to its pervasion in our common life, vehicle-road based transportation system is still the most popular in our common life. In the last decades, the blooming of global economy and the consequent urbanization in both developed and developing countries resulted in an imbalance between transport demand and service capacity of infrastructure. Nowadays, almost all metropolitan areas in the world have congestion problems, resulting in travel delay, fuel waste, air pollution and productivity loss worth of hundred billions of dollars every year. Meanwhile, the corresponding safety situations are getting worse and worse, especially in the fast developing countries. Therefore, it is necessary for authorities to continuously revise their transportation development policies, apply new technologies and develop new management tools to treat traffic jam and travel delay problems as well as improve transportation safety.

1.1 Intelligent transportation systems

Modern computer technology has changed, and is still changing, our daily living style. From desktops to personal digital assistants (PDAs) and to new generations of cellular phones,

computing devices become ubiquitous and communication via all kinds of networks becomes more and more easy and free. Our transportation systems have also been revolutionalized with different information-based technologies: vehicle-based intelligent systems have been installed to assist the human drivers and may replace us in the near future, and the infrastructure becomes dynamic, flexible and more informative. All of these will continuously improve the efficiency and safety of the whole road-based transportation systems. Intelligent transportation systems (ITS) was nominated long time ago and can be classified into two categories: vehicle-based ITS and road-based ITS. The main idea is to equip current transportation systems with all kinds of IT technologies, mainly computing devices, in order to realize human-level intelligence and beyond among the interactions of vehicles, infrastructure and environment. The remainder of this section will focus on introducing some vehicle-based ITS systems.

1.1.1 Adaptive cruise control

Adaptive cruise control (ACC), also called intelligent cruise control (ICC), is a longitudinal active-safety oriented driving assistant system. Vehicles equipped with this kind of system can automatically follow other vehicles in the same lane and hence partial or even full automation in the longitudinal direction of the road can be realized. Thus, workload of drivers can be reduced and general safety can be improved. The design of ACC algorithms is inspired by driver-following behavior, a basic element in general driver behavior. Just like the fundamental position of car-following in the traffic flow theory, ACC is one of the most important underlying algorithms toward the ultimate goal of automated vehicles. The commercial automobiles with various ACC systems have been put on the market since many years ago, though most of the systems were still immature and with simple functions e.g. the constant pre-set speed mode and the distance-keeping mode. However, the research development of ACC has never stopped: with the technical developments of intelligent recognition of driving environment, more advanced ACC systems, e.g. stop-and-go systems and longitudinal crash avoidance systems, may be installed in cars of next generations. Judging from the trends in which airbag and anti-lock brake systems (ABS) have penetrated the automobile market, the future of ACC systems as an intermediate step toward fully automated vehicle seems optimistic.

1.1.2 Intelligent speed adaption

Unlike the ACC systems introduced above, Intelligent Speed Adaption (ISA) systems aim at improving the safety of manually driven vehicles by compelling the human drivers to adapt their speeds to the local speed limitations, which are often coded in vehicle-based GPS systems or informed directly by local intelligent infrastructure via real-time communication through public or ad hoc networks. ISA systems have been extensively tested in some of the west European countries including Sweden. Many of the research results show that they have good potential to improve traffic safety in terms of probability of accident and fatality. This thesis also includes a study of the relation between ISA penetration and safety.

1.1.3 Automated highway systems

The automated vehicle-highway system (AVHS) or (AHS) was originally designed for a different purpose from the ACC system. It is based on the assumption that fully automated control of and coordination between vehicles operating on dedicated lanes in high priority traffic corridors hold realistic promise of being the next means of substantially improving the performance and safety of the highway system. AHS offers the potential for substantial improvements in throughput, safety, travel time predictability, level of service, inclement weather operation, mobility and air quality. Technically, AHS is in essence a conceptual platform based on a collection of different kinds of vehicle and infrastructure based intelligent systems. Moreover, this concept has been explored by various research and development organizations. In principle, a fully automated highway system assumes the existence of fully dedicated highway lanes, where all vehicles running on it are fully automated like robots, with steering, brakes and throttle being controlled by computers. Like in the ACC systems, there are still many technical barriers hurdling the real implementation of such systems and computer simulation is one means to evaluate and further develop AHS systems.

1.2 Simulation models of traffic systems

Simulation is a technique to imitate operations of various kinds of systems in real world by a set of abstracts and assumptions on the systems, which are called models. It is often considered as a foot of a stool in science and engineering with both theory and experiment, and each of them helps us understand the others. In particular, theory of complex systems,

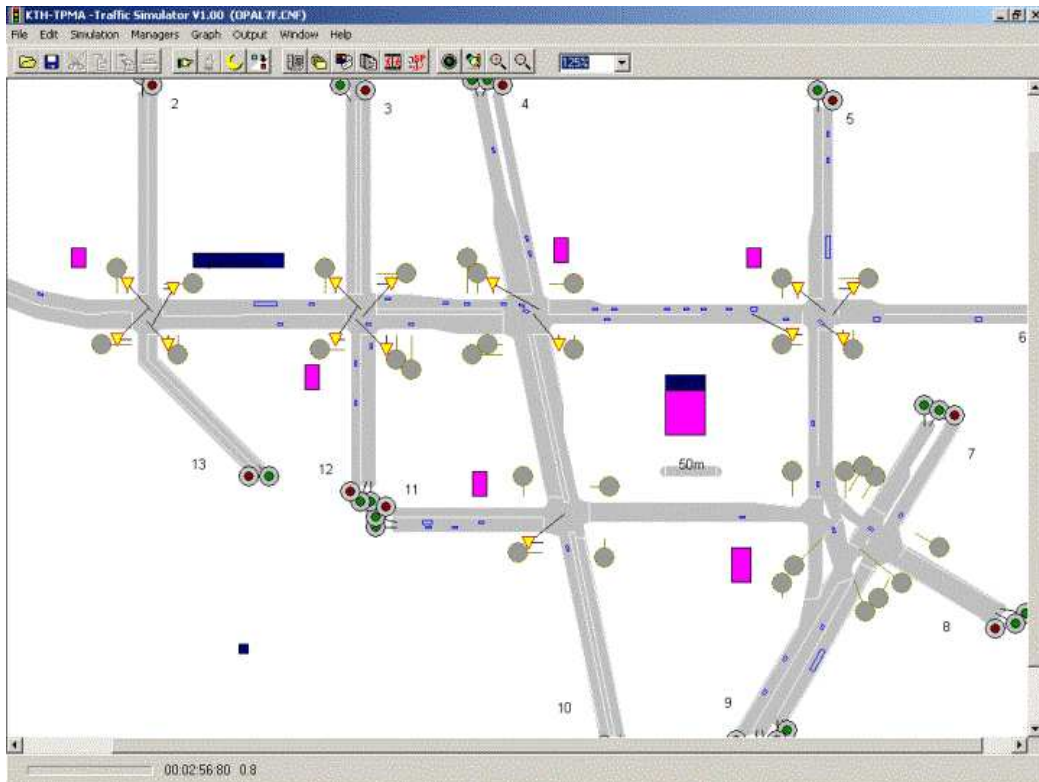


Figure 1.1: The interface of KTH-TPMA traffic simulator.

considered as a simplified representation or model of the complicated physical universe, can be tested by simulation.

Two types of models were traditionally referred to in system research [41]: physical models and mathematical models, both of which are built for the study and analysis of systems. In the past decade, the mathematical or quantitative relation model has been extended to a larger framework where computing is the main theme because of the indispensable role of computer technology and corresponding development of methodology. Thus, the vast majority of models for system study purposes are computational, representing a system in terms of mathematical and logical relationships. Occasionally, analytical solutions can be obtained from traditional mathematical models. In real applications, the systems are dynamic, stochastic and highly complex, precluding any possibility of obtaining a closed-form analytical solution. In this case, computational modeling and simulation become the only

means to study such a system. There are many texts discussing the theory of modeling and simulation, e.g [70]. We will skip the details and focus on the modeling and simulation of traffic systems.

Traffic simulation became an indispensable tool for transport planning due to several driving forces: advances in modern traffic flow theory especially with respect to model complexities, and in computer hardware and software tools and in vehicle technology, development of general information-based infrastructure, and social demand for more detailed analysis of the consequences of traffic measures and plans. Traffic simulation models can be, according to the level of details, classified into: macroscopic, mesoscopic and microscopic. Macroscopic models represent road traffic as flow without considering its components and mutual interactions whereas microscopic models describe in very much detail the components of the traffic systems, e.g. vehicles, traffic signals, road based ITS systems, and their interactions. Mesoscopic models sit in between these two extremes and often represent vehicles based on traffic streams without considering their interactions. All these three levels of models have their own application focus: for example, microscopic simulation can be more adapted to the study of detailed driver behavior aspects while macroscopic models are more appropriate for the strategic evaluation of very large scale planning. Recently, much effort to mix those simulation models into one hybrid platform also deserves mentioning, e.g. in [8]. However, increasingly complicated microscopic simulation models will stand foremost in the long run due to the needs to design new ITS systems, especially vehicle based driving assistant systems, and to study safety and environmental effects of traffic systems. The fast evolution of computer and computational technologies is an impetus for this tendency.

1.3 Driver behavior

Driver behavior has been an academically fascinating area, alluring interdisciplinary research from different scientific or engineering fields including system modeling, psychology, automotive and transportation engineering and computer science. Nowadays, driver behavior models are mainly applied in the simulation of traffic systems and the design of intelligent cruise control and other in-vehicle driving assistant systems.

Driver tasks can be separated into a hierarchy of three levels: strategic, tactical and operational. On the strategic level, drivers make decisions concerning the route choice and trip planning not only before but also during their trip. This is a very important area in

transportation science where travel demand and trip generation are focused. Of course it is also indispensable for the traffic simulation since origin-destination (OD) matrix and en-route route choices affect the traffic distribution over a network. Decisions on the tactical level are driven by goals for speed, safety and comfort, and they determine the movement of vehicles on the roads. At last, operational level behavior includes the skill based activities such as steering, gearing, braking and so on. These activities are mostly done with little conscious effort.

1.3.1 Tactical driver modeling

Although tactical level behavior is directly affected by strategic decision and has to be implemented through operational control of vehicles, it is the main interactive interface between the driving vehicle and other vehicles and the traffic environment. So the models and algorithms on this level are always the main concerns within the topic of driver modeling. In microscopic traffic simulation, researchers are especially interested in mimicking drivers' behavior as close as possible to reality on a tactical level. Tactical models in simulation describe the driver behavior based on the current traffic situation and focus on short-term interactions between the vehicle and its environment, without drawing details on how drivers perceive and operate. Car following model is probably the most essential tactical driving models and describe the driver longitudinal behavior when following another car and trying to adapt its speed and maintain a safe distance. Figure 1.2 shows the typical situation when a driver drives his car n when it is behind a leading car $n - 1$ and advances without any intention of changing lane or overtaking. Gap acceptance models including lane changing, overtaking and merging behavior are also crucial for the authenticity of simulation models but our focus in this thesis is on particularly car-following behavior.

1.3.2 Review of car following models

As far as is known, driver behavior has been studied since the early 1950s. A considerable increase of researches has been taken place in the modeling of driver behavior, especially car following, during the last twenty years. In this section, the main car following models will be shortly reviewed by categories.

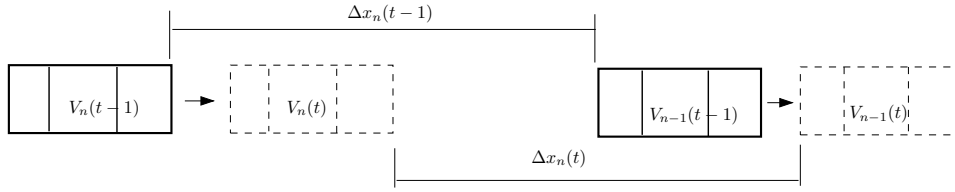


Figure 1.2: A standard situation for car-following.

General Motor (GM) model and its extensions

The well-known mathematical car following model introduced by Gazis et al [17] in 1961 was both an extension of the early models developed in the General Motor laboratory [10] [25] and a summary of the early ideas on stimulus-response type car following models. The model takes the form below

$$a_n(t+T) = \alpha \frac{v_n^l(t+T)}{\Delta x_n(t)^m} (v_{n-1}(t) - v_n(t)) \quad (1.1)$$

where $x(t)$, $v(t)$ and $a(t)$ are the position, speed and acceleration of the cars. T is the term called reaction time of the driver and l, m and α are constants. This model is often called the Nonlinear General Motor (GM) model and has the intuitive hypothesis that the follower's acceleration is proportional to the speed difference, $\Delta v_n(t) = v_{n-1}(t) - v_n(t)$, and its own speed but being inversely proportional to the distance headway, $\Delta x_n(t) = x_{n-1}(t) - x_n(t) - S_{n-1}$, where S_{n-1} stands for the length of the leading vehicle $n-1$. The speed difference part on the right hand side is always translated as 'stimulus term' while the rest is called 'sensitivity term'. Although we know that more factors are involved in a follower's decision in the real traffic environment, these variables show obvious effects and can be identified by modern equipment. Based on this mathematical equation, lots of extensions were made by later research. Subramanian [59] introduced a statistical term assumed to be Gaussian and described reaction time by a truncated log-normal distribution. Yang [68] developed the GM car following model in MITSIMLAB, a microscopic traffic simulator, but assuming the acceleration and deceleration are not symmetric, which is more reasonable in reality. Ahmed [2] extended the GM model with more complexity by allowing the sensitivity term to be correlated with local traffic density and using a piecewise nonlinear function to represent the stimulus term. The model is expressed as follows

$$a_n^{cf,g}(t) = s[X_n^{cf,g}(t - \xi\tau_n)]f[\Delta v_n(t - \tau_n)] + \epsilon_n^{cf,g}(t) \quad (1.2)$$

where,

$$\begin{aligned}
g &\in \{acc, dec\}, \\
s[X_n^{cf,g}(t - \xi\tau_n)] &= \text{sensitivity term}, \\
X_n^{cf,g}(t - \xi\tau_n) &= \text{vector of variables affecting acceleration at time } (t - \xi\tau_n), \\
f[\Delta v_n(t - \tau_n)] &= \text{stimulus term}, \\
\Delta v_n(t - \tau_n) &= (v_{n-1}(t - \xi\tau_n) - v_n(t - \xi\tau_n)), \\
\epsilon_n^{cf,g}(t) &= \text{stochastic random term at time } t, \\
\xi &\in [0, 1].
\end{aligned}$$

The parameter ξ in equation (1.2) describes the fact that drivers may update the perception of the traffic environment during the decision process. The acceleration and deceleration processes are modelled with different parameter sets. Besides the research on extending GM type models above, Addison and Low [1] suggested to add another term to the nonlinear GM model and so the model becomes

$$a_n(t + T) = \alpha \frac{v_n^l(t + T)}{\Delta x_n(t)^m} (v_{n-1}(t) - v_n(t)) + \beta(\Delta x_n(t) - D_n)^3 \quad (1.3)$$

where D represents the desired separation that the follower attempts to achieve. The main consideration for adding this term is that the previous GM type model only intends to match velocity of the consequent cars without attempting to realize a safe inter-vehicle separation. The numerical study of this model shows that it produces chaotic oscillations along the platoon which is often expected from more complicated model forms. So this model can be used to replicate the seemingly random inter-vehicle oscillation e.g. in congested traffic, though the model actually takes a deterministic form.

In addition to the efforts to extend the GM model, a great deal of work has also been performed to calibrate and validate the GM model. However, results from different researches were reported to be more or less different and some of them were even contradictory probably due to the large variations in the measurement means, traffic conditions and so on, all of which can not be completely captured by the GM models.

Linear models

Although the first car following model [10] developed in the General Motor laboratory is linear, this category of car following models was formally embarked by Helly [24] who

proposed that the acceleration of the follower should be not only correlated to the speed difference but adapted to whether the leading car brakes or not. A simplified form of Helly's linear model is represented by

$$\begin{aligned} a_n(t+T) &= C_1 \Delta v_n(t) + C_2 (\Delta x_n(t) - D_n(t+T)) \\ D_n(t+T) &= \alpha + \beta v_n(t) + \gamma a_n(t) \end{aligned} \quad (1.4)$$

where $\Delta v_n(t) = v_{n-1}(t) - v_n(t)$ and $\Delta x_n(t) = x_{n-1}(t) - x_n(t) - S_{n-1}$ are speed difference and space between the two cars at time t . $D_n(t)$ is the desired distance of the driver. T is the driver reaction time. It can be further showed that the acceleration of the following car in the model is linearly correlated with speed difference, previous acceleration, speed of the following car and distance headway between the two cars. Helly's model has been revised and studied by some researchers during the last forty years. Especially, it was implemented in a simulation model SITRA-B [16]. However, this model form is often considered devoid of general validity in capturing traffic characteristics in different situations and intuitively less justifiable than GM type models. Koshi et al [37] proposed in 1992 to extend Helly's model into a nonlinear form

$$a_n(t+T) = \alpha_1 \frac{\Delta v_n(t)}{\Delta x_n(t)^l} + \alpha_2 \frac{(\Delta x_n(t) - D_n(t+T))}{\Delta x_n(t)^m} \quad (1.5)$$

where α_1 , α_2 , l and m are all constants. Later, Xing [66] further extended the model to the general acceleration case using a more complex equation as follow

$$\begin{aligned} a_n(t) &= \alpha_1 \frac{\Delta v_n(t-T)}{\Delta x_n(t-T_1)^l} + \alpha_2 \frac{(\Delta x_n(t-T_2) - D_n(t))}{\Delta x_n(t-T_2)^m} + \alpha_3 \sin \theta + \alpha_4 (v_{des} - v_n(t)) \\ D_n(t) &= a_0 + a_1 v_n(t-T_2) + a_2 v_n(t-T_2)^2 + a_3 v_n(t-T_2)^3 \end{aligned} \quad (1.6)$$

where the last two terms handle the effect of a gradient and the free flow situation respectively. These nonlinear extensions of the Helly's model have not been further studied by numerical data except the calibration made in the original publications.

Gipps models

Considered as a further development of the original collision avoidance or safety based ideas e.g. [36], a mathematical model was proposed by Gipps [18] in a form of the following equations

$$v_n(t+T) = \min \{v_n^a(t), v_n^b(t)\}$$

$$\begin{aligned}
v_n^a(t) &= v_n(t) + 2.5A_nT(1 - v_n(t)/V)\sqrt{0.025 + v_n(t)/V} \\
v_n^b(t) &= B_nT + [B_n^2T^2 - B_n\{2\Delta x_n(t) - v_n(t)T - \frac{v_{n-1}(t)^2}{\hat{B}_{n-1}}\}]^{1/2}
\end{aligned} \tag{1.7}$$

where $v_n^a(t)$ is the follower's adopted speed when the headway between two contiguous cars is large enough so that it can accelerate freely. A_n is an acceleration parameter. The second part $v_n^b(t)$ is our main interest which describes the adopted speed when the headway is smaller. The formula is based on the assumption that each driver sets his speed so as to be able to stop, if adopting the maximal braking rate B_n , without hitting the vehicle in front given that the leading car brakes no more than a certain rate \hat{B}_{n-1} . $\Delta x_n(t) = x_{n-1}(t) - x_n(t) - S_{n-1}(t)$ is the space between the leading and following vehicles.

Although the model was not completely calibrated using measurements, Gipps shows through simulation that the model propagated disturbances within a platoon in a manner close to reality. The model was, later on, preferred in a number of simulation projects and an in-depth numerical analysis of the model was given in [64]. Accordingly, one advantage of this model is that users can calibrate it based on their commonsense of driver behavior, mainly the maximal braking rates and prediction of the braking rate of the leading car. However, the speed model was derived based on the integration of acceleration with a crude time step size equal to the reaction time. Meanwhile, some empirical studies e.g. [55] showed that it performed not as well as other mathematical models.

Psycho-physical models

From a behavior standpoint, Leutzbach [42] introduced the psychological concepts of 'perception threshold' in the car following modeling since drivers do not follow cars at large distance and can not clearly perceive small difference in relative speeds or speed differences. Inspired by these ideas, Leutzbach and Wiedermann [43] presented a model based on drivers perception thresholds of the relative speed or speed difference, the perception limits of opening and closing for small relative speeds and that for perceiving changes in headway distance. In short, four thresholds can be summarized as follows:

- Minimum desired following distance, D_{mind} ,

$$D_{mind} = D_{stop} + D_x\sqrt{v_n} \tag{1.8}$$

where D_{stop} stands for minimum desired spacing between two vehicles when stationary, while D_x is the additional spacing to account for moving speed v_n ;

- Maximum desired following distance, D_{maxd} ,

$$D_{maxd} = D_{stop} + \lambda D_x \sqrt{v_n} \quad (1.9)$$

where λ makes D_{maxd} reasonably larger than D_{mind} ;

- Small negative relative speed

$$\delta v^- = -\Delta x^2 / \theta \quad (1.10)$$

where θ is a constant corresponding to the divergence rate of the visual angle;

- Small positive relative speed

$$\delta v^+ = \Delta x^2 / \epsilon \quad (1.11)$$

where ϵ is a constant corresponding to the convergence rate of the visual angle.

Accordingly, drivers perceive unacceptable changes in distance headway and speed difference and will execute a level transition of their actions. Meanwhile, the desired speed and want for safety were also defined in the model, which make the model reflect not only the human's common psychological limits but the individual difference within drivers. Empirically, Brackstone et al. [7] attempted to calibrate those psychological thresholds using data collected from UK motorways, but those parameters have not been validated by real observation.

Rule-based multi-stage systems

During the early era of developing car following models, researchers had already considered to divide the car following into different stages and situations. For example, Wiedermann described the car-following process by four situations including uninfluenced driving or free flow situation, approaching, braking and following the leader in order to reflect different levels of perception; Yang introduced both a free flow and an emergency braking stage in addition to the normal car-following. Hence a car-following process can be divided into different stages base on simple rules, and model parameters in different stages have to be estimated based on the data from the corresponding ones. All these models face an insurmountable challenge on designing calibration methods in the early period. Kosonen [38] modeled the general driver speed control behavior including the car following by a rule set in a microscopic simulation tool, HUTSIM. Later, this model was adopted in Swedish

TPMA [39]. In the model, the headway measured in time unit is divided into three zones: the free area, the stable region and the forbidden area. Driver actions are decided by the rules such as

$$IF D_{obs} < S_{min}(v_{own}, v_{ahead}) + W_{stab}(v_{own}, v_{ahead}) THEN no acceleration.$$

where S_{min} and W_{stab} stand for the minimum safe distance and width of stable area respectively, which are functions of speeds of both leading vehicle and its own. While the rule-based approach has the advantage regarding driver knowledge representation by a human-like inference, the model does not capture the essential uncertainties during a driving task and the calibration of the model is hampered by the concepts involved in the parameters. In addition, the speed output of the model is discrete and crude, making the simulation output not realistic.

Fuzzy inference models

Fuzzy logic based system has been introduced in modeling driver behavior due to its successful applications in many industry areas. Especially, fuzzy set theory [69] was originally expected by its inventor, Lofti Zadeh, to solve problems in human science. The initial use of this method by Kikuchi et al [34] in driver modeling attempted to design a Sugeno Fuzzy Inference System (Sugeno FIS) by fuzzification of the speed difference δv , headway δx and acceleration a . The FIS system is composed of rules like:

$$\text{Rule i: } IF \delta x = 'ADEQUATE' THEN a_n = (\delta v + a_{n-1}T)/\gamma;$$

In the fuzzy rule above, the variables in the antecedence are described by fuzzy sets or linguistic terms while the consequence is represented by an equation. This system does not need the defuzzification process and is therefore computationally fast. Later on, both Wu et al [65] and Chakroborty et al [9] developed a Mandani Fuzzy Inference System (Mandani FIS) by representing the acceleration action with some fuzzy sets like: { '*Strong Acceleration*', '*Light Acceleration*', '*No action*', '*Light Deceleration*', '*Strong Deceleration*' }. Armed with fuzzy sets in the consequence, the fuzzy rule becomes as follow:

$$\text{Rule j: } IF \delta x = 'ADEQUATE' \text{ and } \delta v = 'LARGE POSITIVE' \text{ and } a_{n-1} = 'LIGHT POSITIVE' THEN a_n = 'LIGHT POSITIVE'.$$

Hence, a defuzzification process is required to get a numerical value on the acceleration output. Despite the semantic advantages of Mandani FIS in delineating the human behavior,

the defuzzification is computationally expensive especially for large scale traffic simulation systems, in which computing speed is always critical.

As far as known, a fuzzy inference system can be calibrated by different techniques such as adaptive neural networks and evolution algorithms. But the calibration methods for the fuzzy systems above have formally applied in the context of modeling driver behaviors, and the applications of fuzzy driver models in large scale traffic simulations have not been reported.

Control-based approaches

Driver behavior modeling has also attracted researchers in automotive and control due to their common interest in designing and installing vehicle based ITS systems such as Adaptive Cruise Control (ACC) systems and analyzing their effects on safety and environment. Therefore, control based model approaches have been implemented as in [6]. Moreover, the state space modeling and subspace identification methods [4] are also applied in the ACC system design as a complementarity to the behavior based model approaches.

1.4 Identification of research needs

It is evident from the previous review that much effort has been spent on research of driver modeling, especially car-following behavior. However, early research focused on studying the physical and numerical properties of different types of car-following models and developing up-to-date traffic flow theory in which car-following model was a sub-task. With increasing interest on the accurate modeling of driver behavior, later researchers started to shift focus toward modeling the perception and reaction of the human driver using measurement data collected from real traffic. One main reason for this trend is the common use of microscopic traffic simulation as a tool for transportation modeling where an authentic replication of driver behavior becomes essentially important for successful applications. Another reason for this tendency in driver behavior research is its wide applications in Intelligent Transportation Systems (ITS) such as vehicle based driving assistant systems, which require detailed understanding of various behavior responses of drivers and their adaption to new systems.

One traditional way to interpret the cognition and decision process of drivers was by a pure mathematical formula based on a concept framework. The famous GM models are time

delay differential equations belonging to the stimulus-response type model. Mathematical formula can describe driver actions exactly and model parameters can be estimated from empirical data based on certain fitness function, e.g. summation of deviation between model and real acceleration. In addition, stochastic terms could also be introduced in order to represent the variances involved in driver behavior. However, a single mathematical equation does not have enough elasticities in representing human knowledge or intelligence. For example, a driver's decision is based on imprecise perceptions of the stimuli from the traffic environment. On the other hand, when the model complexity is enough to reflect the dynamics, the resulting mathematical equations are often difficult to solve.

Since the early research in the development of car-following models, researchers have collected data using different means in order to calibrate their models. Although the quality of the data was limited in the early stage, the criteria for a good model was whether it could fit the empirical data well enough. During the last decade, different measurement devices have become available so that more properties of the interactive objects in transportation systems can be observed with significantly improved accuracy. Hence, new questions come to the research fronts, e.g.

- How can we improve the accuracy and consistency of collected behavior data?
- Are there more flexible ways to describe human ability in the driving task?
- How can we estimate (or train) the behavioral models using real data?

These questions reflect a basic fact: mimicking human level intelligence in both ITS applications and simulation requires better understanding and description of real driver behavior.

Kim et al. [35] emphasized the necessity to construct models from data in order to capture driving patterns completely from observations of traffic. The idea generalizes the new trends in driver modeling method and its evolution from the traditional approaches. This inspires us to explore a way to create our models from real data, i.e. model structure and parameters can be identified by exploration of data measured from real world.

1.5 Objective

The overall objective of this thesis is to develop a methodological framework represented in figure 1.3 for modeling driver behavior, especially car-following, in the following steps:

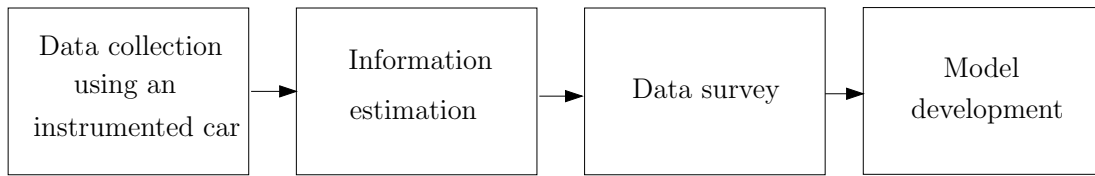


Figure 1.3: A methodological framework for modeling driver behavior.

data collection using an advanced instrumented vehicle, information processing and estimation, data survey and model development. To emphasize the computational aspects of this data-mining procedure, a number of techniques have been extended and applied in different research subtasks, which can be summarized under the umbrella of modern Artificial Intelligence (AI) or Computational Intelligence (CI) methods. For example, fuzzy inference will be used as a modeling approach, and Kalman filter and Genetic algorithm will be applied as the model estimation algorithm (or machine learning method in AI). This thesis also intends to contribute toward a promotion of application of modern computational technology in the fields of driver behavior modeling and ITS development.

1.6 Research contributions

Research contributions of this work are in both transportation and computer science. The main contribution to transportation is the development of a general approach for driver behavioral modeling, as accurate driver modeling is fundamental for simulation in transport analysis. Advanced computing methods have been applied in the methodology development to solve different problems including

- Application of Kalman-Filter based information processing method for accurate estimation of car-following data from real measurement;
- Investigation of the driver properties and estimation of the driver reaction time based on spectrum analysis methods;
- Investigation of the statistical relations between perceptual variables and acceleration response in each regime using regime decomposition and regression analysis.

In this data-mining process, some computing methods have been developed or further extended in order to address corresponding problems given the application context i.e.

- Extension of the classical fuzzy clustering algorithm to a classification task of multivariate time series data;
- Adaption of Kalman-filter for a specific optimization objective and iterative usage of extended Kalman-Filter in a dynamic model estimation context;
- Development of an innovative neural-fuzzy system for regime-based computational modeling of car-following behavior.

Along with driver modeling study, an evaluation of the systematic safety effect of Intelligent Speed Adaption (ISA) was conducted using simulation.

1.7 Thesis outline

The remainder of this thesis consists of two chapters. In the second chapter, some basic computational and modeling techniques in the CI area are generally reviewed; chapter three presents a research summary based on six papers attached; moreover, the research findings in this thesis are concluded with limitation and future study perspectives. At last, six research papers are attached in the appendix.

Chapter 2

Methodological review

This thesis addresses driver modeling using measurement data from real traffic environment. Many approaches adopted in this research belong to the discipline of modern Artificial Intelligence (AI). Hence, a methodological review on modern AI is conducted here before setting foot in the main research topic. Modern AI methods are also called Computational Intelligence (CI) methods in which Soft Computing (SC) is in the research front. Instead of relying mainly on the symbolic manipulation, logic programming and fast searching algorithms in traditional AI, CI methods resort to the numerical computational ability of digital computers, adopt approaches inspired by human intelligence, biological evolutions and natural phenomena, and include many computational algorithms from adaptive system theory, signal processing and numerical analysis. For example, the Artificial Neural Network models are prompted by the structure and functionalities of human brain and neural system, the social behavior of bird flocks are mimicked to solve optimization problems and adaptive filtering theory is applied in machine learning. Although CI methods can be used to solve many problems as independent entities, they are often more powerful when hybridizing with each other as an integrated framework. An obvious example is the combination of fuzzy logic with neural networks and evolutionary algorithms, which formulates the core of the Soft computing methods. In general, CI is a cutting-edge research area in computer science and information technology, with high potentials to gain a broad market in all scientific and engineering fields.

Computational intelligence has shown its strength in a large number of applications in many fields such as automatic control, system identification and modeling, signal processing, time series analysis, decision making, operational research etc. In transportation science, CI

methods have been successfully applied in different subjects: simulation models (e.g driver behavior), traffic forecasting, incident detections, operational optimizations etc. Although most of the CI methods are newly developed, they are essentially rooted in traditional mathematical concepts, e.g. least square method, and often derived using classical analytical tools, e.g. regression analysis and numerical optimization. Therefore, in this chapter we will give a review on some basic mathematical and algorithmic elements being frequently adopted in modern AI methodology. However, the modern AI area has been developed as such a broad field that we can only light a small candle in its universe, and it is difficult to go deep without analytical rigors and application contexts. Therefore, this chapter mainly provides some fundamental concepts and approaches as a background for later research content.

2.1 Least square and numerical optimization

Regression and optimization technologies provide solid mathematical foundations for the basic system modeling methodology in the model structure identification and parameter estimation. In this section, we will start from the basic linear least squares identification and then go further to the derivative-based nonlinear optimization schemes. In addition, heuristic optimization methods based on evolutionary principles such as naive genetic algorithm and swarm based algorithm are also shortly introduced. The review of these topics will only stay in a relatively ordinary form and advanced topics, e.g. generalized and partial least square methods, will not be covered due to the scope limitation. A more rigorous and throughout treatment of most topics in this section can be found in advanced literature e.g. [12].

2.1.1 Least square methods

The discovery of the least squares (LS) method dates back to 1795 when Gauss used it to refine his estimates of the orbit of Ceres. Since then, the LS method has attracted continuing interests and been beneficial for generations of scientists and engineers. In engineering application, it is often necessary to identify the model based on observation of system inputs and outputs in figure 2.1. The most fundamental model often takes a linear form as follows

$$y = \theta_1 x_1 + \cdots + \theta_n x_n$$

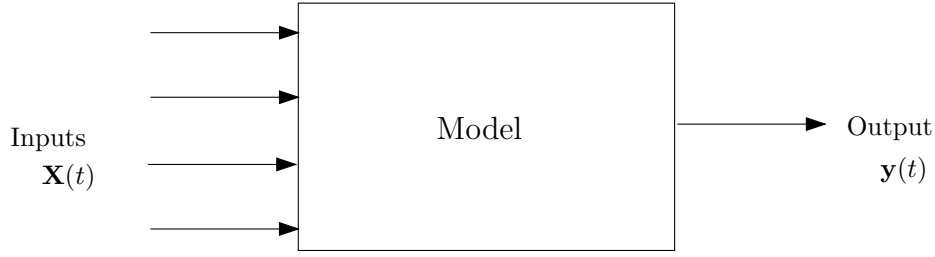


Figure 2.1: A system model with observable inputs and outputs.

or using matrix notations

$$y = \Theta^T \mathbf{x} \quad (2.1)$$

where $\mathbf{x} = [x_1 \cdots x_n]^T$ is the input vector and $\Theta = [\theta_1 \cdots \theta_n]^T$ is the vector of model parameters. When the inputs are simply deterministic, an energy function of LS optimization criterion can be built for the estimation of the parameter vector Θ using observed inputs and corresponding outputs i.e.

$$\mathcal{E}(\Theta) = \sum_{t=0}^{m-1} (y(t) - \Theta^T \mathbf{x}(t))^2 \quad (2.2)$$

or in the matrix form

$$\mathcal{E}(\Theta) = (\mathbf{y} - \mathbf{A}\Theta)^T (\mathbf{y} - \mathbf{A}\Theta) \quad (2.3)$$

where $\mathbf{y} = [y(0) \cdots y(M-1)]^T$ is the measurement of the output and

$$\mathbf{A} = \begin{bmatrix} x_1(0) & \cdots & x_n(0) \\ \vdots & \ddots & \vdots \\ x_1(m-1) & \cdots & x_n(m-1) \end{bmatrix} \quad (2.4)$$

is composed of the measurement of the inputs where $m > n$ and $\text{rank}(\mathbf{A}) = n$.¹

Normally, the problem can be solved by applying the first order condition i.e.

$$\frac{\partial \mathcal{E}}{\partial \Theta} = \mathbf{0} \quad (2.5)$$

¹Here \mathbf{A} is simply assumed non-singular; in the singular case, its pseudo or generalized inverse can be obtained by methods e.g. singular value decomposition (SVD).

which leads to the normal equation

$$(\mathbf{y} - \mathbf{A} \cdot \boldsymbol{\Theta})^T \mathbf{A} = \mathbf{0}. \quad (2.6)$$

Geometrically, the product $\mathbf{A} \cdot \boldsymbol{\Theta}$ can be treated as a hyperplane, which is a linear combination or subspace span of the column vectors $[x_i(0) \cdots x_i(M-1)]^T$ of \mathbf{A} . Hence the LS approach attempts to minimize the distance between the data vector \mathbf{y} and the hyperplane. The error vector $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{A} \cdot \boldsymbol{\Theta}$ is often called the innovation vector and the equation (2.6) can be obtained from the basic orthogonal principle in vector analysis. An alternative method is to reformulate the objective equation (2.3) as a complete quadratic as follows

$$\begin{aligned} \mathcal{E}(\boldsymbol{\Theta}) &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{H} \boldsymbol{\Theta} + \boldsymbol{\Theta}^T \mathbf{H}^T \mathbf{H} \boldsymbol{\Theta} \\ &= (\mathbf{y} - \mathbf{A} \boldsymbol{\Theta}_{\text{opt}})^T (\mathbf{y} - \mathbf{A} \boldsymbol{\Theta}_{\text{opt}}) + (\boldsymbol{\Theta}_{\text{opt}} - \boldsymbol{\Theta})^T \mathbf{H}^T \mathbf{H} (\boldsymbol{\Theta}_{\text{opt}} - \boldsymbol{\Theta}) \end{aligned} \quad (2.7)$$

so that the optimal solution

$$\boldsymbol{\Theta}_{\text{opt}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}, \quad (2.8)$$

can be directly obtained, which has the same form as the solution of the normal equation (2.6). Meanwhile, the optimum of the objective function is

$$\begin{aligned} \mathcal{E}_{\min} &= (\mathbf{y} - \mathbf{A} \boldsymbol{\Theta}_{\text{opt}})^T (\mathbf{y} - \mathbf{A} \boldsymbol{\Theta}_{\text{opt}}) \\ &= \mathbf{y}^T (\mathbf{I} - \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T) \mathbf{y}. \end{aligned} \quad (2.9)$$

When the system inputs are stochastic, the linear model becomes

$$Y = \hat{Y}(\mathbf{X}) = \theta_1 X_1 + \cdots + \theta_n X_n$$

where \mathbf{X} is random vector so that the old LS objective is replaced by the mean square error (MSE)

$$\mathcal{E}_{MSE}(\boldsymbol{\Theta}) = \mathbb{E}[(Y - \hat{Y}(\mathbf{X}))^2] \quad (2.10)$$

where $\mathbb{E}[\cdot]$ is operator of expected value. Using the first order condition to minimize the new objective above, we obtain

$$\frac{d}{d\theta_k} \mathcal{E}_{MSE}(\boldsymbol{\Theta}) = -2\mathbb{E}[(Y - \mathbf{X}^T \boldsymbol{\Theta}) X_k] = 0 \quad k = 1, \dots, n \quad (2.11)$$

which can be further written as

$$\mathbb{E}[\mathbf{X}(Y - \mathbf{X}^T \boldsymbol{\Theta})] = 0. \quad (2.12)$$

This equation fulfills the orthogonal principle when the vector is random. It results in a unique optimal solution for the estimation parameters

$$\Theta_{opt} = \mathbf{R}_{XX}^{-1} \mathbf{r}_{XY} \quad (2.13)$$

where $\mathbf{R}_{XX} = \mathbb{E}[\mathbf{X}\mathbf{X}^T]$ and $\mathbf{r}_{XY} = \mathbb{E}[\mathbf{X}Y]$. Meanwhile, the linear MMSE estimate of y given \mathbf{X} is

$$\hat{y}(\mathbf{X}) = \mathbf{r}_{YX} \mathbf{R}_{XX}^{-1} \mathbf{X} \quad (2.14)$$

and the minimum value of the mean square error can be derived as

$$\mathcal{E}_{MSE}(\Theta_{opt}) = \mathbb{E}[Y^2] - \mathbf{r}_{YX} \mathbf{R}_{XX}^{-1} \mathbf{r}_{XY}. \quad (2.15)$$

In the illustration above, we have simply assumed that the model output is scalar. In fact, it is quite convenient to extend the result to the case of a vector output.

2.1.2 Derivative-based optimization

After considering the linear model identification by the LS criteria, it is natural to extend the context to the nonlinear regression and model identification. Unlike the linear case, nonlinear identification problems can not be commonly solved with an analytical solution, and resorting to numerical methods is often quite necessary. Hence,

Derivative-based optimization technologies serve as a fundamental class of methods for general numerical optimization problems in multidisciplinary research including system identification, numerical analysis, signal processing and among others. In particular, they work as a type of essential algorithms for model learning or training an intelligent machine, e.g. artificial neural networks. In this class of methods, the optimal point is determined by iteratively search in the directions related to the derivative of the objective functions. Steepest decent and Newton's methods are basis of this group of numerical optimization technologies. Steepest decent and conjugate gradient methods are also major algorithms used in artificial neural network learning in conjunction with the backward-error-propagation algorithm. Moreover, methods such as Levenberg-Marquardt become a standard tool used in data fitting and regression involving nonlinear models such as the Neural Networks and Neural-Fuzzy models. Therefore, derivative-based optimization approach is a very important component in the general computational intelligence methodology.

Descent method

Consider the minimization problem

$$\min_{\Theta} \mathcal{E}(\Theta)$$

where $\Theta = [\theta_1 \cdots \theta_n]^T$ is a n -dimensional input vector and $\mathcal{E}(\Theta)$ is the objective function. Since the objective function is sometimes quite complex and nonlinear, we often resort to an iterative algorithm to explore the input space

$$\Theta_{k+1} = \Theta_k + \lambda_k \cdot \mathbf{d}_k \quad (2.16)$$

where λ_k is step size, normally a small positive value and \mathbf{d}_k is the searching direction vector. The iterative method tries to determine the descent direction in each step and then find a step size to go to ensure that

$$\mathcal{E}(\Theta_{k+1}) = \mathcal{E}(\Theta_k + \lambda \cdot \mathbf{d}) < \mathcal{E}(\Theta_k). \quad (2.17)$$

If $\mathcal{E}(\Theta)$ is differentiable and the gradient of the function is defined as

$$\mathbf{g} = \nabla \mathcal{E} \doteq \left[\frac{\partial \mathcal{E}}{\partial \theta_1} \cdots \frac{\partial \mathcal{E}}{\partial \theta_n} \right]$$

the condition for the feasible descent directions satisfies

$$\left. \frac{d\mathcal{E}(\Theta_k + \lambda \cdot \mathbf{d})}{d\lambda} \right|_{\lambda=0} = \mathbf{g}^T \mathbf{d} < 0.$$

In general, the gradient-based descent methods has a form deflecting the gradients with a matrix \mathbf{H}_k , e.g. $\mathbf{d} = -\mathbf{H}_k \mathbf{g}_k$

$$\Theta_{k+1} = \Theta_k - \lambda \mathbf{H}_k \mathbf{g}_k. \quad (2.18)$$

When $\mathbf{H}_k = \mathbf{I}$, it becomes the steepest descent method.

Newton's method

If the initial point is sufficiently close to the optimal point, we can approximate $\mathcal{E}(\Theta)$ as the Taylor expansion

$$\mathcal{E}(\Theta) \approx \mathcal{E}(\Theta_k) + \mathbf{g}^T (\Theta - \Theta_k) + \frac{1}{2} (\Theta - \Theta_k)^T \mathbf{H} (\Theta - \Theta_k) \quad (2.19)$$

where \mathbf{H} is the Hessian matrix, consisting of the second partial derivatives of $\mathcal{E}(\Theta)$. By taking the derivative of this approximation and setting it to zero, we get the equation of the Newton's method

$$\Theta_{k+1} = \Theta_k - \mathbf{H}^{-1} \mathbf{g} \quad (2.20)$$

In practice, a small or adaptive step size is often applied to classical Newton method to improve its convergence property, that is

$$\Theta_{k+1} = \Theta_k - \lambda \mathbf{H}^{-1} \mathbf{g}. \quad (2.21)$$

However, the Newton method will converge to local minimum only if \mathbf{H} is positive definite. When it is not positive definite, the Newton iterations will converge to a local maximum or saddle point. The Levenberg-Marquardt approach can improve the method by adding some positive diagonal to \mathbf{H} to make it positive definite and the iteration becomes

$$\Theta_{k+1} = \Theta_k - \lambda(\mathbf{H} + \eta \mathbf{I})^{-1} \mathbf{g}. \quad (2.22)$$

Nonlinear least-square problem

As we have mentioned in advance, least square problem is most interesting from the model identification point of view. In the nonlinear LS, the general objective function can be further represented as

$$\mathcal{E}(\Theta) = \mathbf{e}(\Theta)^T \mathbf{e}(\Theta) = \sum_{p=1}^m e_p(\Theta)^2 = \sum_{p=1}^m (t_p - f(\mathbf{x}_p, \Theta))^2 \quad (2.23)$$

where $\mathbf{e}(\Theta)$ is the error vector, $f(\mathbf{x}_p, \Theta)$ is the model which takes inputs \mathbf{x}_p and is characterized by a parameter set Θ ; t_p is the real dataset that needs to fit. The gradient vector of $\mathcal{E}(\Theta)$ is

$$\mathbf{g}(\Theta) = \frac{\partial \mathcal{E}(\Theta)}{\partial \Theta} = 2 \sum_{p=1}^m e_p(\Theta) \frac{\partial e_p(\Theta)}{\partial \Theta} = 2 \mathbf{J}^T \mathbf{e}. \quad (2.24)$$

where $\mathbf{J} = [\frac{\partial \mathbf{e}}{\partial \theta_1}, \dots, \frac{\partial \mathbf{e}}{\partial \theta_n}]^T$, the Jacobian matrix of $\mathbf{e}(\Theta)$. The Hessian matrix is represented by

$$\begin{aligned} \mathbf{H}(\Theta) &= \frac{\partial^2 \mathcal{E}(\Theta)}{\partial \Theta \partial \Theta^T} \\ &= 2 \sum_{p=1}^m \left[\frac{\partial e_p(\Theta)}{\partial \Theta} \frac{\partial e_p(\Theta)}{\partial \Theta^T} + e_p(\Theta) \frac{\partial^2 e_p(\Theta)}{\partial \Theta \partial \Theta^T} \right] \\ &= 2(\mathbf{J}^T \mathbf{J} + \mathbf{S}) \end{aligned} \quad (2.25)$$

where \mathbf{S} is defined as the summation of the second term. Applying the Newton's approach and omitting the second order term in this least square problem bring us Gauss-Newton

formula as follow

$$\Theta_{k+1} = \Theta_k - (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \mathbf{e} = \Theta_k - \frac{1}{2} (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{g}. \quad (2.26)$$

It is also worth mentioning that a modification of Gauss-Newton formula adds step size in each iteration, which is called damped Newton methods.

Levenberg-Marquardt approach has also been applied to the Gauss-Newton scheme to handle ill-conditioned matrices $\mathbf{J}^T \mathbf{J}$. Then the iteration becomes

$$\Theta_{k+1} = \Theta_k - \frac{1}{2} (\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})^{-1} \mathbf{g}, \quad (2.27)$$

and there are also some variations on this formula. In general, Levenberg-Marquardt approach works well in practice and becomes the standard solution routines for nonlinear least-square problems.

Recursive least-square method

In this sections, we consider an iterative method to identify linear model equation (2.1) using a more general LS criteria as follows

$$\mathcal{E}_{RLS}(\Theta) = \sum_{t=0}^{m-1} \mu^{m-1-t} (y(t) - \Theta^T \mathbf{x}(t))^2 \quad (2.28)$$

where $0 < \mu \leq 1$ is defined as a forgetting factor so that the new data is given a higher weight than the historical data. According to the Newton approach in equation (2.20), an iterative algorithm as follows can be applied

$$\Theta_k = \Theta_{k-1} - \left[\frac{\partial \mathcal{E}_{RLS}}{\partial \Theta \partial \Theta^T} \right]^{-1} \frac{\partial \mathcal{E}_{RLS}}{\partial \Theta}. \quad (2.29)$$

Based on the following definitions:

$$\mathbf{C}_k^{\mathbf{xx}} = \sum_{t=0}^k \mu^{m-1-t} \mathbf{x}(t) \mathbf{x}(t)^T \quad (2.30)$$

and

$$\mathbf{C}_k^{\mathbf{xy}} = \sum_{t=0}^k \mu^{m-1-t} \mathbf{x}(t) y(t), \quad (2.31)$$

and the recursive relations:

$$\mathbf{C}_k^{\mathbf{xx}} = \mu \mathbf{C}_{k-1}^{\mathbf{xx}} + \mathbf{x}(k) \mathbf{x}(k)^T \quad (2.32)$$

and

$$\mathbf{C}_k^{\mathbf{x}y} = \mu \mathbf{C}_{k-1}^{\mathbf{x}y} + \mathbf{x}(k)y(k), \quad (2.33)$$

we get the basic form of recursive least square (RLS) algorithm

$$\begin{aligned} \Theta_k &= \Theta_{k-1} - [\mathbf{C}_k^{\mathbf{x}\mathbf{x}}]^{-1}(\mathbf{C}_k^{\mathbf{x}y} - \mathbf{C}_k^{\mathbf{x}\mathbf{x}}\Theta_{k-1}) \\ &= \Theta_{k-1} - [\mathbf{C}_k^{\mathbf{x}\mathbf{x}}]^{-1}\mathbf{x}(k)(y(k) - \mathbf{x}^T(k)\Theta_{k-1}). \end{aligned} \quad (2.34)$$

Defining $\mathbf{Q}(k) = [\mathbf{C}_k^{\mathbf{x}\mathbf{x}}]^{-1}$ and using the matrix inverse lemma², we can finally get a compact and efficient format of RLS algorithm as follows

$$\Theta_k = \Theta_{k-1} - \mathbf{Q}(k)\mathbf{x}(k)(y(k) - \mathbf{x}^T(k)\Theta_{k-1}) \quad (2.35)$$

$$\mathbf{Q}(k) = \frac{1}{\mu}(\mathbf{Q}(k-1) - \frac{\mathbf{Q}(k-1)\mathbf{x}(k-1)\mathbf{x}(k-1)^T\mathbf{Q}(k-1)}{\mu + \mathbf{x}(k-1)^T\mathbf{Q}(k-1)\mathbf{x}(k-1)}). \quad (2.36)$$

Note $\mu = 1$ (no forgetting or equally weighted form) is often used in practice. As an approximate second-order iterative method for nonlinear models, RLS algorithm has faster convergence rate than common LS methods. Hence, it is popular in many model estimation applications within the computational intelligence framework.

Conjugate gradient method

Conjugate gradient method belongs to Krylov subspace methods [12], which is a group of methods based on the conjugacy, a concept extended from orthogonality. In general, for a symmetric matrix \mathbf{H} , two n -dimensional vectors \mathbf{d}_j and \mathbf{d}_k are conjugate with respect to \mathbf{H} if the following equation holds

$$\mathbf{d}_j^T \mathbf{H} \mathbf{d}_k = 0. \quad (2.37)$$

When \mathbf{H} is positive definite, the n mutually conjugate (nonzero) vectors \mathbf{d}_k are linearly independent. It is well-known that Gram-Schmidt orthogonalization can be performed to convert an arbitrary basis $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_k\}$ into an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$, that is

$$\mathbf{u}_k = \mathbf{s}_k - \sum_{j=0}^{k-1} \frac{\mathbf{u}_j^T \mathbf{s}_k}{\mathbf{u}_j^T \mathbf{u}_j} \mathbf{u}_j. \quad (2.38)$$

²Let A be square matrix and B, C and D be matrices such that BCD is well defined and has the same dimension as A , then $(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$.

Since conjugacy is a concept extended from orthogonality, the basis $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_k\}$ can be converted into descent vectors $\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k\}$, which are mutually conjugate with respect to \mathbf{H}

$$\mathbf{d}_k = \mathbf{s}_k - \sum_{j=0}^{k-1} \frac{\mathbf{d}_j^T \mathbf{H} \mathbf{s}_k}{\mathbf{d}_j^T \mathbf{H} \mathbf{d}_j} \mathbf{d}_j. \quad (2.39)$$

When the gradient vector is used to determine conjugate directions, we get

$$\mathbf{d}_k = -\mathbf{g}_k + \sum_{j=0}^{k-1} \frac{\mathbf{d}_j^T \mathbf{H} \mathbf{g}_k}{\mathbf{d}_j^T \mathbf{H} \mathbf{d}_j} \mathbf{d}_j \quad (2.40)$$

Differentiating equation (2.19) yields

$$\mathbf{H}(\Theta_{k+1} - \Theta_k) = \mathbf{g}_{k+1} - \mathbf{g}_k$$

where $\Theta_{k+1} - \Theta_k = \lambda \mathbf{d}_k$, so

$$\mathbf{d}_k = -\mathbf{g}_k + \sum_{j=0}^{k-1} \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{g}_k}{\mathbf{d}_j^T (\mathbf{g}_{k+1} - \mathbf{g}_k)} \mathbf{d}_j. \quad (2.41)$$

It can be proved that if \mathbf{d}_j ($j = 0, 1, \dots, k-1$) are mutually conjugate with respect to \mathbf{H} then the gradients satisfies

$$\mathbf{g}_k \mathbf{d}_j = \mathbf{0} \text{ for } j < k.$$

Hence, we get the most common conjugate gradient iteration

$$\mathbf{d}_k = -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}. \quad (2.42)$$

where

$$\beta_k = \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T \mathbf{g}_k}{\mathbf{d}_{k-1}^T (\mathbf{g}_k - \mathbf{g}_{k-1})}.$$

In principle, conjugate gradient method converges, if not considering rounding errors and other factors, within the steps of the space dimension n when treating with linear least square problem. For the nonlinear LS case, it still keep an approximately second order convergence rate [12]. Thus, it is a popular approach for LS identification.

Convergence

Although the derivative-based algorithms introduced above have been widely applied in practice, there is no guarantee for any of them to find the global optimum of a complex

objective function within finite computational effort. Since all of these algorithms are deterministic in the sense that they inevitably lead to the local minimum, selecting initial positions have a decisive effect on the global convergence. However, it is impossible in practice to know a good starting point, and thus a stochastic mechanism should be introduced in the initialization to escape from trapping into a local minimum point. But the involvement of a stochastic scheme also means more computational time in solving optimization problems.

2.1.3 Evolutionary-based optimization

Evolutionary-based methods have been extensively used in solving optimization problem due to its ability in searching global optimum. The most widely used methods include *genetic algorithms* (GA), *simulated annealing* (SA), *particle-swarm-based algorithms* (PSA) and so on; These methods do not need derivative information. Instead, they rely on the repeatedly evaluation of the objective function based on some heuristic guidelines motivated by the wisdom of nature, such as biology evolution, physical phenomena and social behavior of animals and insects. In this section, only GA and PSA algorithms are briefly introduced and figure 2.2 illustrates the principle and computational procedures of both methods.

GA might be the most widely used evolutionary optimization technique. It is inspired by the evolution of biological systems. A solution to the problem is corresponding to the characteristics of such a system coded in the form of gene sequence, chromosome. The algorithm starts with a random population of solution chromosomes. The fitness function is evaluated for each solution chromosome and the Darwinian law of natural evolution is simulated iteratively: best chromosomes are survived and a random part of them exchange or mutate the genes through the crossover and mutation procedure to produce more advanced offspring. A more detailed description can be found in the literature e.g. [19].

During the last decade, a number of other type evolution optimization schemes [14] are developed based on the simulation of social behavior of various species e.g. ants, birds, frogs and so on. The performance of these methods are different in solving various optimization problems, and the PSA algorithm was reported to be better performed than others [14]. Thus, this algorithm is briefly reviewed here. The PSA is inspired by the social behavior of a bird flock. A bird is corresponding to a particle. Instead of producing new generations, the birds evolve their behavior toward a destination. Each bird looks a certain direction and through communication they determine the best location. The birds have their own

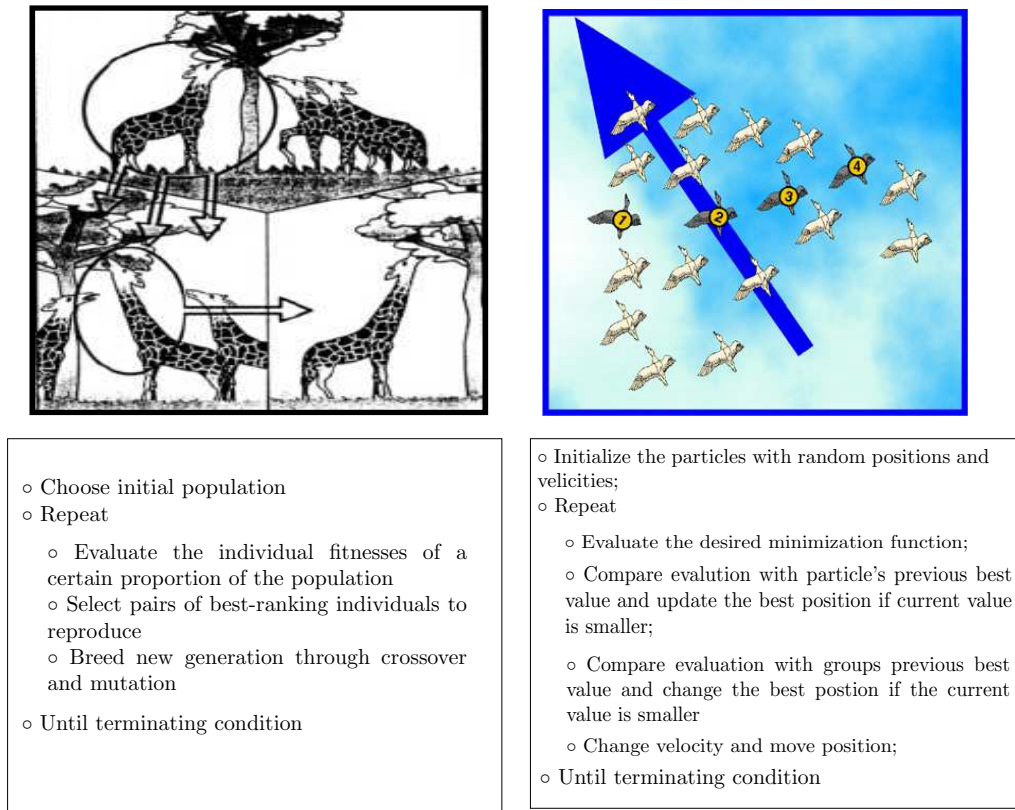


Figure 2.2: Illustrations of the genetic algorithm (left) and particle swarm based optimization algorithm (right).

speed according to their current positions. This process involves two essential parts: the intelligence of each bird makes them learn from the experience (local search) and the communication between birds allows them learn from others (global search).

2.2 Kalman filtering method

Kalman filter [32] is one of the greatest discoveries in the history of statistical estimation theory. It has found applications in almost every engineering field where real-time evaluation of complex dynamic systems is needed. In theory, Kalman filter is actually an estimator for the linear-quadratic problems in which the instantaneous states of a linear dynamic system are disturbed by white noise. However, as a filter it also plays an important role in the

adaptive filter theory [22], which has been applied in the machine learning problems. In the real applications, the measurements are often a linear function of the system states but also corrupted by white noise. Kalman filter is statistically optimal with respect to the quadratic loss function of the estimation errors. In this section, we will start from the basic linear system theory and then step further to the state space models. As a result, Kalman filter is illustrated for the state estimation of state space models.

2.2.1 Linear systems and state space models

Since the first application in the planet motion modeling by Newton, differential equation has been the most important and concise mathematical models for many dynamic systems. A first order ordinary differential equation (ODE) can be written as

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{C}(t)\mathbf{u}(t) \quad (2.43)$$

where $\mathbf{x}(t)$ is the deterministic state vector at time t and $\mathbf{u}(t)$ is the system input. In fact, high order ODE can often be converted into the first order ODE by reformulating state variables as a vector. To solve this equation, it is often first considering the homogeneous case i.e. $\mathbf{u}(t) = 0$. A state transition matrix (evolution operator) from t to τ can be defined for the homogeneous case

$$\Phi(\tau, t) = \phi(\tau)\phi^{-1}(t) \quad (2.44)$$

where $\phi(t)\mathbf{x}(0) = \mathbf{x}(t)$ and $\Phi(\tau, 0) = \phi(\tau)$. Using the matrix Φ , we can obtain the solution to the inhomogeneous ODE (2.43)

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{C}(\tau)\mathbf{u}(\tau)d\tau. \quad (2.45)$$

When the system is time invariant or stationary, the transition matrix will only depend on the time interval between t and τ , i.e.

$$\Phi(t, \tau) = e^{\mathbf{F}(t-\tau)}, \quad (2.46)$$

so the solution of the ODE can be derived as

$$\mathbf{x}(t) = e^{\mathbf{F}(t-\tau)}\mathbf{x}(t_0) + \int_{t_0}^t e^{\mathbf{F}(t-\tau)}\mathbf{C}(\tau)\mathbf{u}(\tau)d\tau. \quad (2.47)$$

Due to the popularity of modern digital computer, it becomes much easier and more convenient to solve the discretized systems or continuous systems by numerical methods. Using

first order approximation, a continuous system such as equation (2.43) can be represented as a discrete linear equation

$$\mathbf{x}_k = \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{C}_{k-1}\mathbf{u}_{k-1}. \quad (2.48)$$

Having considered the system representation, we still need to measure the system outputs for the estimation of system states. This leads us to the measurement equation

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \quad (2.49)$$

which is often modeled as a linear combination of system states and the input vectors. For a discrete system, it can be rewritten as

$$\mathbf{z}_k = \mathbf{H}_{k-1}\mathbf{x}_{k-1} + \mathbf{D}_{k-1}\mathbf{u}_{k-1}. \quad (2.50)$$

In real application, the system state sequence is usually a stochastic process, which is corrupted by noises. Meanwhile, the measured output may also involve some noise process. In general, the continuous system can then be formulated as

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{C}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) + \mathbf{v}(t) \end{aligned} \quad (2.51)$$

where $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are white noise process. The system becomes a stochastic differential equation and can not be solved by ordinary calculus. *Itô* calculus is often used to solve such type differential equations. On the other hand, the discrete model can be represented as a standard state space model

$$\begin{aligned} \mathbf{x}_k &= \Phi_{k-1}\mathbf{x}_{k-1} + \mathbf{C}_{k-1}\mathbf{u}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_{k-1}\mathbf{x}_{k-1} + \mathbf{D}_{k-1}\mathbf{u}_{k-1} + \mathbf{v}_{k-1} \end{aligned} \quad (2.52)$$

where \mathbf{w}_k and \mathbf{v}_k are white noise sequences. In practice, the noise in the plant and observer may be colored, then shaping filters can be used to convert the state space model into another form in which noise sequences are white.

2.2.2 Kalman filter algorithms

Kalman filter is designed to solve state estimation problem by using measurements that are linear functions of the states. The control input $\mathbf{u}(t)$ is often not considered in estimation

problems. The remainder of this section will simply pave a road to the final derivation of Kalman filtering algorithms without going much to the theoretical details.

Kalman filter is originally developed to solve discrete systems, which happen in our common practice in engineering. A continuous version of Kalman filter is Kalman-Bucy filter and will not be treated within this text. Our main concern here is to estimate the states based on observation of the system outputs, and the plant and measurement models can be summarized as a special form of the previous general state space model

$$\begin{aligned}\mathbf{x}_k &= \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_{k-1}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}\end{aligned}\tag{2.53}$$

where \mathbf{w}_k and \mathbf{v}_k are white noise sequences with

$$\begin{aligned}\mathbb{E}[\mathbf{x}_k] &= \mathbb{E}[\mathbf{v}_k] = 0 \\ \mathbb{E}[\mathbf{w}_k\mathbf{w}_i] &= \Delta(k-i)\mathbf{R}_k^1 \\ \mathbb{E}[\mathbf{v}_k\mathbf{v}_i] &= \Delta(k-i)\mathbf{R}_k^2\end{aligned}$$

and \mathbf{x}_k and \mathbf{w}_k are the $n \times 1$ vectors whereas \mathbf{z}_k and \mathbf{v}_k are the $l \times 1$ vectors. Meanwhile, $\mathbf{\Phi}_k$ and \mathbf{G}_k are $n \times n$ matrices whereas \mathbf{H}_k is a $l \times n$ matrix. \mathbf{R}_k^1 and \mathbf{R}_k^2 are auto-covariance or auto-correlation matrix of the plant and measurement noise sequences and will be constant matrices \mathbf{R}^1 and \mathbf{R}^2 if the process is stationary. $\Delta(\cdot)$ is the *Kronecker delta function*. In real application, the noise sequences are often assumed to be white Gaussian processes and the initial value \mathbf{x}_0 is known with respect to the mean and covariance matrix.

After formulating the state space model above, the main problem becomes to find the linear Minimum Mean Square Error (MMSE) estimator $\hat{\mathbf{x}}(k|k)$ i.e. \mathbf{x}_k given the observation sequence \mathbf{z}_k , $1 \leq k \leq n$. To estimate $\hat{\mathbf{x}}(k|k)$ based on observation \mathbf{z}_k , we are interested in finding an optimal linear combination of \mathbf{z}_k and the a priori estimate $\mathbf{x}(k|k-1)$, which a linear estimation of \mathbf{x}_k give the observation of $\mathbf{z}_0 \cdots \mathbf{z}_{k-1}$, that is

$$\hat{\mathbf{x}}(k|k) = \Theta_k\hat{\mathbf{x}}(k|k-1) + \Gamma_k\mathbf{z}_k.\tag{2.54}$$

This optimal linear estimate will be equivalent to the general nonlinear optimal estimator if the variates \mathbf{x} and \mathbf{z} are jointly Gaussian.

Based on the orthogonal condition

$$\mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}(k|k))\mathbf{z}_i^T] = 0 \text{ for } i = 1, 2, \dots, k\tag{2.55}$$

equation (2.53) and (2.54), and the assumption that the system noise and the measurement noise are uncorrelated, that is

$$\mathbb{E}[\mathbf{w}_i \mathbf{v}_i] = 0, \quad (2.56)$$

it can be showed that the following equation

$$\Theta_k = \mathbf{I} - \Gamma_k \mathbf{H}_k \quad (2.57)$$

has to be fulfilled. To further derive Kalman filter algorithm, some notations are necessary to be introduced

$$\begin{aligned} \tilde{\mathbf{x}}(k|l) &\doteq \hat{\mathbf{x}}(k|l) - \mathbf{x}_k \\ \mathbf{P}(k|l) &\doteq \mathbb{E}(\tilde{\mathbf{x}}(k|l)\tilde{\mathbf{x}}(k|l)^T) \end{aligned} \quad (2.58)$$

which is the error covariance matrix, and

$$\begin{aligned} \tilde{\mathbf{z}}_k &\doteq \hat{\mathbf{z}}(k|k-1) - \mathbf{z}_k \\ &= \mathbf{H}_k \hat{\mathbf{x}}(k|k-1) - \mathbf{z}_k. \end{aligned} \quad (2.59)$$

Since $\hat{\mathbf{x}}(k|k)$ depends linearly on \mathbf{x}_k , which also depends linearly on \mathbf{z}_k , we can obtain the following relation

$$\mathbb{E}[(\mathbf{x}_k - \hat{\mathbf{x}}(k|k))\tilde{\mathbf{z}}_k^T] = 0. \quad (2.60)$$

Expanding the equation above, we can finally obtain

$$\begin{aligned} \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \Gamma_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}(k|k-1)) \\ \Gamma_k &= \mathbf{P}(k|k-1)\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}(k|k-1)\mathbf{H}_k^T + \mathbf{R}_k^2]^{-1} \end{aligned} \quad (2.61)$$

where Γ_k is often called Kalman gain in this form of notation. Meanwhile, from the state space model of equation (2.53) and the orthogonal principle, the following relation exists

$$\hat{\mathbf{x}}(k|k-1) = \Phi_{k-1}\hat{\mathbf{x}}(k-1|k-1), \quad (2.62)$$

from which the a priori covariance matrix can be derived as

$$\begin{aligned} \mathbf{P}(k|k-1) &= \mathbb{E}[\tilde{\mathbf{x}}(k|k-1)\tilde{\mathbf{x}}(k|k-1)^T] \\ &= \Phi_{k-1}\mathbf{P}(k-1|k-1)\Phi_{k-1}^T + \mathbf{G}_{k-1}\mathbf{R}_{k-1}^1\mathbf{G}_{k-1}^T. \end{aligned} \quad (2.63)$$

Still, we need to consider how to compute the posterior error covariance matrix $\mathbf{P}(k|k)$. According to the equation (2.61), we get

$$\begin{aligned}\tilde{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k) - \mathbf{x}_k \\ &= (\mathbf{I} - \Gamma_k \mathbf{H}_k) \tilde{\mathbf{x}}(k|k-1) + \Gamma_k \mathbf{v}_k\end{aligned}\quad (2.64)$$

and that bring us the following equation on the update of the posterior covariance matrix

$$\begin{aligned}\mathbf{P}(k|k) &= \mathbb{E}[\tilde{\mathbf{x}}(k|k) \tilde{\mathbf{x}}(k|k)^T] \\ &= (\mathbf{I} - \Gamma_k \mathbf{H}_k) \mathbf{P}(k|k-1) (\mathbf{I} - \Gamma_k \mathbf{H}_k)^T + \Gamma_k \mathbf{R}_k^2 \Gamma_k^T.\end{aligned}\quad (2.65)$$

By substituting Γ_k by the equation (2.61), we get

$$\mathbf{P}(k|k) = (\mathbf{I} - \Gamma_k \mathbf{H}_k) \mathbf{P}(k|k-1).\quad (2.66)$$

Finally, the Kalman filtering algorithm is summarized in table 2.1 with both state update and measurement update procedures. Kalman filter has different versions according to the application purposes: filtering, prediction and smoothing. In the real time filtering, the algorithm is used to estimate the online states i.e. $\hat{\mathbf{x}}(k|k)$. In the prediction, it is applied to predict the future states using current measurements i.e. $\hat{\mathbf{x}}(k+m|k)$. In the fix-interval smoothing, the Kalman algorithm is further developed to estimate the states after obtaining information of a known process i.e. $\hat{\mathbf{x}}(k|N)$ with $0 \leq k \leq N$. It is very useful algorithm in the off-line noise cancellation if a state-space model is identified.

Kalman filter is an essential part of adaptive filter theory [22], especially its extended versions, e.g. extended Kalman filter and Unscented Kalman filter, has abilities to deal with nonlinear systems. Thanks to its iterative nature, it becomes powerful and efficient as a machine learning algorithm or a calibration method in modeling. We have in our research extensively applied the Kalman filter algorithm and its extensions.

2.3 Artificial neural networks

It is well known that modern computers have the Von Neumann architecture and outperform the human brain in numerical computations and symbol manipulations. However, humans can easily solve complex problems but difficult for a computer such as driving a car in different traffic conditions, which dwarf the fastest computer in the world. Inspired by the

State update:

$$\begin{aligned}\hat{\mathbf{x}}(k|k-1) &= \Phi_{k-1}\hat{\mathbf{x}}(k-1|k-1) \\ \mathbf{P}(k|k-1) &= \Phi_{k-1}\mathbf{P}(k-1|k-1)\Phi_{k-1}^T + \mathbf{G}_{k-1}\mathbf{R}_{k-1}^1\mathbf{G}_{k-1}^T\end{aligned}$$

Measurement update:

$$\begin{aligned}\Gamma_k &= \mathbf{P}(k|k-1)\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}(k|k-1)\mathbf{H}_k^T + \mathbf{R}_k^2]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \Gamma_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}(k|k-1)) \\ \mathbf{P}(k|k) &= (\mathbf{I} - \Gamma_k\mathbf{H}_k)\mathbf{P}(k|k-1)\end{aligned}$$

Table 2.1: The standard Discrete-Time Kalman Filter algorithm

functions and structures of biological neural network systems, researchers from a number of fields designed the *artificial neural networks* (ANN) or neural networks model to solve problems in pattern classification, clustering, function approximation, time series predictions and so on. ANN is modeled as a parallel and distributed machine consisting of a huge number of neural processors, between which connections are existed. Basically, ANNs are characterized by their architectures, neuron models and learning algorithms. Many texts, such as [23], [28] and [15], are available to introduce the ANN in details. In this section, only the principles of *multilayer feed-forward neural networks* (MLFNN) and *radial basis function* neural network (RBFNN) are presented. The feed-forward type neural network will be introduced first with the back-propagation training algorithm. The principles of RBF neural model is then explained with prototype initialization and learning methods. Finally, a brief overview of new learning algorithms for NN will be given.

2.3.1 Multilayer feed-forward neural networks

Network architecture

A 3-layer feed-forward Neural Network is shown in figure 2.3. Normally, there are three types of layers: input layer, hidden layer and output layer. Only one layer of input neuros and one layer of output neuros are used in the ANN model. But the number of hidden layers can be defined by the users according to their purposes. There are n neurons and

a bias node in the input layer; p hidden neurons and a bias node in the hidden layers; m output neurons in the output layer.

Computational model of neurons

Similar to the biological neuro cells, artificial neurons are information processing units. The neurons collect information from the input signals based on the following formula:

$$net_j = \mathbf{W}_j^T \cdot \mathbf{X} + b_j = \sum_{i=1}^n w_{ij} X_i + b_j$$

where,

net_j = the net input to the hidden neuron Z_j ;

$\mathbf{X} = [X_1, \dots, X_i, \dots, X_n]^T$ the input vector;

$\mathbf{W}_j = [w_{1j} \dots w_{ij} \dots w_{nj}]^T$ the weight vector on the connections to the neuron Z_j toward neuron j .

At the hidden and output neurons, an activation function, $Z_j = f(net_j)$, is defined in terms of the activity level at its input. In the literatures, a number of different function forms are adopted. One of the most typical activation function is in the form of:

$$f(net_j) = \frac{1}{1 + e^{-\sigma_j net_j}} \quad (2.67)$$

The activation function is monotonically non-decreasing and its output range can be scaled arbitrarily. The derivative of the activation function can be easily represented as:

$$f'(net_j) = \sigma_j f(net_j)(1 - f(net_j)). \quad (2.68)$$

Back-propagation training algorithms

The design purpose of ANN is to adjust the weights of the network connections during learning the input patterns and corresponding targets. The networks can be trained by the feedforwarding information processing and sequentially backwarding error corrections. The training algorithm of multiple layer feedforward networks in table 2.2 is derived by minimizing the sum of the square of the errors at each output neuron for each training sample, which can be represented by the objective function:

$$E_p = \frac{1}{2} \sum_{k=1}^m (t_k - Y_k)^2. \quad (2.69)$$

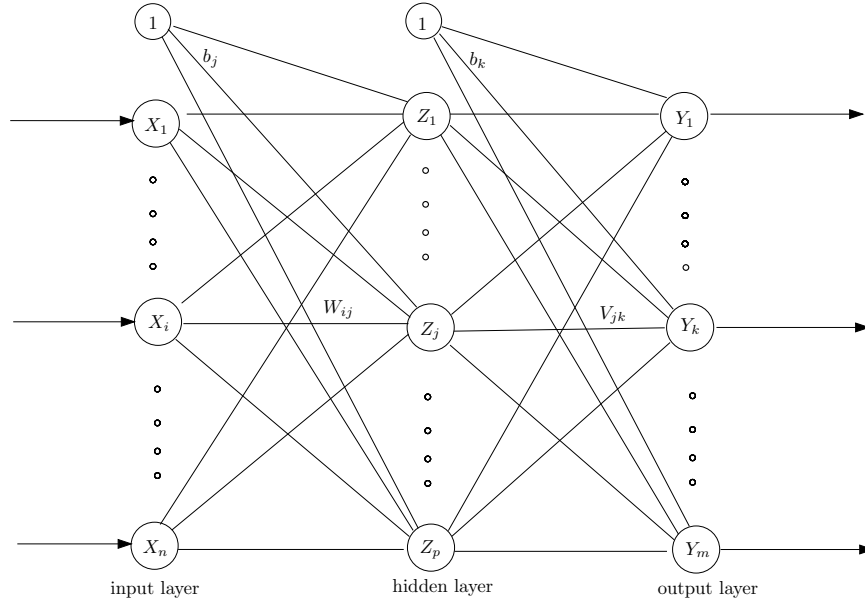


Figure 2.3: The architecture of a simplified 3-layer feedforward artificial neural network model.

Hence, by gradient descent searches the minimum of the objective function can be found iteratively. The gradient with respect to the v_{jk} can be computed by:

$$\begin{aligned}
 \frac{\partial E_p}{\partial v_{jk}} &= -(t_k - Y_k) \frac{\partial}{\partial w_{jk}} f(\text{net}_k) \\
 &= -(t_k - Y_k) f'(\text{net}_k) \frac{\partial}{\partial v_{jk}} (\text{net}_k) \\
 &= -\delta_k Z_j
 \end{aligned} \tag{2.70}$$

where $\delta_k = (t_k - Y_k) f'(\text{net}_k)$.

The gradient with respect to w_{ij} can be derived from the chain rule as follow:

$$\begin{aligned}
 \frac{\partial E_p}{\partial w_{ij}} &= -\sum_k (t_k - Y_k) f'(\text{net}_k) \frac{\partial}{\partial w_{ij}} (\text{net}_k) \\
 &= -\sum_k \delta_k \frac{\partial}{\partial w_{ij}} (\text{net}_k) \\
 &= -\sum_k \delta_k \frac{\partial \text{net}_k}{\partial Z_j} \frac{\partial Z_j}{\partial v_{ij}}
 \end{aligned}$$

initialize the weights of the connections (w_{ij} and v_{jk})
while stopping condition is false do
 for each training pair
 Feedforward information processing:
 Hidden layer: $net_j = \mathbf{W}^T \mathbf{X} + b_j$, $H_j = f(net_j)$;
 Output layer: $net_k = \mathbf{V}^T \mathbf{H} + b'_k$, $Y_k = f(net_k)$;

Backward error propagations:
 Output layer: $\delta_k = (t_k - Y_k) f'(net_k)$ (the error information)
 $\Delta w_{jk} = \alpha \delta_k z_j$, $\Delta b'_k = \alpha \delta_k$.
 Hidden layer: $\delta_j^b = \sum_{k=1}^m \delta_k w_{jk}$ (gather back-propagated errors)
 $\delta_j = \delta_j^b f'(net_j)$;
 $\Delta v_{ij} = \alpha \delta_j x_i$, $\Delta b_j = \alpha \delta_j$.

Updating weights and biases:
 $w_{jk} = w_{jk} + \Delta w_{jk}$; $v_{ij} = v_{ij} + \Delta v_{ij}$.

end for
end while

Table 2.2: Back-propagation training algorithms for multi-layer feed-forward Neural Networks.

$$\begin{aligned}
 &= - \sum_k \delta_k v_{jk} f'(net_j) X_i \\
 &= -\delta_j X_i
 \end{aligned} \tag{2.71}$$

where $\delta_j = \sum_k \delta_k v_{jk} f'(net_j)$.

2.3.2 Radial basis function neural network

Radial Basis Function (RBF) neural network was presented as a neural network model in 1980's and since then have been applied in different applications such as pattern recognition and function approximation. However, the theory behind the RBF network can be attributed to the linear model taking the form of

$$f(\mathbf{x}) = \sum_{i=1}^m w_i \phi_i(\mathbf{x}) \tag{2.72}$$

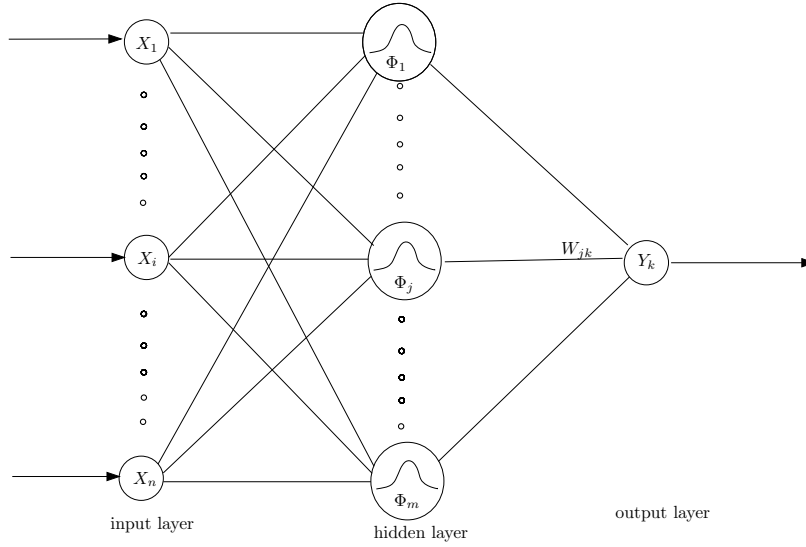


Figure 2.4: The architecture of radial basis function neural network model.

where w_i is a weight and $\phi_i(\cdot)$ is any basis function e.g. sinusoid, polynomial and so on. These directly relate it to the theory of Fourier series and numerical interpolation. The fact of its origin from a linear model makes the RBF network possess an advantage over nonlinear models e.g. multiple layer back-propagation networks, i.e. given the training data it is possible to derive and solve an equation set for estimation of connection weights.

RBF neural network can in general be understood as a method for the recovery of the hyperplane space from the existed training dataset. A special class of function, Gaussian function (other function, e.g bell function, can also be used) is adopted in the form of

$$\phi_i(\mathbf{x}) = \exp[-(\mathbf{x} - \mu_i)^T \Sigma_i (\mathbf{x} - \mu_i)] \quad (2.73)$$

where μ_i and Σ_i are the mean vector and covariance matrix of the i th Gaussian basis function. The architecture of RBF neural network can be illustrated in figure 2.4. An obvious difference between RBF and MLP network is that RBF has only three layers and connection weights exist only between hidden layer and output layer. Being a linear model, an overdetermined equation system can be formulated given the training data vectors $\{\mathbf{x}_1, \dots, \mathbf{x}_p\}$

$$\Phi \cdot \mathbf{w} = \mathbf{y} \quad (2.74)$$

where Φ , the design matrix, is

$$\Phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_m(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_m(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_p) & \phi_2(\mathbf{x}_p) & \cdots & \phi_m(\mathbf{x}_p) \end{bmatrix} \quad (2.75)$$

with $p \gg m$ and this is easy to solve by the least square method introduced early in this chapter with the objective to minimize

$$\mathbf{J} = \sum_{i=1}^p (y_i - f(\mathbf{x}_i))^2. \quad (2.76)$$

In real application, the conflict between bias and variance may lead to the problem of ill-conditioning, i.e. the bias in the equation (2.76) is minimized but the variance (probably caused by noise in the data) makes the identified model very poor in prediction. In the book of Haykin [23], the regularization theory is introduced in much detail and the ill-condition problem is solved by adding a constraint on the parameters so that the supplement of prior information can satisfy the balance between model bias and model variance, i.e.

$$\mathbf{J} = \sum_{i=1}^p (y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{k=1}^m w_k^2. \quad (2.77)$$

This optimization problem leads to a new linear system

$$\Phi^e \cdot \mathbf{w} = (\Phi + \lambda \mathbf{I}) \cdot \mathbf{w} = \mathbf{y}. \quad (2.78)$$

The selection of λ is quite difficult since it is problem oriented. Different cross-validation criterion can be adopted but a strategy of trial-and-error may be necessary. In real application, fast training methods that separate the tasks of radial basis determination and weight optimization were extensively adopted, though it might not take full advantage of the information of the training data. In the weight optimization, an online learning algorithm is widely applied for training the RBF model and delta rule [23] is the common choice. Table 2.3 shows a typical learning algorithm for the RBF neural network based on the delta rule and η is the learning rate constant.

Another essential problem for RBF networks is how to initialize the RBF neural networks, i.e. choose the number of basis functions and identify the function parameters (mean and standard deviation). This is in fact the research front of RBF neural network. Several

```

initialize the weights of the connections ( $\mathbf{w}_0 = \{w_{jk}^0\}$ )
while stopping condition is false do
  for each training pair ( $\mathbf{x}_t, y_t$ )
    Feed-forward information processing:
    Hidden layer:  $\Phi^e(\mathbf{x}_t) = [\phi_1^e(\mathbf{x}_t) \cdots \phi_m^e(\mathbf{x}_t)]^T$ ;
    Output layer:  $\hat{y} = \Phi^e(\mathbf{x}_t)^T \mathbf{w}_t$ ;
    Delta rule training phase:
     $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta(y - \hat{y})\Phi^e(\mathbf{x}_t)$ .
  end for
end while

```

Table 2.3: The algorithms of delta rule for training RBF Neural Networks.

approaches has been proposed, and one of them is to apply a competitive learning algorithm used in self-organization map (SOM), which is presented in many literature e.g. [15]. The basic idea is to cluster data into groups that the data is distributed. To fulfill this procedure, other clustering methods, e.g. K-means algorithm, are also applicable. A recent idea is to initialize the RBF neural network by decision tree [40].

2.3.3 Innovative training algorithms

In the last subsections, it was showed that gradient based methods (back-propagation and delta rule) could be applied in machine learning of neural network models. However, these methods normally adopt first order gradient-based approach and has relatively slower convergence and therefore computationally more expensive. In practice, various second order derivative based methods including recursive least square, conjugate gradient method and Kalman filter have been extensively applied in the learning phase of difference neural network models (e.g. [56] [60] [54] [57] and [67]). These approaches show superior performance than classic training algorithms e.g. back-propagation and delta rule, in both faster convergence rate and lower minima. Another research thread in the learning of neural networks is to apply evolution-based methods e.g. Genetic algorithm. This approach has advantages in more successful finding of global optimum but suffers drawbacks of slow convergence and lengthy computational time.

2.4 Fuzzy sets and inference principles

Fuzzy set and fuzzy logic are probably among the most important concepts of CI. There are many books describing the fuzzy logic and inference principle in details such as [28], [71] and [63].

2.4.1 Fuzzy sets

As the name implies, a fuzzy set is a set without a crisp boundary. A membership function is defined for the fuzzy set to map each member to a certain membership grade. Mathematically, a fuzzy set \tilde{A} is defined as:

$$\tilde{A} \equiv \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$$

where x is a set element, $\mu_{\tilde{A}}$ is the membership function and X is the universe of discourse. Based on the definition, the elementary set-theoretic representation and operators shall be redefined correspondingly ³:

- *Integral*: a fuzzy set \tilde{A} is often represented by the integral or collection of pairs of set elements and their corresponding fuzzy memberships, that is:

$$\tilde{A} = \int_X \frac{\mu_{\tilde{A}}(x)}{x};$$

- *Subset*: a fuzzy set \tilde{A} is a subset of a fuzzy set \tilde{B} if and only if $\mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x)$, that is:

$$\tilde{A} \subseteq \tilde{B} \iff \mu_{\tilde{A}}(x) \leq \mu_{\tilde{B}}(x).$$

- *Disjunction*: the union or sum of two fuzzy sets, $\tilde{A} \cup \tilde{B}$, has the membership function:

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \oplus \mu_{\tilde{B}}(x)$$

where \oplus is the fuzzy sum operator that can be defined as for example $\max\{\cdot, \cdot\}$.

- *Conjunction*: the intersection or product of two fuzzy sets, $\tilde{A} \cap \tilde{B}$, has the membership function:

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \otimes \mu_{\tilde{B}}(x)$$

where \otimes is the fuzzy product operator that can be defined as for example $\min\{\cdot, \cdot\}$.

³Some variations of notations from the conventional logic and set operations are used to highlight the difference.

- *Negation*: the complement of a fuzzy set \tilde{A} , denoted by $\neg\tilde{A}$ can be defined by relations e.g.

$$\mu_{\neg\tilde{A}} = 1 - \mu_{\tilde{A}}.$$

Unlike the classic set, fuzzy set operators including conjunction, disjunction and negation can be defined by the user himself according to application context. The introduction of the concept of fuzzy set brings us flexibility but also sometimes confusion in selection of set operators. Fortunately, the criteria has been described in the text of [71] which suggests considering eight facets of the problem: axiomatic strength, empirical fit, adaptability, numerical efficiency, compensation and its range, aggregating and scale level of the membership function. In some literatures, the fuzzy union and intersection operators are generalized to T-conorm (or S-norm) and T-norm operator. The different definition of these operators gives different basic properties of the consequent fuzzy mathematics.

2.4.2 Fuzzy relations and reasoning

With the fuzzy set and redefined set operators, the relation between two sets evolves as:

$$\tilde{\mathbf{R}} \equiv \{(x, y), \mu_{\tilde{\mathbf{R}}}(x, y) | (x, y) \in X \times Y\}$$

where the membership function of the relation is based on the well-known extension principle [71]:

If function f is a mapping from a n-dimensional Cartesian product space $\mathbf{X} = X_1 \times X_2 \times \dots \times X_n$ to an one-dimensional universe Y such that $y = f(x_1, x_2, \dots, x_n)$ and $\tilde{A}_1, \dots, \tilde{A}_n$ be fuzzy sets in X_1, \dots, X_n respectively. Then we can define a fuzzy set \tilde{B} in Y by $\tilde{B} = \{(y, \mu_{\tilde{B}}(y)) | y = f(x_1, \dots, x_n), (x_1, \dots, x_n) \in \mathbf{X}\}$ with

$$\mu_{\tilde{B}}(y) = \begin{cases} \sup_{(x_1, \dots, x_n) \in f^{-1}(y)} \{\mu_{\tilde{A}_1} \otimes \mu_{\tilde{A}_2} \otimes \dots \otimes \mu_{\tilde{A}_n}\} & \text{if } f^{-1}(y) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Fuzzy relations

Based on the extension principle above, it is possible to compose the binary relation to a fuzzy mapping between fuzzy sets and figure 2.5 shows a composition of two fuzzy relations

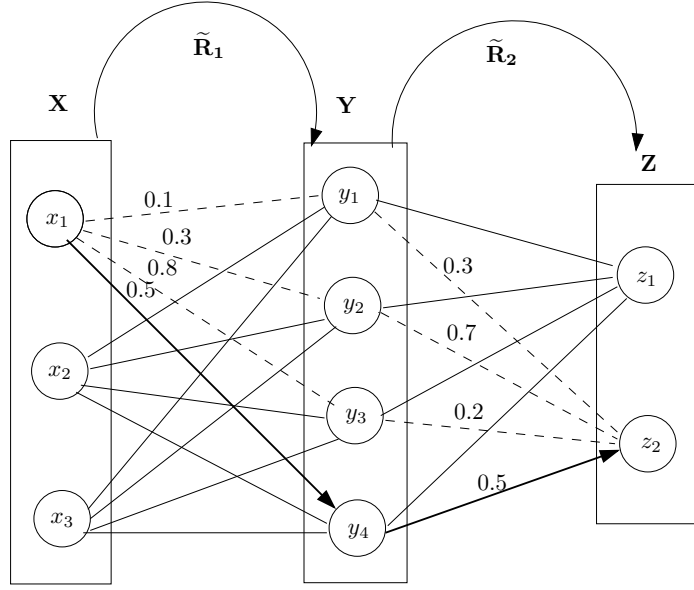


Figure 2.5: Composition of two fuzzy relations.

$\tilde{\mathbf{R}}_1 \circ \tilde{\mathbf{R}}_2$, which can be defined as:

$$\tilde{\mathbf{R}}_1 \circ \tilde{\mathbf{R}}_2 = \{[(x, z), \sup_y \{\mu_{\tilde{\mathbf{R}}_1} \otimes \mu_{\tilde{\mathbf{R}}_2}\}] | x \in \mathbf{X}, y \in \mathbf{Y}, z \in \mathbf{Z}\}.$$

Fuzzy if-then rules and reasoning

In the conventional logic, implication or inference is an operator between two sets. For example, we can assume the rule form:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B.$$

where A and B are crisp sets. This can be written as $A \rightarrow B$, which is always explained as $\neg A \cup B$ in basic logical operations. With fuzzy sets \tilde{A} and \tilde{B} , however, we can either derive the fuzzy implication by using

$$\neg \tilde{A} \cup \tilde{B} = \int_{X \times Y} \frac{\mu_{\tilde{A} \rightarrow \tilde{B}}(x, y)}{(x, y)} = \int_{X \times Y} (1 - \mu_{\tilde{A}}) \oplus (\mu_{\tilde{A}} \otimes \mu_{\tilde{B}}) / (x, y)$$

or other implication functions introduced in [28].

Based on the quantification of fuzzy implication, it is possible to apply approximate reasoning with computation of compatibility degrees, e.g., giving a fuzzy input and a fuzzy

rule it is possible to derive the consequence as a fuzzy set. For example, if we have the fuzzy rule:

$$\text{IF } x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B} \text{ THEN } z \text{ is } \tilde{C}.$$

and the fact that x is \tilde{A}' and y is \tilde{B}' , we can first transform the rule as a fuzzy relation mentioned above based on Mandani's fuzzy implication function as follow:

$$\tilde{\mathbf{R}}_M(A \times B \rightarrow C) = \int_{X \times Y \times Z} \mu_{\tilde{A}}(x) \otimes \mu_{\tilde{B}}(y) \otimes \mu_{\tilde{C}}(z) / (x, y, z). \quad (2.79)$$

Then, with fuzzy input \tilde{A}' and \tilde{B}' and the derived fuzzy relation above we can inference the consequence as follow:

$$\begin{aligned} C' &= (A' \times B') \circ (A \times B \rightarrow C) \\ &= \int_Z \sup_{x,y} \{ \mu_{\tilde{A}'}(x) \otimes \mu_{\tilde{B}'}(y) \otimes \mu_{\tilde{\mathbf{R}}_M(A \times B \rightarrow C)}(x, y, z) \} / z. \end{aligned} \quad (2.80)$$

It can be proved that the equation (2.80) is equivalent to

$$C' = \int_Z w_1 \otimes w_2 \otimes \mu_{\tilde{C}}(z) / z \quad (2.81)$$

where $w_1 = \sup_x \{ \mu_{\tilde{A}'}(x) \otimes \mu_{\tilde{A}}(x) \}$ and $w_2 = \sup_y \{ \mu_{\tilde{B}'}(y) \otimes \mu_{\tilde{B}}(y) \}$ are the firing strengths or compatibility degrees between the fuzzy input and the antecedents of the fuzzy rule. Using the same principles we can combine the fuzzy rule consequences by fuzzy T-conorm if there are multiple rules being fired by the inputs.

2.4.3 Fuzzy modeling

Based on the concepts reviewed before, the *fuzzy inference system* (FIS) serves as a popular modeling framework in many modern scientific fields. The basic principle of FIS is to construct a rule base which contains a selection of fuzzy rules, a database which defines the membership functions for the variables in the rule base, and a inference mechanism performing the reasoning procedures based on predefined fuzzy operations.

Linguistic variables

Before reviewing some common FIS systems, linguistic variables shall be introduced as an important concept. The invention of fuzzy set theory was partly aiming at applying human

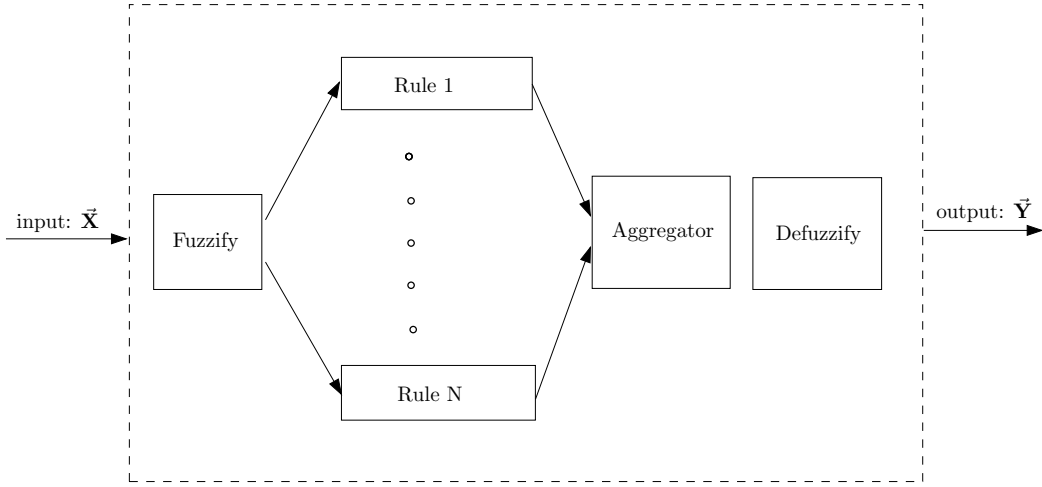


Figure 2.6: Structure of the typical Mamdani fuzzy inference systems.

knowledge in a computing environment and hence scientists needed to interpret the expert knowledge into a computable form. Different from the conventional symbolic approach, FIS reasons on human language by modeling the human linguistic meaning as a collection of fuzzy sets. For example, the linguistic variable 'age' can be represented as a collection of fuzzy sets $\{ 'young', 'middle', 'old' \}$. Therefore, linguistic variables serves an essential concept in the fuzzy inference systems.

Mamdani fuzzy models

Mamdani fuzzy models are one of the most commonly used fuzzy inference systems. It derive the consequence as a fuzzy set, therefore a defuzzification is needed to obtain a crisp numerical result as the output. Figure 2.6 shows the basic structure of a Mamdani FIS system. There are different ways to defuzzify the consequence such as centroid of the area, mean of the maximum and bisector of the area of the fuzzy consequence. Mathematically, the Mamdani fuzzy model using centroid of the area as defuzzification value can be written as:

$$f(\hat{x}) = \frac{\int_Z z \cdot \sup_{C_i} \{ R_{\{\tilde{A}_1 \times \dots \times \tilde{A}_m \rightarrow \tilde{C}_i\}}^i (\mu_{\tilde{A}_1} \otimes \dots \otimes \mu_{\tilde{A}_m}, \mu_{\tilde{C}_i}(z)) \} dz}{\int_Z \sup_{C_i} \{ R_{\{\tilde{A}_1 \times \dots \times \tilde{A}_m \rightarrow \tilde{C}_i\}}^i (\mu_{\tilde{A}_1} \otimes \dots \otimes \mu_{\tilde{A}_m}, \mu_{\tilde{C}_i}(z)) \} dz} \quad (2.82)$$

where R^i is the implication operation at rule i and $\sup_{C_i} \{R^i_{\{\tilde{A}_1 \times \dots \times \tilde{A}_m \rightarrow \tilde{C}_i\}}(\mu_{\tilde{A}_1} \otimes \dots \otimes \mu_{\tilde{A}_m}, \mu_{\tilde{C}_i}(z))\}$ gives the implication grade at z after examining all fuzzy rules.

Sugeno fuzzy models

Mamdani model gives clear linguistic meaning not only in the antecedent part of the fuzzy rule but also in the consequent part of the rule. But the defuzzification process is computationally very expensive in some applications. Therefore, Sugeno fuzzy models were invented to unload the computation due to the defuzzification. In Sugeno model, the fuzzy set at consequence part of the rule is replaced by a polynomial, e.g., the fuzzy rule may look like as follow:

$$\text{IF } x \text{ is } \tilde{A} \text{ and } y \text{ is } \tilde{B} \text{ THEN } z = \alpha x^l + \beta y^k.$$

Hence, a reduction of computation is realized by losing linguistic meaning of the model to some degree. Meanwhile, determination of the parameters is always important but not very easy.

Hybrid systems

The generic fuzzy modeling methodology can be summarized into four procedures as follows:

- Choose the inputs and outputs;
- Determine the FIS type and the number of linguistic terms (fuzzy sets) and membership functions associated with each input and output;
- Determine the fuzzy rule set from expert knowledge or identify them by methods e.g. simple clustering methods;
- Refine the membership functions by optimization techniques such as genetic algorithm (GA).

The second and third steps are referred to fuzzy knowledge base construction, which is nowadays often explored from data based on automatic optimization methods such as GA. This leads to the Genetic-Fuzzy system (GFS) [11], a cutting-edge research topic in AI. In addition, tuning of membership functions can also be included in building general fuzzy knowledge base, which leads to a global optimum approach [11]. Another extensively used

method is to represent fuzzy system by Neural Networks and then optimize the fuzzy systems based on training algorithms of Neural Networks using numerical or linguistic data, which results in Neural-Fuzzy System (NFS) and Genetic-Neural-Fuzzy systems (GNFS) [11] [28] [63].

2.5 Clustering techniques

Clustering is one the most primitive mental activities of human being, used to process huge amount of information we obtain everyday since comprehensive handling each piece information is an impossible task. Clustering scheme can categorize an information set into several information subsets, which we humans can efficiently deal with. This ability enables us to explore its principle and tries to apply that in building intelligent machines. In order to develop a clustering method, several steps are necessary: selecting features, determining proximity measure, giving an interpretable clustering criterion, designing the clustering algorithm, and explaining the results. Variety in those steps may result in different clustering consequence.

Algorithm design plays as the core of clustering analysis. Many clustering algorithms has been developed and used extensively to applications such as data classification, data encoding and model construction. Those algorithms partition an object set or a multivariate dataset into several groups such that the similarity within a group is larger than that among groups. Clustering algorithms can also be classified into three categories: sequential algorithms, Hierarchical algorithms and cost function based algorithms. A thorough explanation of all algorithms can be found in literature e.g. [61]. Because of the research contexts, the optimization function based clustering scheme is the focus of this section.

2.5.1 Bayesian approaches

Bayesian philosophy is a widely used reasoning scheme in many areas where statistical concepts are involved. In the basic task of clustering N vectors $\mathbf{x}_i, i = 1 \cdots N$ into m categories $C_j, j = 1 \cdots m$ or clusters, an obvious criterion is

$$A \text{ vector } \mathbf{x}_i \text{ is belonging to } C_j \text{ if } P(C_j|\mathbf{x}_i) > P(C_k|\mathbf{x}_i), \forall k = 1 \cdots m, k \neq j.$$

where $P(C_j|\mathbf{x}_k)$ is the conditional probability that the vector belongs to category j .

In the literature of [13], Bayesian decision theory is introduced for pattern classification problems given the prior information of the classifications. Thus, a decision boundary can be derived using the statistical information or pdfs [13] [61]. However, the prior knowledge such as the form or parameters or both of pdfs is often unknown in a classification context. One possible idea is to estimate the pdfs from data. For example, if the forms of pdfs are known, the parameters of distributions can be estimated through the Maximum Likelihood (ML) method i.e.

$$\hat{\theta}_{ML} = \arg \max_{\theta} \mathbf{p}(\mathbf{X}; \theta). \quad (2.83)$$

It is known that the ML estimator is asymptotically unbiased and consistent and achieves Cramer-Rao lower bound in its variance when $N \rightarrow \infty$ [33]. In the clustering context, however, no information is given for the labels of the data. Thus, the clustering problem can be treated as classification of incomplete data. According to Bayesian theory, the pdf of data can be approximated by a linear combination of the conditional pdfs in the form of

$$p(\mathbf{x}) = \sum_{j=1}^m p(\mathbf{x}|C_j)P_j \quad (2.84)$$

with the constraints

$$\sum_{j=1}^m P_j = 1 \quad (2.85)$$

$$\int_{\mathbf{x}} p(\mathbf{x}|C_j)d\mathbf{x} = 1. \quad (2.86)$$

Hence, the ML method can be applied with the logarithm likelihood function defined as

$$Q(\Theta) = \log \prod_i p(\mathbf{x}_i; \Theta, P_1, \dots, P_m) = \sum_{i=1}^N \log(p(\mathbf{x}_i|C_i; \Theta)P_{C_i}). \quad (2.87)$$

It is necessary first to choose the appropriate parametric form of $p(\mathbf{x}|C_j; \Theta)$. The maximization of the objective function with respect to Θ above can be solved by a number of nonlinear optimization techniques. The literature [61] introduces Expectation-Maximization (EM) algorithm due to its great advantages in smooth convergence and invulnerable stability.

2.5.2 Fuzzy clustering

The difficulty involved in the Bayesian approach is the probability density functions, for which suitable models have to be assumed. Meanwhile, the Bayesian approach is more

suited for the compact clusters, where data is smoothly and centrally distributed. In this section, another family of clustering algorithms, fuzzy clustering scheme, is reviewed. K-means clustering, also known as ISODATA [3], is an earliest and widely used unsupervised clustering algorithm to attribute multi-dimensional data into different clusters according to certain dissimilarity measures. In this method, the cluster number has to be determined at first and each data point can be either a member of a cluster (membership degree u is 1) or not a member of a cluster (membership degree u is 0). By randomly determining the initial cluster centers, the algorithm searches iteratively the optimal combination of the membership degree matrix and cluster centers in order to minimize the summation of dissimilarity measures of data vectors to its cluster center, that is,

$$\mathbf{J} = \sum_{i=1}^C \sum_{k \in C_i} \mathbf{D}(\mathbf{x}_k, \mathbf{c}_i) \quad (2.88)$$

where \mathbf{x}_k is the k -th data vector and \mathbf{c}_i is the cluster center of i -th cluster C_i . $\mathbf{D}(\cdot, \cdot)$ is the measure of dissimilarity, and the Euclidean distance $\mathbf{D}(\mathbf{x}_k, \mathbf{c}_i) = d_{ik}^2 = \|\mathbf{x}_k - \mathbf{c}_i\|^2$ is one of the mostly applied.

Fuzzy C-means clustering [5] generalizes the K-means algorithm with adoption of the fuzzy membership degree, a concept introduced in the fuzzy set theory. That is, each data vector can be attributed to several clusters with certain membership degrees between 0 and 1. But the summation of the membership degrees for each data vector must be 1. Hence the objective function becomes as follow:

$$\mathbf{J} = \sum_{i=1}^C \sum_{k=1}^N u_{ik}^m d_{ik}^2 \quad (2.89)$$

with the constraint

$$\sum_{i=1}^C u_{ik} = 1 \quad \forall i \in [1, N]. \quad (2.90)$$

Solving the optimization problem leads to the Fuzzy C-means algorithm in table 2.4. However, this algorithm is only local optimal and can find the global optimum only if the initial guess is close enough to the optimum point. Hence, multiple random initial guesses are often used in order to increase the chance to obtain the global optimum.

**Initialize a random membership matrix U with Eq. (2.90) fulfilled;
While stopping condition is false do**

1. calculate C cluster centers \mathbf{c}_i for $i = 1, \dots, C$ using

$$\mathbf{c}_i = \frac{\sum_{k=1}^N u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^N u_{ik}^m};$$

2. update the membership matrix U using

$$u_{ik} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{c=1}^C (d_{ck})^{-2/(m-1)}};$$

3. compute the objective function.

end do

Table 2.4: The Fuzzy C-means clustering algorithm.

2.6 Summary

In the last sections, many of the important components of modern AI techniques have been briefly covered. Unlike the initial flourishing development of different computational intelligence methods, modern AI has been evolved into a maturing stage with a tendency to forge those hammers in various applications. It is in the application context that all the developed tools can be utilized in a collaboration fashion and thus the theory of AI can make further progress. Therefore, the research methodology in modern AI fields has become problem-oriented with a hybrid of different techniques. Consequently, the overview of modern AI techniques in this chapter is limited as the soul of AI pervades in the broad content of various applications.

Chapter 3

Research summary

This chapter is a comprehensive summary of the research outcome achieved mainly in the selected research papers attached. To illustrate the research content in a structured form, the text follows research procedures and briefly covers each step with references to the corresponding papers. Six referred papers are attached behind.

3.1 Measurement on driver behavior

In order to study the driver patterns from real traffic and mimic them in a driver model, data collection is an essential procedure. There are a number of types of driving data acquired by different means, e.g. laser and radar sensor data, GPS data, and image data. An instrumented vehicle designed as a floating car has been adopted in our data collection experiment.

In *paper I* [49], a detailed description of the experiment hardware and software has been demonstrated. The equipped car was produced at Volvo technology in Göteborg. With a speedometer, GPS receiver and computer system installed, the vehicle can log the real-time information on its own states (acceleration, speed and position) as well as indexes of driver actions e.g. pedal and brake pressure and wheel angle. To observe other vehicles, two video cameras and laser sensors are concealed in the front and rear of the equipped car. Laser sensor is a widely available and inexpensive equipment, but very sensitive to poor weather and road conditions (blocked by fog and dirt in the air). However, the laser beam scans with a frequency of 50 Hz, high enough for tracking other objects. In addition, it has a full view range of 180 degrees and a long distance range of 100 m. The camera is connected

with a central computer and records images synchronized with real-time data.

The experiment design of driver behavioral data collection has been demonstrated in [47] and [49]. The measurement was conducted on a section of the Swedish motorway E18, north of Stockholm, composed of “2 + 2” four lane divided road and “2 + 1” three lane divided road. Behavior of more than 30 random followers was observed in 15 test runs and recorded through a data-fusion software, Volvo ERS logger. States of both vehicles in the driver-following stage were abstracted from raw data as multivariate time series.

3.2 Information estimation

After obtaining the state time series data of all pairs of vehicles, the true car-following information was estimated by canceling measurement noise using Kalman smoothing algorithm. This estimation procedure is illustrated in *paper I* [49] and [47]. The collected information of the following vehicles includes position, speed and acceleration. However, only the position is directly measured by laser scanning whereas the speed and acceleration are derived by difference equations and processed through a linear adaptive filtering algorithm enabled in Volvo ERS logger.

In the basic idea of vehicle tracking, a state-space model for leading and following vehicles can be formulated respectively. Vehicle dynamics can be easily represented by the well-known differential equation as follows

$$\frac{d^2s(t)}{dt^2} = a(t) \quad (3.1)$$

or in the discrete form i.e.

$$\begin{aligned} s(t) &= s(t-1) + v(t-1)\Delta t + \frac{1}{2}a(t-1)\Delta t^2 \\ v(t) &= v(t-1) + a(t-1)\Delta t. \end{aligned} \quad (3.2)$$

where Δt is the system time interval. After investigation on autocorrelation and partial autocorrelation functions of acceleration time series, it was modeled as a first order autoregressive (AR) process i.e.

$$a(t) = \phi a(t-1) + \theta_a(t-1) \quad (3.3)$$

where $\theta_a(t)$ is white noise and random walk model is adopted i.e. $\phi = 1$ as Δt is very small. Although the ERS logger has ability to adaptively filter the direct distance measurement

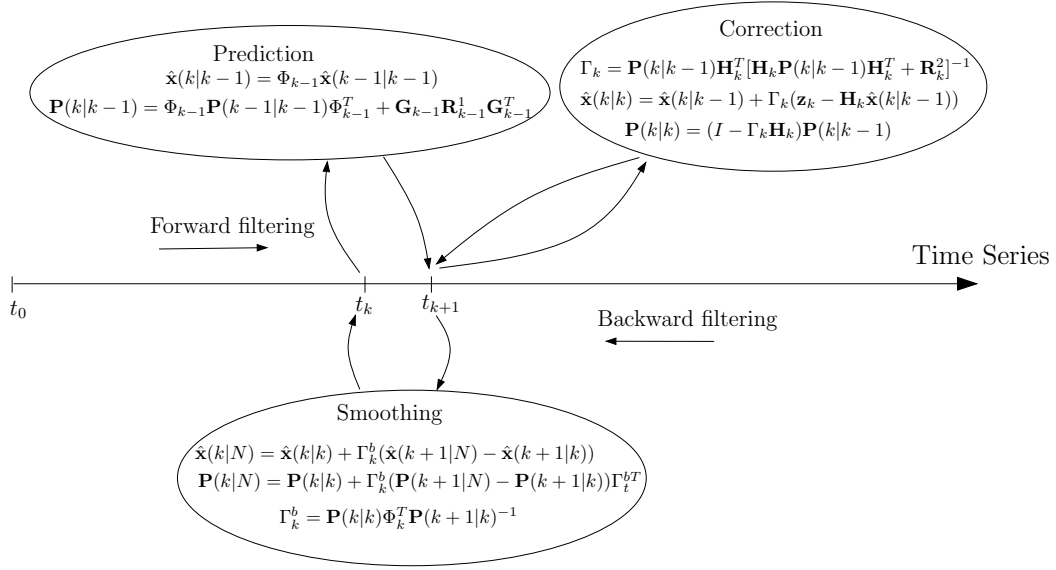


Figure 3.1: A diagram representation of driver-following data estimation procedures using Kalman filtering and smoothing techniques.

from the laser sensor and to estimate first and second order derivatives, the result is not satisfying as noise appears apparently in the speed and acceleration profiles. In *paper I*, a state-space model was formulated as follows

$$\mathbf{X}(t+1) = \mathbf{F} \cdot \mathbf{X}(t) + \mathbf{V}(t) \quad (3.4)$$

$$\mathbf{Y}(t) = \mathbf{H} \cdot \mathbf{X}(t) + \mathbf{W}(t). \quad (3.5)$$

where $\mathbf{X}(t) = [s_n(t) \ v_n(t) \ a_n(t)]^T$ is the state vector; $\mathbf{V}(t) = [\theta_s(t) \ \theta_v(t) \ \theta_a(t)]^T$ is the system noise vector; $\mathbf{Y}(t) = [\hat{s}_n(t) \ \hat{v}_n(t) \ \hat{a}_n(t)]^T$ is the measurement vector; $\mathbf{W}(t)$ is the measurement noise, whose dimension depends on \mathbf{H} . In the car-following estimation, $\mathbf{H} = [1 \ 0 \ 0]^T$ shows that only the trajectory measurement is considered as real measurement and

$$\mathbf{F} = \begin{pmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & \phi \end{pmatrix}$$

is the state transition matrix. Kalman filter and smoother were then applied in the data estimation process according to their ability to cope with the state-space model in an on-line

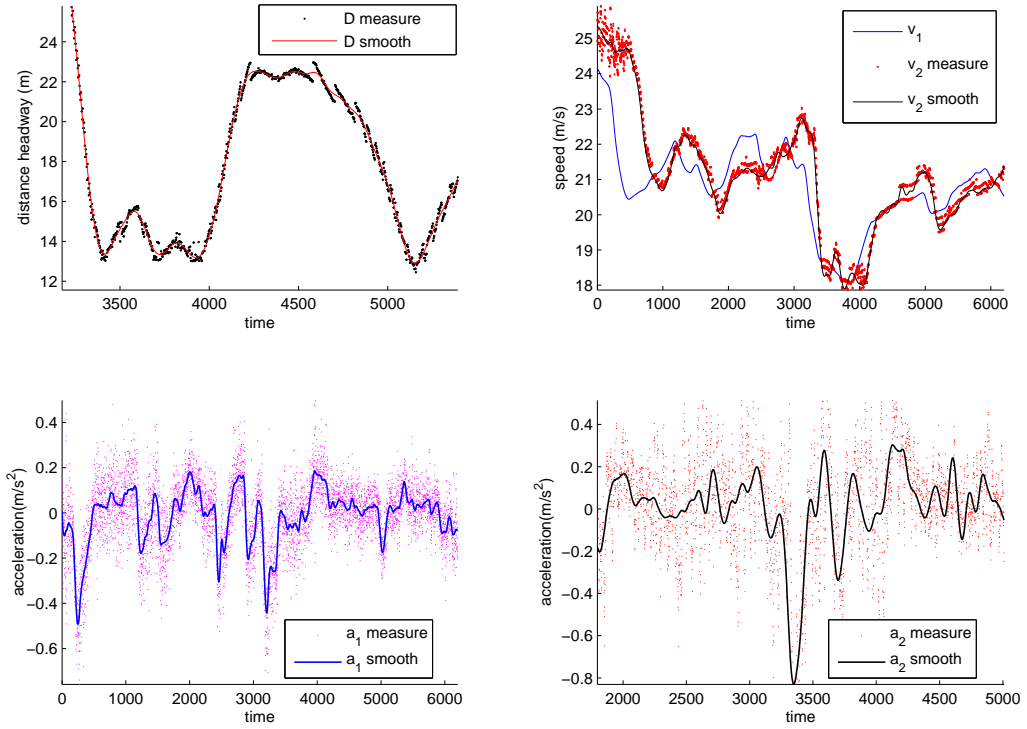


Figure 3.2: An example of smoothing result of car-following data using Kalman filter and smoother.

batch mode (in figure 3.1). One possible problem of this approach is that noise processes existed in the model and measurement are not directly known. However, according to experience, the performance of Kalman filters is determined mainly by the ratio between the noise power in the state and measurement equations. Therefore, a good state estimate can be obtained if the magnitude of the ratio between those two power levels is tested out. In *paper I*, a filter design approach is applied with the rather good estimation result in most of the cases partly because of the high measuring frequency in the experiment. Figure 3.2 presented one example of the estimation result of car-following states based on the method.

In addition to driver-following information estimation by noise cancellation, it is also necessary to determine driver properties. Among them, bandpass property and reaction

delay are widely known. A study on the bandpass property can help us validate the quality of data collected and understand the relations between perceptual variables and acceleration outputs. In *paper II* [50], the de-noised data process was further analyzed for those driver properties. The main idea is to investigate the car-following time series in the frequency domain by applying the theoretical Fourier transform (FT) [58]

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)e^{-j\omega x} dx \quad (3.6)$$

which becomes discrete Fourier transform (DFT) in practice, i.e.,

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}. \quad (3.7)$$

where k is the discrete frequency component and $j = \sqrt{-1}$. This discrete implementation leads to estimation bias in the frequency domain due to the finite length of data $x(n)$. However, this has not obstructed our investigation through spectrum analysis of car-following data, in which the FTs of auto-covariance of time series and cross-covariance between input and output time series are studied. Power spectra of perceptual input and reactive output data are found to be narrow-band signals; the transfer function estimate gives various bandpass filters, though similar shape, for different drivers. Further analysis shows that humans only react to certain bandwidth stimulus of input signal and an inherent difference exists between human drivers.

In addition to the bandpass property, delay can be clearly observed in the phase plots of estimated transfer functions of all drivers. Based on the fixed delay assumption, two methods are adopted to identify driver reaction time from real data in the paper. The reaction delay estimates are further used in the model calibration study. In fact, delay times computed from large data sample can be used to estimate reaction time distribution, which is widely used in any microsimulation tool. In our initial estimation of a population of 12 drivers, the estimated reaction time is between 0.52 and 1.24 secs. The limited number of subjects observed in our initial experiment run prevents us from giving an estimation of the distribution. Although spectrum based delay estimation is theoretically sound, the estimation suffers a critical flaw in spectrum estimation bias, which undermines the reliability of obtained results. Thus repeated spectrum computation has to be performed using different parameters (window type, frequency number of fast FT etc. [21]) in order to find a reasonable delay. Based on the estimated delay time, a generalized GM model is estimated

Action	Parameters		
	α	β	γ
acceleration	0.0025	1.49	0.25
deceleration	-5.20	1.49	2.71

Table 3.1: The estimated parameters of the generalized GM-type model in the stable following regime

using data in stable following regime in *Paper II*. The estimated parameters are presented in the table 3.1.

3.3 Driving regime analysis

In real world, it is not uncommon to observe various characteristics for a physical phenomenon in different conditions or environments. Learning from the nature, we humans also have adaptive reactions to the physical world in our cognition and decision procedure i.e. we behave differently in various physical conditions and environmental contexts. This “natural” talent of human being has been applied as a strategy in different scientific developments, namely “divide and conquer” (DAC).

In a driving task, regime adaptivity, which indicates that drivers show different operational characteristics in different regimes, is a remarkable feature of human behavior. This feature has been recognized and utilized in modeling of driver behavior, especially car-following, in literature summarized in chapter one. For example, in both psycho-physical models [43] and MITSIMLab models [2], different behavioral regimes have been defined and model parameters in those regimes are estimated using empirical data. Different from their approach, we try to identify the regime properties from collected real data by exploring and classifying the patterns. At the same time, video recordings in Volvo ERS help us understand the general behavioral process in the images. Five regimes are considered, including following, acceleration, braking, approaching and opening. Special regime like cut-in will be merged into following.

Clustering is a computing method to group data into regimes of similar properties. Fuzzy C-means algorithm, introduced in chapter 2, is an efficient way to cluster a multivariate dataset into several categories by obliging a general distance measure. However, it suffers from an inherent assumption of independence between different data points. In our application, time series of car-following data need to be classified and the neighboring data

**Initialize a random membership matrix U with Eq. (3.10) fulfilled;
While stopping condition is false do**

1. calculate C cluster centers \mathbf{c}_i for $i = 1, \dots, C$ using

$$\mathbf{c}_i = \frac{\sum_{t=1}^N u_{it}^m \mathbf{x}_t}{\sum_{t=1}^N u_{it}^m};$$

2. update the membership matrix U using

$$u_{it} = \frac{(d_{it}^2 + 2\alpha \sum_{\delta t=-l, \delta t \neq 0}^l \Phi_{\delta t} R_{it, \delta t})^{-1/(m-1)}}{\sum_{c=1}^C (d_{ct}^2 + 2\alpha \sum_{\delta t=-l, \delta t \neq 0}^l \Phi_{\delta t} R_{ct, \delta t})^{-1/(m-1)}};$$

3. compute the objective function.

end do

Table 3.2: The consolidated Fuzzy C-means clustering algorithm.

points have a natural correlation. This property of time series data forces us to develop a more robust clustering method in *paper I* [49] with consideration of dependence between neighboring data points. This leads to an idea to reformulate a general clustering objective with an additional regulation term, and a new optimization function is proposed as follows:

$$\mathbf{J}_n = \sum_{i=1}^C \sum_{t=1}^N u_{it}^m d_{it}^2 + \alpha \sum_{i=1}^C \sum_{t=1}^N u_{it}^m \sum_{\delta t=-l, \delta t \neq 0}^l \Phi_{\delta t} R_{it, \delta t} \quad (3.8)$$

and

$$R_{it, \delta t} = \sum_{p=1, p \neq i}^C u_{p, t+\delta t}^m \quad (3.9)$$

where u_{it} is the membership of sample \mathbf{x}_t belonging to the i -th cluster whose center is \mathbf{c}_i . $R_{it, \delta t}$ is the summation of the m exponent of the membership degrees that the neighbor point at $t + \delta t$ belongs to all clusters except cluster i ; α represents the impact factor of the regularization term and the function $\Phi_{\delta t}$ describes the contribution weight from the data point's neighborhood, which can be defined based on e.g. local correlations information of

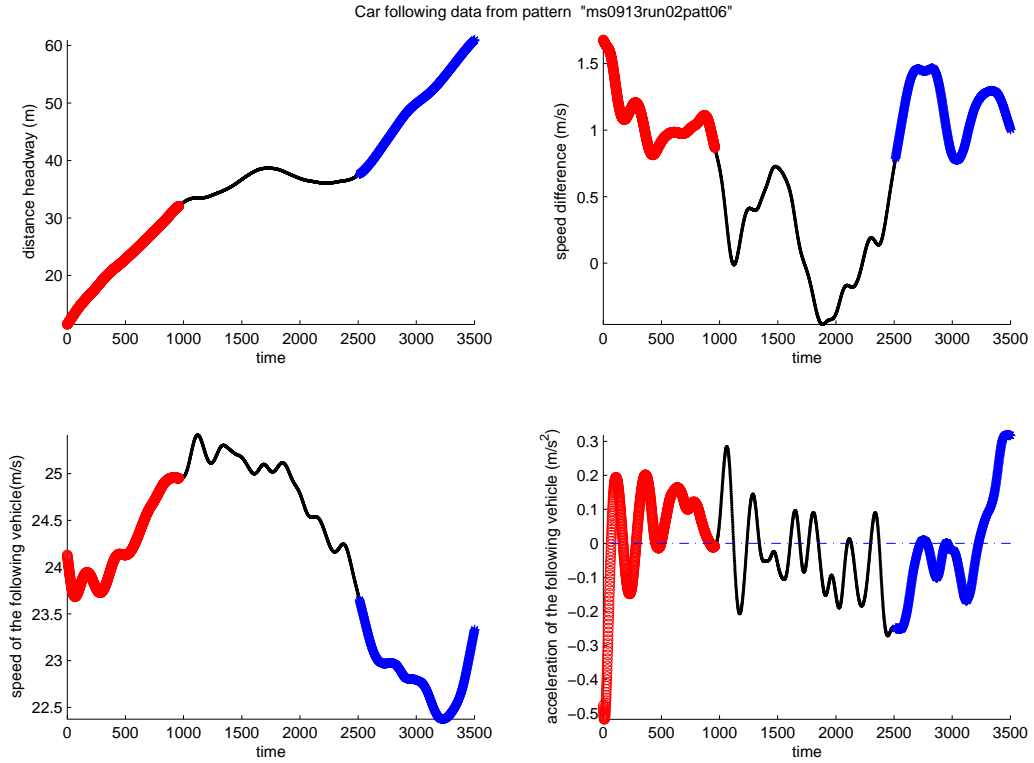


Figure 3.3: An example of classification of car-following data into several regimes based on fuzzy clustering methods.

the time series. A constraint on the optimization objective is necessary to be fulfilled

$$\sum_{i=1}^C u_{it} = 1 \quad \forall t \in [1, N]. \quad (3.10)$$

By introducing the additional term in equation (3.8), we can restrict the case that a data vector belongs to a certain class with a high degree and at the same time its neighbors have a high degree of association with other classes (vice versa). So the final membership function will be somehow regularized. Based on the new objective function, a clustering algorithm originated from the fuzzy C-means algorithm is derived in *paper I*. Table 3.2 illustrates this robust version of the fuzzy C-means algorithm. The algorithm has been applied successfully in classification of the car-following time series data unable to be classified by standard

fuzzy C-means algorithm. However, this method has a limitation on its suboptimal fact i.e. the algorithm can not ensure finding a global optimum solution to the equation (3.8). Meanwhile, the determination of the penalty weight α is empirical, not necessary optimal in respect of the general objective function, and extensive trial-and-error procedures have to be conducted to separate the multivariate time series data into reasonable regimes. In addition, the cluster number needs to be determined in advance. We decide the cluster number based on analysis of video record as well as results from experiments. In figure 3.3, we plot an example that a car-following process is successfully clustered into three regimes.

After clustering more than thirty time series into five main behavioral regimes, it is necessary to study the relation between perceptual variables and acceleration output in each regime. The main objective is to evaluate whether regime classification can help model construction and estimate a proper model with respective to each regime. In *paper III* [51], we started with regression and correlation analysis in each regime. Analysis of data of each regime gives us better understanding of how to model the behavior e.g. acceleration can be modeled in a linear form with high adequacy. Analysis result confirms that regime classification can cluster data into groups with high similarity and then reduce the complexity involved in the modeling phase. Correlation analysis between acceleration of leading and following vehicles suggests that acceleration of the leading vehicle shall be adopted as a predictor variable of a car-following model in certain regimes.

There are two key questions for building a computational model based on multiple behavior regimes: how to build a general computational model to incorporate the sub-models in each regime? how to estimate the model based on real data? The first question can be interpreted as a task to design a model generalization framework. The second question involves development of a global model estimation method for realistic description of driver-following dynamics. In fact, a good model estimation scheme is not only a challenge for regime based model estimation but also for any other types of models. In the next section, we will focus on the study of model estimation methods for general car-following models.

3.4 Model estimation scheme

Model estimation is an essential procedure in general model construction. Model parameters are required to be identified from real data so that the model can be used in real-time

prediction or simulation. In transport studies, this procedure is often called model calibration. Traditionally, car-following behavior models were frequently estimated based on the least square approach with an objective to minimize the deviation between model output and real acceleration data, i.e.

$$\min_{\theta} \mathbb{E}[e_n^i(t)^2] = \min_{\theta} \mathbb{E}[(\tilde{a}_n^i(t) - f(\hat{\mathbf{x}}_n^i(t), \hat{\mathbf{x}}_{n-1}^i(t), \Theta))^2] \quad (3.11)$$

where \tilde{a}_n^i is the real acceleration of the following vehicle n for the data sequence i and $\hat{\mathbf{x}}_n(t)$ is the physical state of vehicle n at time t ; Θ is the parameter set and $f(\cdot)$ is the car-following model. In general, this is a nonlinear optimization problem, and numerical optimization algorithms including gradient or conjugate gradient based methods and Gauss-Newton approaches are common tools to handle this problem. In fact, the numerical optimization methods involved in our model calibration study have been reviewed in the second chapter. A common critical flaw of all these methods is that they are suboptimal local search algorithms and run a risk to get trapped into local minimum solution. An improvement to these methods is to sample several random initial points, then run the numerical algorithm from those initial points in parallel and finally choose the minimum solution. The introduction of a stochastic scheme to those methods consolidates their ability to find the global optimum solution but boosts up the computational cost. In *paper II* [50], both gradient search and Gauss-Newton methods are applied to estimate an extended GM model with fixed reaction delay obtained in spectrum analysis of car-following data. The calibrated GM-type model was evaluated by closed-loop simulations on both state replication and cross validation tests. The tests showed that it could replicate, though with certain bias, speed and trajectory profiles of followers in the stable-following regime where inter-vehicle space fluctuates within a small range. A major question to calibration objective of equation (3.11) is that the model inputs are empirical data, and this makes it inefficient in capturing the car-following dynamics, in particular when large oscillations exist in the process. In addition, trajectory and speed are normally the variables directly measured and the optimization functions based on them are smoother than the acceleration based objective function.

The insufficiency of the static calibration method promotes us to explore a dynamic calibration approach using a general optimization function on the basis of whole vehicle states in *paper IV* [53]. According to the idea from adaptive filtering theory [22], a nonlinear state-space model is formulated with an extended state vector including model parameters

State update (prediction):

$$\begin{aligned}\hat{\mathbf{x}}(t|t-1) &= \mathbf{f}(\hat{\mathbf{x}}(t-1|t-1), \mathbf{u}(t-1), t-1), \\ \mathbf{F}(t-1) &= \nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}, \mathbf{u}(t-1), t-1)|_{\mathbf{x}=\hat{\mathbf{x}}(t-1|t-1)}, \\ \mathbf{P}(t|t-1) &= \mathbf{F}(t-1)\mathbf{P}(t-1|t-1)\mathbf{F}(t-1)^T + \mathbf{R}_e\end{aligned}$$

Measurement update (correction):

$$\begin{aligned}\Gamma(t) &= \mathbf{P}(t|t-1)\mathbf{H}(t)^T[\mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}(t)^T + \mathbf{R}_m]^{-1}, \\ \mathbf{H}(t) &= \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x}, t)|_{\mathbf{x}=\hat{\mathbf{x}}(t|t-1)} \\ \hat{\mathbf{x}}(t|t) &= \hat{\mathbf{x}}(t|t-1) + \Gamma(t)(\mathbf{y}(t) - \mathbf{h}(\hat{\mathbf{x}}(t|t-1), t)) \\ \mathbf{P}(t|t) &= (\mathbf{I} - \Gamma(t)\mathbf{H}(t))\mathbf{P}(t|t-1)\end{aligned}$$

Table 3.3: An illustration of discrete extended Kalman Filter algorithm

and vehicle states

$$\begin{aligned}\mathbf{X}_n(t+1) &= \mathbf{F}(\mathbf{X}_n(t), \mathbf{x}_{n-1}(t), e(t)) \\ \mathbf{y}_n(t) &= \mathbf{H} \cdot \mathbf{X}_n(t) + \mathbf{m}(t)\end{aligned}\tag{3.12}$$

where $\mathbf{X}_n(t) = [s_n(t) \ v_n(t) \ a_n(t) \ \Theta(t)]^T$ is an extended state vector of the following vehicle and $\mathbf{x}_{n-1}(t)$ is the physical state vector of the leading vehicle, which can be treated as a control input vector; $\mathbf{y}_n(t) = [\hat{s}_n(t) \ \hat{v}_n(t) \ \hat{a}_n(t)]^T$ is the measurement vector (if all state variables are used in the car-following model calibration problem); $\mathbf{m}(t) = [m_1(t) \ m_2(t) \ m_3(t)]^T$ is the measurement noise vector. A new objective function has been formulated as follows

$$\min \mathbb{E}[(\mathbf{y}_n(t) - \mathbf{H} \cdot \mathbf{X}_n(t))^T \mathbf{W}(\mathbf{y}_n(t) - \mathbf{H} \cdot \mathbf{X}_n(t))]\tag{3.13}$$

where $\mathbf{X}_n(t)$ and $\mathbf{y}(t)$ are the physical state and measured state of the following vehicle n at time t and $\mathbf{H} = [\mathbf{I}_3 \ \mathbf{0}]$ and \mathbf{I}_3 is a 3-order unit matrix. To optimize the objective function using real data, extended Kalman-Filter (EKF) algorithm (demonstrated in table 3.3 based on the notation of the standard KF algorithm in chapter 2) has been iteratively used to solve the dual estimation problem. In *paper IV*, we have revealed the implicit connection between recursive least square (RLS) method and Kalman filter in order to build a theoretical foundation to fulfill the new objective function of equation (3.13) using the Kalman filter algorithm. The IEKF method has been implemented for the estimation of both linear Helly's

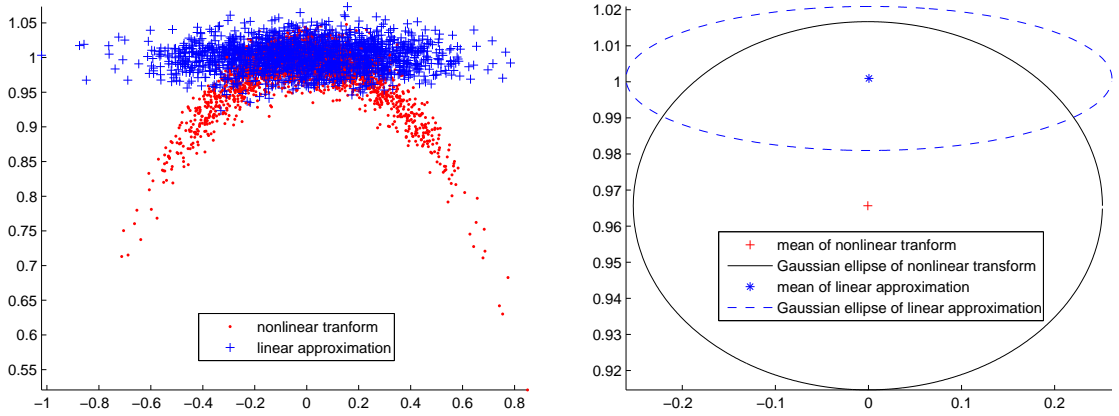


Figure 3.4: A comparison of Monte-Carlo simulation (left) of nonlinear transform and linear approximation of Polar to Cartesian coordinate and their first and second order statistic properties or Gaussian ellipse (right).

model and generalized GM model in our numerical experiments. The results show that models estimated using the dynamic calibration approach have more robust performance in describing car-following dynamics than the static calibration method.

One limitation of the IEKF based approach is the requirement on existence of derivatives or Jacobin matrix, but some car-following models, e.g. rule based system and neural fuzzy systems, do not have a direct derivative form. Another limitation of model estimation approach based on iterative EKF method is the possible divergence. Factors leading to divergence are not easy to discern immediately. In the standard Kalman filter, rounding errors may result in filter divergence. For EKF, the nonlinear system is approximated by its first order Taylor expansion and the approximation bias can result in worse performance of the filtering algorithm. Of course, whether a model form is proper also affects the performance of the estimation scheme. To illustrate the linear approximation bias, we can be simply look at a typical example in radar tracking in which Polar coordinates needs to be transformed into Cartesian coordinates by the following equations:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta \\ r \sin \theta \end{pmatrix} \text{ with } \nabla = \begin{pmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{pmatrix}. \quad (3.14)$$

Problems arise when the bearing error is much more significant than the range error, e.g. $d\theta = 15^\circ$ and $|\frac{dr}{r}| = 2\%$. In figure 3.4, we illustrate the transformation bias of linearization based on the Jacobian in equation (3.14) using Monte-Carlo simulation. This type of bias

together with rounding error in the filter may lead to divergence of the filtering algorithm in the model learning process, in particular when the car-following model is highly nonlinear.

To overcome the defects brought by Jacobin derivation, we are now improving our model estimation scheme based on Unscented Kalman-Filter (UKF) [30] [31]. Since this research is still ongoing, we only reveal the mathematical principle in this section. UKF uses deterministic σ -points to approximate the probability distribution after transformation through the nonlinear model and it replaces the extended Kalman-Filter as a more accurate and efficient method in the development of the estimation mechanism for highly nonlinear and chaotic systems, e.g. the extended GM model [1], and models without a mathematical representation of Jacobian matrix, e.g. a neural-fuzzy systems in *paper V* [46]. According to the original idea in [31], $(2n + 1)$ σ -points can be chosen through following algorithms

$$\begin{aligned}\mathcal{X}_0 &= \bar{\mathbf{x}} \\ \mathcal{X}_i &= \bar{\mathbf{x}} + (\sqrt{(n + \kappa)\mathbf{P}_{xx}})_i \\ \mathcal{X}_{i+n} &= \bar{\mathbf{x}} - (\sqrt{(n + \kappa)\mathbf{P}_{xx}})_i\end{aligned}\quad (3.15)$$

with the corresponding weights

$$\begin{aligned}\mathcal{W}_0 &= \kappa/(n + \kappa) \\ \mathcal{W}_i &= 1/2(n + \kappa) \\ \mathcal{W}_{i+n} &= 1/2(n + \kappa),\end{aligned}\quad (3.16)$$

where $\kappa \in \mathcal{R}$ and $(\sqrt{(n + \kappa)\mathbf{P}_{xx}})_i$ is the i th row or column of the matrix square root. σ -points have important properties of capturing the same mean and covariance irrespective of which square root matrix is adopted. In addition the parameter κ provides a freedom to tune the approximation of high-order moments, which can be used to reduce the overall prediction error. A detailed mathematical proof of properties of σ -point set can be referred to [31] and the early work referenced. Since σ -point set can be used to propagate statistical distributions through nonlinear systems, it is simply applied the nonlinear Kalman-Filters. A general UKF algorithm is illustrated in table 3.4 and adopts a different form from the standard Kalman filtering algorithm in table 2.1 of chapter 2. In order to sufficiently understand the general UKF algorithm for nonlinear systems, the following indirect relations, normally appeared in the mathematical proof of the standard linear Kalman filtering algorithm [20] [26], are worth mentioning:

$$\mathbf{P}_{\hat{x}\hat{y}}(t|t-1) = \mathbf{P}_{\hat{x}\hat{x}}(t|t-1)\mathbf{H}_t^T$$

-
1. Create σ -point set using algorithm of equation (3.15) and (3.16);
 2. Transform the σ -points through the plant model:

$$\mathcal{X}_i(t|t-1) = \mathbf{f}(\mathcal{X}_i(t-1|t-1), \mathbf{u}(t-1), t-1) \quad \forall i \in \{0 \leq k \leq 2n | k \in \mathbb{Z}\};$$
 3. Predict the mean by weighted summation of transformed σ -points:

$$\hat{\mathbf{x}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{X}_i(t|t-1);$$
 4. Predict the state covariance based on transformed σ -points:

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \{\mathcal{X}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)\} \{\mathcal{X}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)\}^T + \mathbf{R}_k^1;$$
 5. Transform the σ -points through the observation model:

$$\mathcal{Y}_i(t|t-1) = \mathbf{h}(\mathcal{X}_i(t-1|t-1), \mathbf{u}(t-1), t-1) \quad \forall i \in \{0 \leq k \leq 2n | k \in \mathbb{Z}\};$$
 6. Predict the observation state by weighted summation of transformed σ -points:

$$\hat{\mathbf{y}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{Y}_i(t|t-1);$$
 7. Compute the covariance of predicted observation state:

$$\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \{\mathcal{Y}_i(t|t-1) - \hat{\mathbf{y}}(t|t-1)\} \{\mathcal{Y}_i(t|t-1) - \hat{\mathbf{y}}(t|t-1)\}^T + \mathbf{R}_k^2;$$
 8. Compute the cross-covariance between process and observation states:

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \{\mathcal{X}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)\} \{\mathcal{Y}_i(t|t-1) - \hat{\mathbf{y}}(t|t-1)\}^T;$$
 9. Measurement updates in the correction phase:

$$\begin{aligned} \Gamma(t) &= \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}}(t|t-1) \mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t|t-1)^{-1} \\ \hat{\mathbf{x}}(t|t) &= \hat{\mathbf{x}}(t|t-1) + \Gamma(t) (\mathbf{y}(t) - \hat{\mathbf{y}}(t|t-1)) \\ \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t|t) &= \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t|t-1) - \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}}(t|t-1) \mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t|t-1)^{-1} \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}}(t|t-1)^T \end{aligned}$$
-

Table 3.4: Illustration of an iteration step of general UKF algorithm in prediction and correction phases using unscented transform of σ -point set.

1. Create σ -point set using algorithm of equation (3.15) and (3.16);

2. Transform the σ -points through the plant model:

$$\mathcal{X}_i(t|t-1) = \mathbf{f}(\mathcal{X}_i(t-1|t-1), \mathbf{u}(t-1), t-1) \quad \forall i \in \{0 \leq k \leq 2n|k \in \mathbb{Z}\};$$

3. Predict the mean by weighted summation of transformed σ -points:

$$\hat{\mathbf{x}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{X}_i(t|t-1);$$

4. Predict the state covariance based on transformed σ -points:

$$\mathbf{P}_{\hat{x}\hat{x}}(t|t-1) = \sum_{i=0}^{2n} \mathcal{W}_i \{\mathcal{X}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)\} \{\mathcal{X}_i(t|t-1) - \hat{\mathbf{x}}(t|t-1)\}^T + \mathbf{R}_k^1;$$

5. Measurement updates on the correction phase:

$$\begin{aligned} \Gamma_k &= \mathbf{P}_{\hat{x}\hat{x}}(k|k-1) \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_{\hat{x}\hat{x}}(k|k-1) \mathbf{H}_k^T + \mathbf{R}_k^2]^{-1} \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \Gamma_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}(k|k-1)) \\ \mathbf{P}_{\hat{x}\hat{x}}(k|k) &= (\mathbf{I} - \Gamma_k \mathbf{H}_k) \mathbf{P}_{\hat{x}\hat{x}}(k|k-1) \end{aligned}$$

Table 3.5: An illustration of an iteration step in semi-unscented Kalman filter (SUKF) algorithm using σ -point set.

$$\mathbf{P}_{\hat{y}\hat{y}}(t|t-1) = \mathbf{H}_t \mathbf{P}_{\hat{x}\hat{x}}(t|t-1) \mathbf{H}_t^T + \mathbf{R}_t^2. \quad (3.17)$$

In our car-following model identification application in equation (3.12), only the plant or state equation is nonlinear whereas the measurement equation is linear. Hence, the general UKF algorithm in table 3.4 can be simplified to a semi-unscented Kalman-Filtering (SUKF) algorithm in table 3.5.

3.5 Computational model based on a neural-fuzzy system

In section 3.3, the car-following data collected was clustered into several regimes so that proper models could be determined for each regime based on data analysis. This DAC based strategy helps us step aside from the challenging task of identifying a general mathematical model form complex enough to capture behavior in all regime conditions. However, an essential purpose in our study is to build a real-time model able to predict the car-following process e.g. in a simulation environment. Thus, regimes have to be determined in real time in order to apply a single model at each regime. In the former regime based car-following

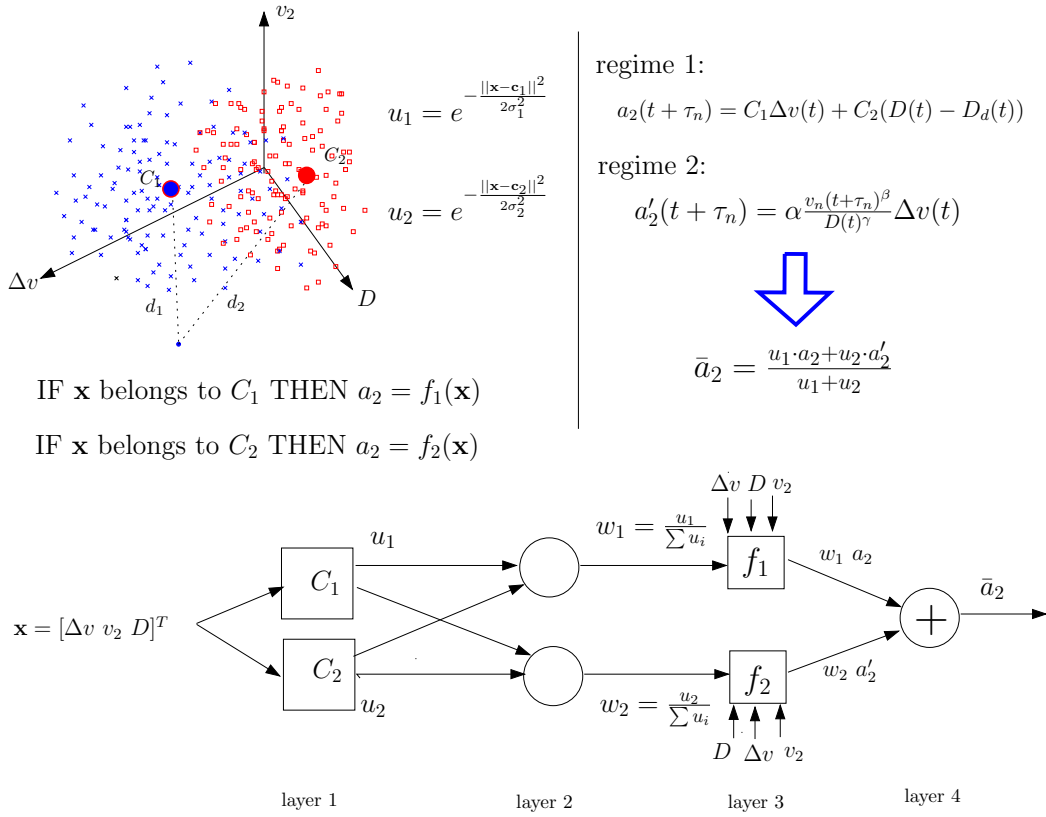


Figure 3.5: Regime based fuzzy inference system and its representation in an adaptive neural network.

models (e.g. extended GM model in MITSIMLab), the stage that a vehicle belongs to is crisply determined against a single state variable e.g. space or time headway [38]. When such kind of model is applied in simulation, frequent leaps might appear among regimes. This is contradictory to real human behavior. Meanwhile, the regime concept involves behavioral uncertainty since human drivers can transit between different regime conditions in a smooth manner according to the goal of driving comfort.

In *paper V* [46], we propose to determine the regime according to regime centers, described by a vector composed of space headway, speed difference, acceleration etc. Therefore, the following inference can be used to predict the run-time simulation output i.e.

$$\text{IF } \mathbf{x}(t) \text{ belongs to regime } C_i \text{ THEN } a_n(t + 1) = f_i(\mathbf{x}(t)) \quad (3.18)$$

where C_i can be either approaching, stable following, braking or other. f_i is the mathematical model corresponding to the regime C_i . Hence a rule based system is constructed to describe human knowledge in car-following. For run-time simulation, a crisp determination of the regime can be obtained by comparison on a similarity measure i.e.

$$j = \arg \min_i \|\mathbf{x} - \mathbf{c}_i\|_s \quad (3.19)$$

where \mathbf{x} is the current state, \mathbf{c}_i is the regime center of C_i and $\|\cdot\|_s$ is the similarity measure e.g. L_2 norm. Since whether a vehicle driver is in a certain regime is not deterministic in reality, fuzzy set theory [69] is applied to cope with regime uncertainty. Membership degrees are assigned to represent the quantitative information on the belongingness of a data point to each regime. They are basically computed by

$$u_{it} = \frac{\exp\left(-\frac{\|\mathbf{x}(t) - \mathbf{c}_i\|_s^2}{2\sigma_i^2}\right)}{\sum_{k=1}^C \exp\left(-\frac{\|\mathbf{x}(t) - \mathbf{c}_k\|_s^2}{2\sigma_k^2}\right)} \quad (3.20)$$

where \mathbf{c}_i and σ_i^2 are the center and variance (assuming spherical-cluster shape after multi-scaling in all dimensions) of a regime i respectively. Therefore, a regime-based fuzzy inference system is formulated as a computational model for driver-following behavior. To afford the model learning ability, the new inference system is reformulated as an adaptive neural network comparable to the original ANFIS model [29] for TSK fuzzy inference system (FIS). Figure 3.5 shows how the new computational model is represented as a modified adaptive neural-fuzzy network. Details are further illustrated in *paper V*.

There are two main features that make the computational model differ from the ANFIS system. First, the inference is based on regime, not on single input variable of each dimension. Second, the inference consequence involves general nonlinear model forms, not only polynomials. These two fundamental variations lead to diversifications of its model learning algorithm from the ANFIS model. In the paper, we proposed and implemented a Genetic algorithm (GA) based learning algorithm using a static calibration objective of equation (3.11). Numerical experiment e.g. in figure 3.6 shows that the model can successfully learn the acceleration as a function of perceptual input values, i.e. input and output mapping. But it has the same problem as other models in its ability to replicate car-following dynamics. This returns to our early study on model estimation scheme. Therefore, efficient learning approaches, especially using a dynamic estimation objective function, have large potential to improve the prediction ability of the model on car-following processes.

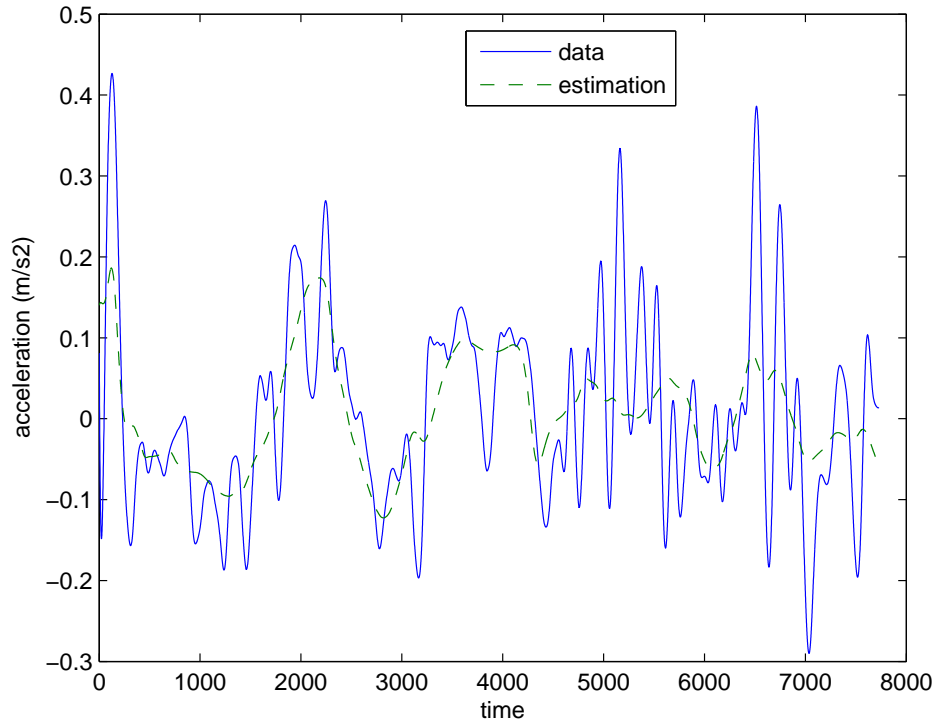


Figure 3.6: A comparison of measurement data and output of the computational model after training by the dataset composed approaching and following regimes.

3.6 Safety study on ISA

Besides modeling of driver-following behavior, a study of Intelligent Speed Adaption (ISA), a type of vehicle-based ITS systems, had been conducted in the early phase of this doctoral research. The results were reported in two publications, *paper VI* [48] and [52]. The main idea of the research project is to extend the reduction effect on free flow speed of single ISA vehicles [62] to an aggregated scale (reshaping of speed distribution) so that the relation between ISA penetration level and traffic safety can be revealed using the macroscopic traffic flow data. As it is economically unrealistic to install a large number of ISA equipments, simulation becomes the main study means. With simulation, an obvious difficulty of the research is modeling of behavioral difference between ISA and normal drivers, an area rich

of arguments according to the previous publications. Hence, the research scope in our study is limited by only considering the transformation of desired speed distributions between normal and ISA vehicle in the simulation environment. TPMA [39] is adopted as the traffic simulation tool due to the availability of source code; and in particular, it has been reported being calibrated using data collected on Swedish roads. ISA vehicle is defined as one type of vehicle in the simulator and a speed adaption object was further developed to be able to trigger reshaping of desired speed distribution according to the random seed assigned to each ISA vehicle when it was generated, that is,

$$R_s = \frac{v - v_m}{\sigma_m} = \frac{v' - v'_m}{\sigma'_m} \quad (3.21)$$

where R_s is the seed, v and v' are the desired speed assigned at the vehicle generation and the new desired speed respectively, v_m and σ_m are the mean and standard deviation before a distribution reshaping whereas v'_m and σ'_m are the mean and standard deviation after the transformation. By adjusting the percentage of ISA type vehicles, different traffic compositions can be simulated based on the Monte-Carlo method. Two types of roads were considered for simulation study: single-lane road (no overtaking) and double-lane road (overtaking). The traffic flow characteristics, e.g. spot speed, time headway etc, were measured 1 km to 2 km downstream from the speed adaption sign. Consequently, simulation can generate traffic flow data with different flow levels, traffic compositions (mainly ISA penetration), detecting distances at the downstream, initial speed distributions and road types. All simulated traffic flow data are then analyzed for safety evaluation from a microscopic perspective.

From the early study on individual ISA vehicles [27], significant reductions on both free flow mean speed and variance were reported for individual ISA vehicles. Reshaping effect on aggregated free flow speed distribution was also demonstrated in a field study [45] on free flow speed of ISA drivers in urban areas of Lund, a Swedish city. The results on free flow speed distributions were adopted in our simulation study. Gap distributions under different ISA and normal vehicle compositions are an important index for safety research since all gaps less than 3 second are considered as risky platoon mode, a safety score [44] in traffic engineering. Simulations on the single-lane road show that the risky platoon percentage is, although trivially, reduced with the increasing ISA penetrations in many cases [52]. One possible explanation is that the impact on traffic flow from reshaping of desired speed distribution of ISA vehicles overwhelms that from covariance between speed distributions

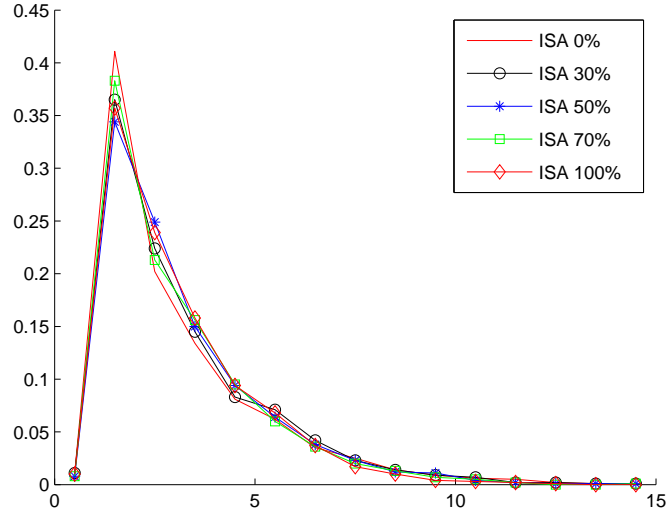


Figure 3.7: Gap distribution in different ISA penetration with speed limit 70 km/h and flow level 750 vehicles/h.

of two types of vehicles. However, the change on the distribution is not obvious and no significant conclusion can be drawn based on the numerical results. On the double-lane road, simulation result also can not lead to any safety conclusion in terms of gap distribution since the ISA effect is mitigated by the overtaking possibility.

In general, reduction of speed variance may lead to direct explanation on safety improvement. In particular, speed management potentially affects the pedestrian safety. Thus, an investigation (*paper VI* [48]) on the pedestrian safety in terms of probability of collision and death was conducted through recursive Monte-Carlo experiments based on speed outputs from traffic simulator and a vehicle-pedestrian collision model in which an unwary pedestrian darts out without care of possible collision with an oncoming vehicle. A significant reduction on pedestrian risk has been conclusively obtained and details are explained in the *paper VI*. To reveal the safety effect of ISA in depth, detailed microscopic modeling of ISA drivers seems inescapable because safety simulation needs more accurate and explicit description of behavior adaption to the new system, e.g. longitudinal following. Thus, one conclusion on safety study is that current microscopic simulation tools have limitations for direct evaluation of safety effects of all ITS systems.

3.7 Synthesis and future perspectives

Computational modeling of driver behavior from real data has met its research needs in recent transport development, in particular for accurate representation of normal driver behavior in microscopic simulation and driving simulator. A realistic driver behavior model can improve the simulation fidelity as well as provide abundant clues for evaluation and improvement of the state-of-the-art for driving assistant systems.

In this thesis, a comprehensive modeling methodology has been established on car-following behavior, from original data acquisition on the Swedish motorways to behavioral modeling by exploration of real data. Apart from data collection experiment design, our particular focus was on data processing methods, driver property estimation and development of a computational model driven by microscopic car-following data.

Different from traditional model development approach, the methodology developed begins with the information estimation and exploration of real data. Information estimation has not been properly emphasized in previous driver behavior study based on real data collection. In this study, it plays a key role since all further studies assume highly accurate and consistent perceptual and decision information. Apart from the estimation of accurate car-following data, we also explore the dataset in the frequency domain. In particular, delay estimation on the basis of spectrum analysis is proved to work as a quantitative approach, alternative to the traditional graphical method. The utility of spectrum analysis supports it as a useful tool for traffic psychologists and safety researchers to estimate driver property from observed data.

Regime adaptivity has been identified as a common characteristic of human drivers in the domain of traffic analysis. Given that it is unrealistic to formulate a general mathematical model for car-following, regime decomposition has been introduced in the modeling task in order to handle different regime conditions. Among the multiple definitions of regime concepts in the car-following process, psycho-physical based definition is the most widely accepted one. Following the regime concepts, clustering analysis has been applied in real data to identify regime boundaries. Instead of concerning with only a single state variable, the clustering approach explores car-following data in a multi-dimensional space. A consolidated fuzzy C-means algorithm is developed and applied for the multivariate time series whenever the classical fuzzy C-means algorithm does not bring understandable results.

Data classified into regimes are studied using regression and correlation analysis. According to the model adequacy result, braking and acceleration regimes can be modeled based on simple linear regression approach. The correlation analysis supports that acceleration information of the leading vehicle has its impact on the follower acceleration profile. In general, data analysis in regimes provides much information and could be a very useful approach for the design of driving assistant systems.

An ultimate goal of this research is to build a computational model from the collected data in order to apply at run-time in simulation. After regime decomposition, traditional mathematical models can be adopted in each regime, giving a general model transparency in those regimes. A fuzzy computing model on the basis of the regime inferences is developed to integrate the single regime sub-models in a seamless and unobscured manner. Moreover, the fuzzy system is represented as an adaptive neural network so that the model can be trained with empirical data under the framework. A GA-based estimation method has been proposed as the learning algorithm for this neural-fuzzy system.

In analogy to parameter identification in a function approximation task, traditional calibration methods for car-following models suffer from the defect of static estimation. Inspired by the adaptive filtering theory, a state space system is formulated with concerns on dual estimation of parameters and system states. Different Kalman filters have been applied to solve the problem, in particular when the basic car-following model is nonlinear. This method bring us promising result in simulating car-following dynamics, not just fitting the function output on acceleration.

In general, driver modeling covers many research subjects and is a multidisciplinary research task. This thesis focuses on the methodological development from data collection to data analysis, and to model development. Computational aspects of this procedure are emphasized but their application has not been deeply explored from the views of transport science. Nevertheless, the methods developed have a large potential to be used in transport analysis if corresponding datasets are available, e.g.

- The state-space approach and Kalman smoothing algorithm could be applied for car-following data preprocessing using other data-collection means, e.g. GPS data and image data from remote sensing;
- Spectrum analysis methods could be used for estimation of driver reaction time distribution given enough empirical datasets;

- Clustering analysis could be used for identification of psycho-physical thresholds;
- The dynamic model estimation based on Kalman-filtering could be applied as a standard scheme on car-following model calibration for traffic simulation.

In fact, to advocate and extend the proposed methods, further application and validation are quite necessary.

An innovative computational model on car-following based on regime inferences has been proposed in this work. Initial numerical test based on two regimes data supports the applicability of the model but further evaluation using data from all regimes are indispensable before its real implementation in simulation. In addition, model estimation scheme based on the dynamic approach has not been fulfilled, and model forms adopted in each regime should be further justified by numerical experiment in order to capture real dynamics. At last, to evaluate the computational model, it should be implemented in a simulation environment (traffic simulation model or driving simulator). Besides the future work in modeling car-following behavior, the methodological approach to create simulation model from real data has promising perspectives for other behavior models such as lane changing.

Bibliography

- [1] P.S. Addison and D.J. Low. A novel nonlinear car-following model. *Chaos*, 8(4):791–799, December 1998.
- [2] K.I. Ahmed. *Modeling Drivers' Acceleration and Lane Changing Behavior*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [3] G.H. Ball and D.J. Hall. A clustering technique for summarizing multivariate data. *Behavioral Science*, **12**:153–155, 1967.
- [4] J. Bengtsson. *Adaptive Cruise Control and Driver Modeling*. PhD thesis, Dept. of Automatic Control, Lund Institute of Technology, Sweden, December 2001.
- [5] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York, 1981.
- [6] T. Bleile. A new microscopic model for car following behavior in urban traffic. In *Proceedings of the 4th World Congress on Intelligent Transportation Systems*, Berlin, 1997.
- [7] M. Brackstone, B. Sultan, and M. McDonald. Motorway driver behavior: studies on car following. *Transportation Research Part F*, 5:31–46, 2002.
- [8] W. Burghout. *Hybrid microscopic-mesoscopic traffic simulation*. PhD thesis, Royal Institute of Technology, Sweden, 2004.
- [9] P. Chakroborty and S. Kikuchi. Evaluation of General Motors based car following models and a proposed fuzzy inference model. *Transportation Research Part C*, 7:209–235, 1999.

- [10] R.E. Chandler, R. Herman, and E.W. Montroll. Traffic dynamics: studies in car following. *Operational Research*, **6**:165–184, 1958.
- [11] O. Cordon, F. Herrera, F. Gomide, F. Hoffmann, and L. Magdalena. Ten years of genetic fuzzy systems: Current framework and new trends. *Fuzzy Sets and Systems*, 141:5–31, 2004.
- [12] G. Dahlquist and Å. Björn. *Numerical Methods for Scientific Computing (draft)*. SIAM, Philadelphia, 2001.
- [13] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, Hong Kong, China, 2001.
- [14] E. Elbetagi, T. Hegazy, and D. Grierson. Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics*, 19:43–53, 2005.
- [15] L. Fausett. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Prentice Hall, New Jersey, 1994.
- [16] J.F. Gabard, J.J. Henry, J. Tuffal, and Y. David. Traffic responsive or adaptive fixed time policies: critical analysis with SITRA-B. In *Proceedings of the International Conference on Road Traffic Signaling*, pages 89–92, Institute of Electrical Engineering (IEE) press, 1982.
- [17] D.C. Gazis, R. Herman, and R.W. Rothery. Nonlinear follow-the-leader models of traffic flow. *Operational Research*, **9**, 1961.
- [18] P.G. Gipps. A behavioural car following model for computer simulation. *Transportation Research B*, 15:105–111, 1981.
- [19] F. Goldberg. *Genetic algorithm in search, optimization and machine learning*. Addison-Wesley Publishing, Reading, MA, 1989.
- [20] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice Hall, New Jersey, 1993.
- [21] M.H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley Sons, Toronto CA, 1996.

- [22] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.
- [23] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, Ontario CA, 1999.
- [24] W. Helly. Simulation of bottlenecks in single lane traffic flow. In *Proceedings of the Symposium on Theory of Traffic Flow*, pages 207–238, Research laboratories, General Motors, New York: Elsevier, 1959.
- [25] R. Herman, E.W. Montroll, R.B. Potts, and R.W. Rothery. Traffic dynamics: analysis of stability in car following. *Operational Research*, **7**:86–106, 1959.
- [26] H. Hjalmarsson and B. Ottersten. *Lecture Notes in Adaptive Signal Processing*. Royal Institute of Technology, Stockholm, 2002.
- [27] M. Hjlmdahl and A. Várhelyi. Speed regulation by in-car active accelerator pedal: effects on driver behaviour. *Transportation Research Part F*, **7**:77–94, 2004.
- [28] C.-T. Sun J.-S. Jang and E. Mizutani. *Neuro-Fuzzy And Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, New Jersey, 1997.
- [29] J.-S. Jang. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. on Systems, Man and Cybernetics*, pages 665–685, 1993.
- [30] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman-Filter to nonlinear systems. In *Proceedings of SPIE AeroSense Symposium*, FL, April 21-24 1997.
- [31] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of IEEE*, **92**(3):401–422, 2004.
- [32] R.E. Kalman. A new approach to linear filtering and prediction problems. *ASME Transaction: Journal of Basic Engineering*, **D**:35–45, 1960.
- [33] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, New Jersey, 1993.
- [34] C. Kikuchi and P. Chakroborty. Car following model based on a fuzzy inference sytem. *Transportation Research Record*, **1365**:82–91, 1992.

- [35] T. Kim, D.J. Lovell, and Y. Park. Limitation of previous models on car-following behaviors and research needs. In *Proceedings of the 82th TRB annual meeting*, Washington D.C., 2003.
- [36] E. Kometani and T. Sasaki. Dynamic behavior of traffic with a nonlinear space-speed relation. In *Proceedings of the Symposium on Theory of Traffic Flow*, pages 105–119, Research laboratories, General Motors, New York: Elsevier, 1959.
- [37] M. Koshi, M. Kuwarara, and H. Akahane. Capacity of sags and tunnels on japanese motorways. *ITE Journal*, 62:17–22, 1992.
- [38] I. Kosonen. *HUTSIM - Urban Traffic Simulation and Control Model: Principles and Applications*. PhD thesis, Helsinki University of Technology, 1999.
- [39] I. Kosonen, A. Gutowski, F. Davidsson, T. Sauerwein, P. Blad, and D. Aalto. Traffic performance on major arterials. Final report part (b-f), Royal Institute of Technology, Sweden, Center for Traffic Simulation Research, 2001.
- [40] M. Kubat. Decision trees can initialize RADIAL-BASIS function networks. *IEEE Transactions on Neural Networks*, 9(5):813–821, September 1998.
- [41] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis, 3rd edition*. McGraw-Hill Higher Education, 2000.
- [42] W. Leutzbach. *Introduction to the Theory of Traffic Flow*. Springer Verlag, Berlin, 1968.
- [43] W. Leutzbach and R. Wiedemann. Development and applications of traffic simulation models at the karlsruhe institut fur verkehrwesen. *Traffic Engineering and Control*, pages 270–278, May 1986.
- [44] G. Lind. *Strategic Assessment of Intelligent Transport Systems: A User Oriented Review of Models and Methods*. PhD thesis, Royal Institute of Technology, Sweden, 1997.
- [45] A. Várhelyi M. Hjälm Dahl and S. Almqvist. Effekt av aktiv gaspedal på körmönster (in swedish), effect of active accelerator pedal on the driving pattern. Technical report, Lund University of Technology, Sweden, Division of Traffic Engineering, Department of Technology and Social Science, 2002.

- [46] X. Ma. A computational model of driver-following behavior using a neural-fuzzy system. *Neurocomputing*, 2006. submitted for publication.
- [47] X. Ma and I. Andréasson. Dynamic car following data collection and noise cancellation based on the Kalman smoothing. In *Proceedings of IEEE International Conference on Vehicular Electronics and Safety (ICVES'05)*, 2005.
- [48] X. Ma and I. Andréasson. Predicting the effects of ISA penetration grades on pedestrian safety by simulation. *Accident Analysis and Prevention*, 37:1162–1169, 2005.
- [49] X. Ma and I. Andréasson. Behavior measurement, analysis and regime classification in car-following. *IEEE Transactions On Intelligent Transportation Systems*, 2006. in press.
- [50] X. Ma and I. Andréasson. Driver reaction time estimation from the real car following data and its application in GM-type model evaluation. *Transportation Research Record - Journal of Transportation Research Board*, 2006. sponsored by traffic flow theory committee, in press.
- [51] X. Ma and I. Andréasson. Statistical analysis of driver behavioral data in different regimes of the car-following stage. *Transportation Research Record - Journal of Transportation Research Board*, 2006. submitted for publication.
- [52] X. Ma, L. Engellsson, I. Andréasson, and G. Lind. Evaluation of safety effects of various ISA vehicle penetration grades by microscopic simulations. Technical report, Royal Institute of Technology, Sweden, Center for Traffic Simulation Research, February 2004.
- [53] X. Ma and M. Jansson. A general Kalman-Filtering based model estimation method for car-following dynamics in traffic simulation. *Transportation Research C: Emerging Technologies*, 2006. submitted for publication.
- [54] A.G. Parlos, S.K. Menon, and A.F. Atiya. An algorithmic approach to adaptive state filtering using recurrent neural networks. *IEEE Transactions on Neural Networks*, 12(6):1411–1432, 2001.
- [55] P. Ranjitkar and T. Nakatsuji. Car-following model: an experiment based benchmarking. *Journal of EASTS*, 6:1582–1596, 2005.

- [56] I. Rivals and L. Personnaz. A recursive algorithm based on the extended Kalman filter for the training of feed-forward neural network models. *Neurocomputing*, 20:274–294, 1998.
- [57] D. Simon. Training radial basis neural networks with the extended Kalman filter. *Neurocomputing*, 48:455–475, 2002.
- [58] G. Strang. *Introduction to Applied Mathematics*. Wellesey-Cambridge Press, Wellesley MA, 1986.
- [59] H. Subramanian. Estimation of car following models. Master’s thesis, Massachusetts Institute of Technology, 1996.
- [60] J. Sum, C. Leung, G.H. Young, and W. Kan. On the Kalman filtering method in neural network training and pruning. *IEEE Transactions on Neural Networks*, 10(1):161–166, January 1999.
- [61] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier Science, USA/Singapore/China, 2003.
- [62] A. Várhelyi and T. Mkinen. The effects of in-car speed limiters: field studies. *Transportation Research Part C*, 9:191–211, 2001.
- [63] L.X. Wang. *Adaptive fuzzy systems and control: design and stability analysis*. Prentice Hall, New Jersey, 1994.
- [64] R.E. Wilson. An analysis of Gipps’ car-following model of highway traffic. *IMA Journal on Applied Mathematics*, 2001. submitted Oct. 2000 and accepted Mar. 2001.
- [65] J.P. Wu, M. Brackstone, and M. McDonald. Fuzzy sets and systems for a mortorway microscopic simulation model. *Fuzzy sets and systems*, 116:65–76, 2000.
- [66] J. Xing. A parameter identification of a car following model. In *Proceedings of the 2nd World Congress on ATT*, Yokohama, 1995.
- [67] Y. Xu, K.W. Wong, and C.S. Leung. Generalized RLS approach to the training of neural networks. *IEEE Transactions on Neural Networks*, 17(1):19–34, January 2006.
- [68] Q. Yang. *A Simulation Laboratory for Evaluation of Dynamic Traffic Management Systems*. PhD thesis, Massachusetts Institute of Technology, 1997.

- [69] L.A. Zadeh. Fuzzy sets. *Information and Control*, **8(3)**:338–353, 1965.
- [70] B.P. Zeigler, H. Praehofer, and T.G. Kim. *Theory of Modeling and Simulation: Integrating discrete event and continuous complex dynamic systems, 2nd edition*. Academic Press, 2000.
- [71] H.J. Zimmermann. *Fuzzy Set Theory and Its Applications*. Kluwer Academic, Boston/Dordrecht/London, 1996.