Analysis, modeling and application of personalized driving behavior based on multi-mode dynamical systems

Thomas WILHELEM

Department of Mechanical Science and Engineering

Graduate School of Engineering, Nagoya University

A thesis submitted for the degree of

Doctor of Engineering

2017

Contents

List of Figures

List of Tables

Notations

Time

- t: continuous time
- k: discrete time step
- T: continuous time delay
- τ: discrete time delay

Vehicle

- N: vehicle number
- s: curvilinear position
- v: vehicle velocity
- a: vehicle acceleration
- Δx or R: Range to the leading vehicle
- Δv or RR: Range-rate to the leading vehicle
- THW: Time headway to the leading vehicle
- FV: Following vehicle
- VLV: Virtual leading vehicle
- RLV: Real leading vehicle

Energy consumption evaluation

- $\dot{vol}_F$: Fluid volume flow
- $\dot{m}_F$: Specific fluid mass flow
- ζ_F: Fluid density
- ω_{Eng}: Engine rotation frequency

- Trq_{Eng}: Engine torque at the crank shaft
- P_{Eng}: Engine output power
- FC: Fuel consumption

Multi-mode dynamical system models

- n_a: autoregressive order of ARX model
- n_b: exogenous order of ARX model
- n: model input number
- q: model output number
- x: model state
- z: observation
- m: mode index
- M: number of modes
- r: regression vector or input vector
- u: exogenous model input
- $\hat{y}$: estimated model output
- θ: ARX model parameter
- η: softmax function parameter
- Θ: ARX model parameter space
- h and H: matrices for piecewise model partition description
- X: data partition
- μ: mode index vector
- P: mode presence probability

System identification

- e: modeling error
- ϵ: parameter estimation error
- θ: identified parameters vector

- Θ: identified parameter space
- lr: resampled particle index
- ls: sampled particle index
- Lr: resampled particle number
- Ls: sampled particle number
- w: weight
- i: iteration
- Ξ: particle set
- p and g: probability distributions

Chapter 1

Introduction

1.1 Background

Since the introduction of the first mass produced vehicle, the Ford Model T in 1908, the automotive has been a vector of movement for millions of persons. Motor vehicles enable effortless and comfortable transportation to any road paved destination at affordable cost. Long distance transportation, that was luxury for centuries, became reachable by almost anybody. Up to this day, the enthusiasm for the property of personal vehicles led to a non-stopping increase of the number of vehicles on the planet (see Figure 1.1). City infrastructures, businesses, law and entire parts of the economy have been reshaped by this invention.

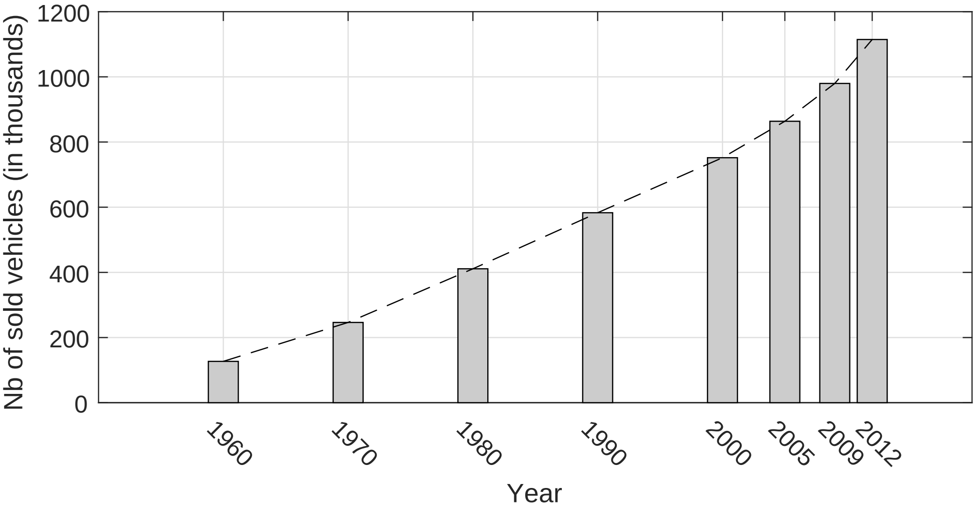


Figure 1.1: Historical trend of worldwide vehicle registrations. Data extracted from [1].

The global interest toward motor vehicles rapidly extended to a reflection on broader topics, such as city urbanization and planning, and infrastructures management. On the end of the twentieth century, a change of focus took place toward safety and pollution concerns. It has to be noted that the freedom provided to individuals became a right acquired with years, forbidding to solve pollution problems by simple removal of vehicles. In the meantime, the democratization of Internet provided alternate methods to shorten distances between individuals. Nevertheless, it is likely that the use private vehicles will still be a central element in our way exchange with the world for the coming generations.

The above mentioned topics lead the creation of a rich literature on fundamental reflections on the analysis of human perception and cognition, on the understanding and modeling of traffic flows, and on the possible improvement of the vehicles themselves. Based on these research works, application studies proposed point-of-views and technical solutions to facilitate the creation of road infrastructure, to development safer and less polluting cars, and to improve vehicle's overall design.

A way to discover new technologies, to improve vehicle usability, to better integrate vehicles to the environment and to propose solutions to reduce the carbon footprint left by cars on the planet is to better understand the human driver. By doing so, realistic traffic flow models can be generated, vehicle dynamics can be optimized to reduce energy consumption, and advanced driver assistance systems can be embedded with harmony in the driver environment. These topic are the source of inspiration of this thesis.

A topic worth mentioning due to its potential to revolutionize the way to conceive complex systems is the notion of virtual conception. Virtual conception of complex systems goes further than bringing some elements of expertise or some physical rules into an existing creation framework. Virtual conception describes a way to engineer products by virtually estimating the entire functioning of a system. To do so, all the individual models describing a system have to communicate to work as one entity. They also have to be able to evaluated all possible realistic scenarios. As such, the modeling of the systems is not only descriptive but also predictive. A way to cover all possible scenarios is to change a part of the modeling approach from a deterministic to a probabilistic approach. In the case of vehicles design, the inputs of the system are provided by the driver and the outside world scenario. While the outside world scenarios are well known and can be modeled without much difficulty, the modeling human driver modeling is uniquely challenging.

An applied example to virtual conception is the design of vehicle powertrain . Constrained by regulatory emission standards [4, 5], car manufacturers have to find clever solutions to reduce the emitted pollutants of their machines. Embedded systems are nowadays becoming complex, and powertrain technology varied. The conversion of stored energy can be done by combustion engines technologies, electric motors, fuel cells, hydraulic systems, or their combination [6–8]. To be able to handle such complex system, provide flexibility and to shorten development cycle duration, design with hardware-in-the-loop, and design performance comparison by simulation is commonly used. From this point of view, driving behavior modeling is a central topic. This domain enables to verify the good functioning of designed vehicle powertrain based on real-usage simulations. As for today, it could be used together with hardware-in-the-loop (HIL, see Figure 1.2) integrations, and it would allow increased flexibility in development and testing abilities.

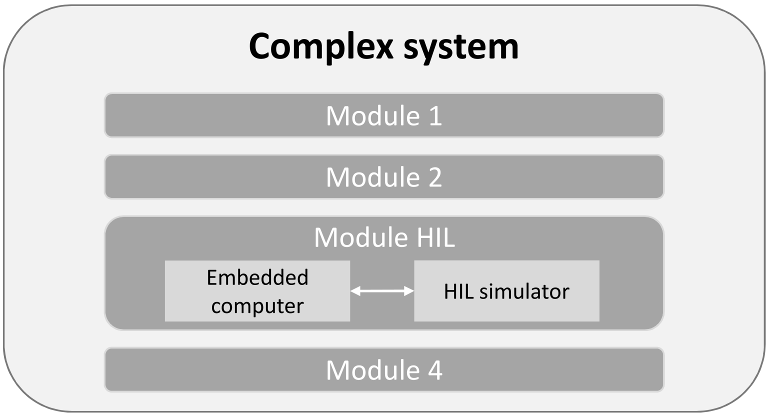


Figure 1.2: Example of system composed of four physical modules, including one replaced by a hardware-in-the-loop module.

Finally, modern vehicles integrate a growing number of advanced driver assistance systems (ADAS), such as emergency braking systems, adaptive cruise control systems (ACC) (see Figure 1.3), or lane keeping systems. Driver behavior personalization of these systems is a way to bring confidence and better acceptability by the user. These systems can also be used by the vehicles to better understand the driver and other vehicles behavior, and thus help to avoid dangerous situations by predicting their own or other vehicles trajectories.

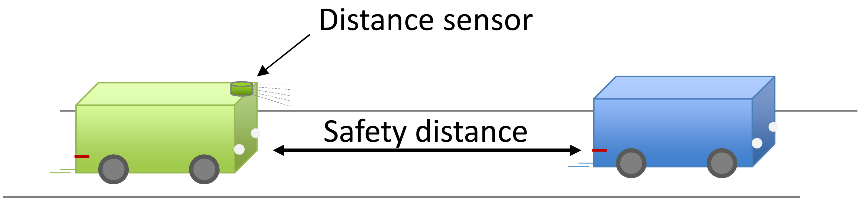


Figure 1.3: Illustration of an two vehicles following each other with an adaptive cruise control system.

Inspired by these topics, this doctoral research is focused on the understanding and the modeling of the human personalized driving behavior in car-following situation. The car-following situation has been selected for its ability to scale to the widest range of problems: it can be used for human cognition understanding, trends analysis, traffic flow modeling, advanced driving assistance systems, and vehicle powertrain design. In this study, driving behavior understanding has been done by:

- collecting driving measurement data,
- analyzing driving behavior dynamics,
- selecting and designing driver modeling frameworks,
- the implementation a novel system parameters identification method.

Driver behavior modeling has been tackled by creating a framework for identification and validation of the driver behavior simulation, using a data centric approach and the unique abilities of multi-mode dynamical system models (see Figure 1.4).

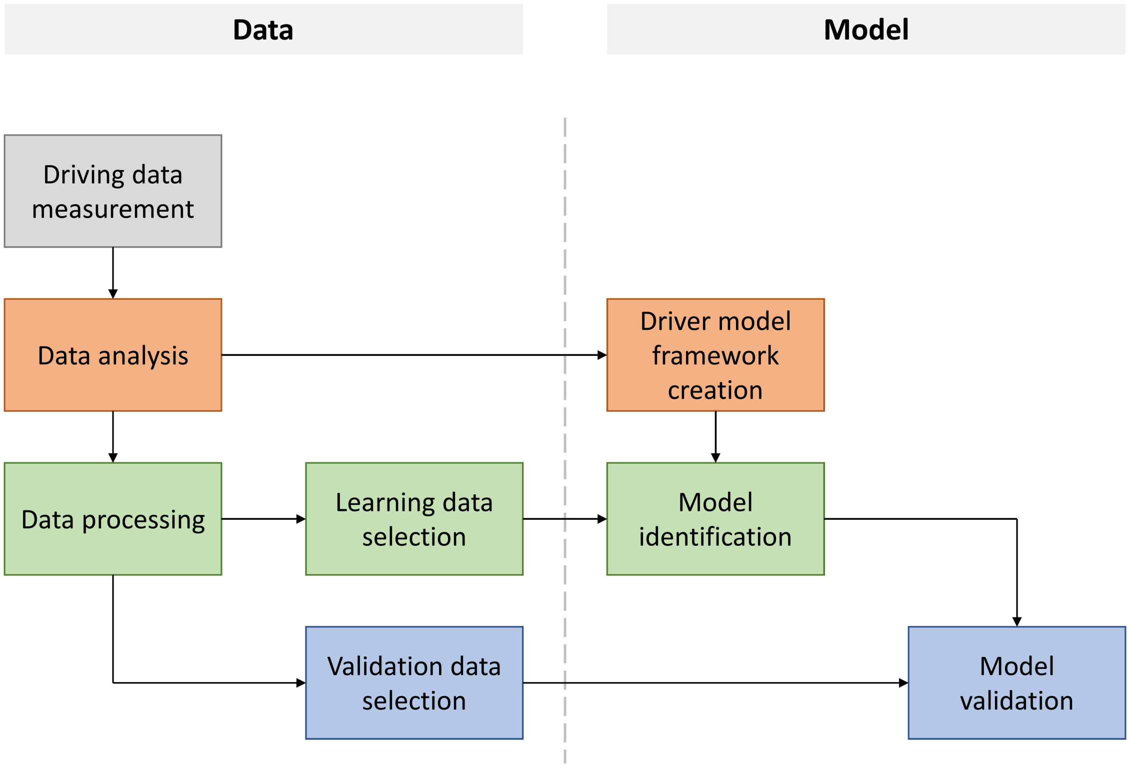


Figure 1.4: Perspective on data centric modeling.

The completeness of this approach enabled us to have a global understand of the possibility of human driver behavior analysis and modeling. As a result of this research, several application models are proposed. At first a behavior personalized cruise-control ADAS framework has been designed [9] to test the most classic traffic flow models in a unified environment. Then a driver personalized energy consumption evaluation method has been developed [10] to show the unique abilities of joint usage of system identification and multi-mode dynamic modeling. Finally, a novel parameters identification scheme has been proposed to describe and understand subtle drivers' personalized behavior [11, 12].

1.2 Previous works

Before going into more detailed explanations about the elements of this research, this section proposes a literature review of the state of the art driver models, and some details about specific features of human drivers' behavior.

1.2.1 Driving behavior modeling

Modeling of driver behavior has been approached since the 50's. At first, number of researches have been undertaken to understand the biological and psychophysical behavior of human [13–17]. These approaches, focused on the human-machine interaction, were mostly inspired from aeronautic and space developments. In the 50's and 60's, most modeling studies were focused on geometrical and control approaches [18–20]. Biological centered studies by Rashevsky concluded that clear-cut separation of the mechanical and the human entity would be difficult to understand. The car and the driver constitute a complex feedback system, such that they should not be separated in two distinct components [21, 22].

Nevertheless, most of the modern understanding and modeling methods of the driver behavior dates from the 80's. Analysis of longitudinal driver behavior has been done by two approaches: understanding of the human performance, and understanding of the driver behavior. Human perception and sensory performance has been analyzed and a few general models were proposed [2, 23–25]. These studies do analyze the sensory limitations of humans, and they discuss the decision making structure that should be used (as an example, see Figure 1.5). Regarding the longitudinal driving task, numerous studies focused on the understanding and the reproduction of specific driving tasks, such as emergency braking, following behavior and velocity in curve [26–33]. These research works provide interesting models, but they are focused on very specific purposes.

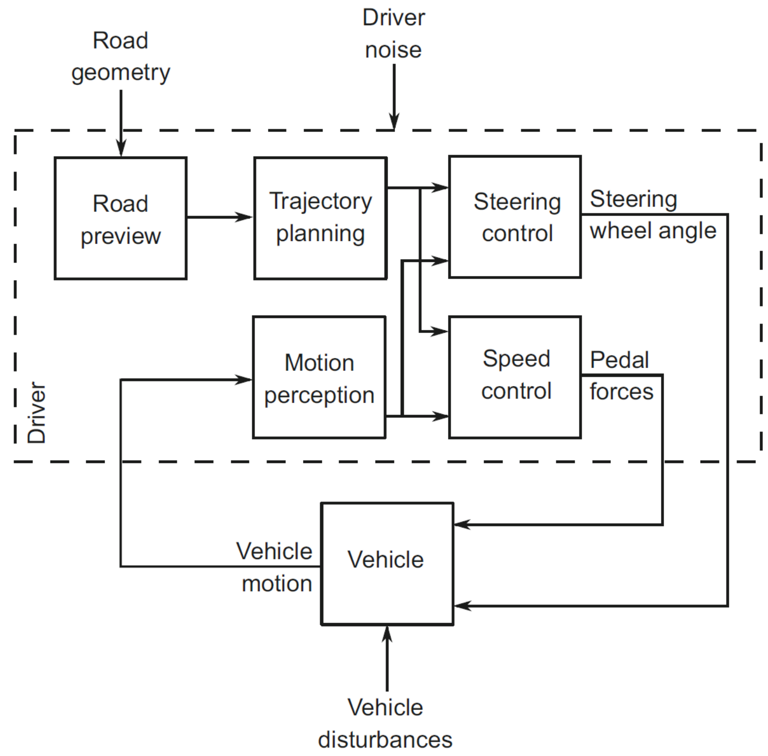


Figure 1.5: Block diagram of a cognitive model representative of the tasks carried out while driving. The driver is processing data from the upcoming road geometry and from the vehicle dynamics to proving the steering wheel angle and the pedal operation. Figure from [2].

Longitudinal motion of vehicles has also widely been investigated for traffic flow study and modeling. Traffic flow modeling is usually separated in two classes: macroscopic traffic flow modeling and microscopic traffic flow modeling [34]. Macroscopic traffic flow modeling is focused on the simulation of the behavior of complete road sections, while microscopic modeling approaches the problem by modeling each vehicle independently. The second approach the most interesting for driver behavior reproduction. The first microscopic traffic flow simulation papers appeared in the 50's, and have been followed by numerous measurement campaigns, analysis papers and model propositions [3, 35–38]. Most of the modeling propositions fit a microscopic scale vehicle dynamic model to statistically observed traffic flow dynamics. These models are usually focused on two main tasks: free driving and car-following tasks (see Figure 1.6). While the each individual vehicle of the traffic flow is modeled, validation of the model is done statistically, with specific traffic flow metrics, such as the speed vs density, and the speed vs flow graphs [34]. Some of the most popular models, due their simplicity to represent driving dynamics from a single equation, are the stimulusresponse Gazis-Herman-Rothery (GHR) model, also called GM models [39], and the linear Helly models [40]. These minimalistic approaches allow to represent first order individual driver behavior, but do not allow to reproduce detailed nonlinear and stochastic human behavior. To improve model's accuracy without overcomplicated formulations more recent researches proposed multi-modal approaches. The main contenders of such model type are the Gipps collision avoidance model [41], and the Wiedemann psychophysical (also named action-point) model [42]. More details about these models are available in the following sections. Some studies highlight the limitations of all aforementioned methods, as they usually neglect human task scheduling and attention management phenomenons [43–45]. These remarks are interesting and should be considered for the future development of human driver modeling as they do not seem to be much integrated in research studies to this day.

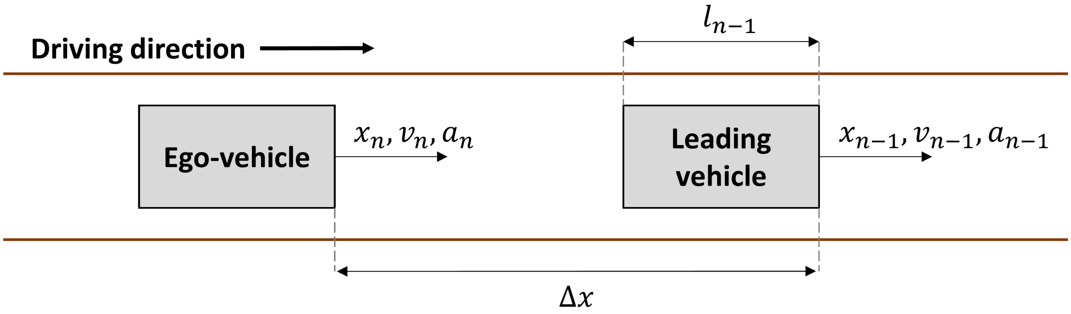


Figure 1.6: Essential notations in microscopic car-following traffic flow models.

Recent research on driver modeling extensively uses optimal control theory, with a specific focus on model predictive or quadratic controllers. These researches propose different driving modeling methods, with complex feedback systems, linear and non-linear control laws, preview abilities. These models can be fit to measured data thanks to system identification methods [2,46–49]. These methods can include advanced neuromuscular dynamics, sensory dynamics, physical and biochemical limitations, leading to complex neuro-physiological processes models (see Figure 1.7). They provide an interesting understanding of the human behavior, but they have the drawback to usually required large sets of recording data, leading to an averaged-over-time behavior reproduction. Optimal non-linear vehicle control is another frequent approach to driver modeling [49–51]. Finally optimal control and stochastic optimal control can be used to optimize some aspects of the driving such as the energy consumption of the vehicles based on the road topology and the vehicle powertrain [52–58]. These optimal control based methods can provide good specialized models, and perform well for energy consumption minimization and professional driver behavior reproduction. Nevertheless their efficiency can be limited when dealing with non-professional drivers, whom most important input stimuli and driving motivation can drastically vary [2].

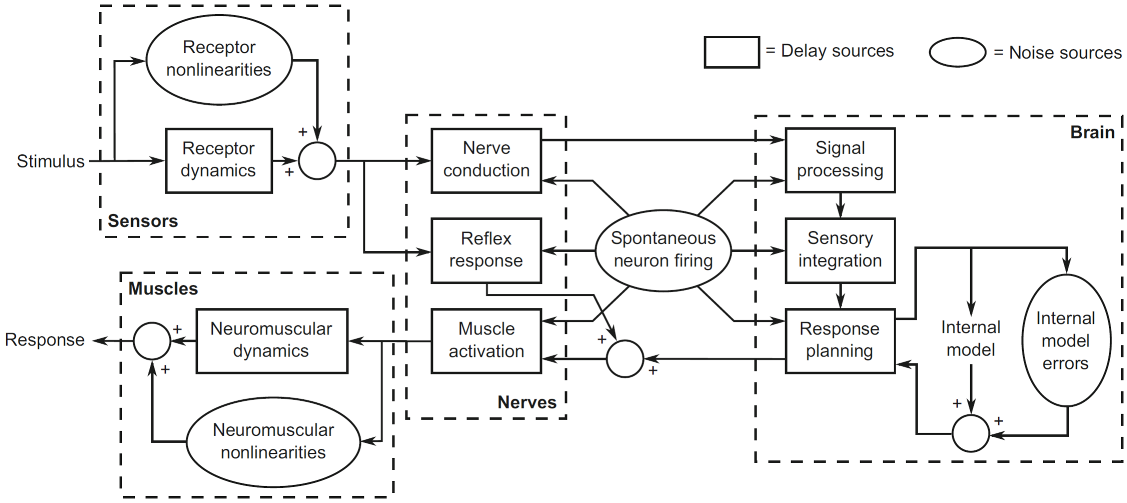


Figure 1.7: Diagram representative of the main neurophysiological processes involved in driving task. ”Stimuli” are the human system input and ”Response” the human system output. Figure from [2].

Lastly, numerous researches are studying driving behavior modeling based on direct identification from recorded data. These approaches are not going as deep as former researches on the understanding of the human behavior, but instead use conventional modeling methods, such as Hybrid Dynamical Systems Models (HDSM), fuzzy logic approaches, Hidden Markov Models (HMM) and Neural Networks Models (NNM) [59–65]. Hybrid dynamical system models are an extension of dynamical system model, using dynamical system models together with a mode switching strategy (decision taking process) to model system where the different states are ruled by different physical laws, or different inputs/outputs relationships. HMSMs can be seen as grey box models, where the structure is globally understandable and designed upon the analysis of the modeled system, and where the parameters are identified automatically based on data. The identification of grey box models can lead to the understanding of the physical meaning of the model’s parameter if their number is sufficiently low [59]. NNMs and HMMs are usually classified as a black box model as few prior model structure is assumed, and the resulting set of parameters automatically identified based on data does not provide much information on the model's processing manner. HDSMs are used for modeling complete system dynamics, including the decision taking processes. NNMs can be used both for situation modeling, implemented in a recursive form, and for decision taking tasks. Finally, fusy logic models and HMMs are commonly used for decision taking tasks.

Figure 1.8 shows simple representations of a three modes hybrid system models and of a neural network model composed of three input neurons and four neurons on a single hidden layer.

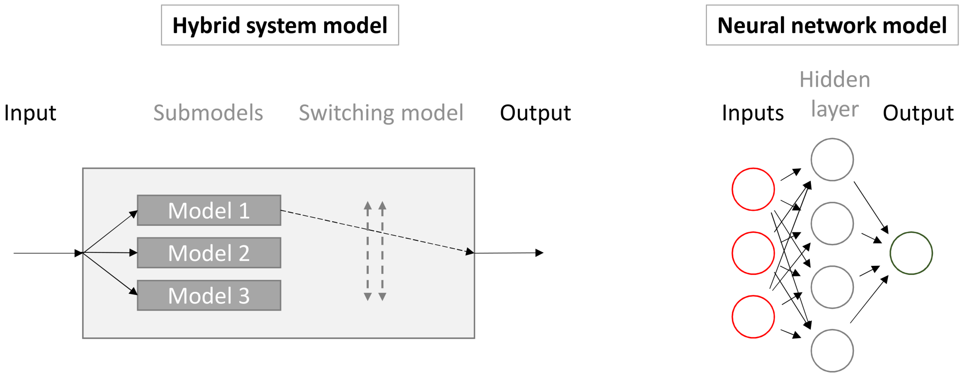


Figure 1.8: Simple representation of a general hybrid system model and an artificial neural network model.

A summary of this literature review of the driver behavior understanding and modeling field is proposed with the diagram of Figure 1.9. This diagram is modeling oriented. It includes the main modeling methods and the most encountered keywords observed in the literature, organized in a way to assist the reader to understand the ability of each modeling method. The included methods are focused toward dynamical systems, thus not including classification and clustering methods.

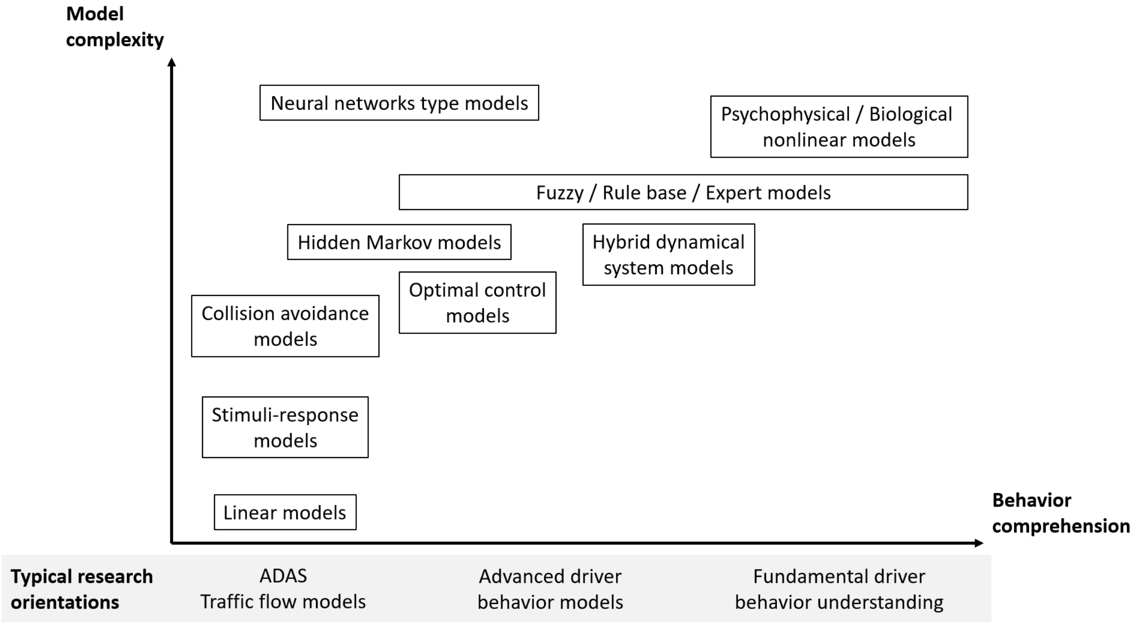

Model complexity
Neural networks type models
Psychophysical / Biological nonlinear models
Fuzzy / Rule base / Expert models
Hidden Markov models
Hybrid dynamical system models
Optimal control models
Collision avoidance models
Stimuli-response models
Linear models
Behavior comprehension
Typical research orientations
ADAS
Traffic flow models
Advanced driver behavior models
Fundamental driver behavior understanding

Figure 1.9: Vision of the main modeling methods for driver behavior analysis and reproduction. The horizontal axis represents increasing driver behavior comprehension abilities toward the right direction, and the vertical axis represents the model's structure complexity increasing upward.

In the following section, the most significantly influential models on the development of this doctoral thesis are detailed. At first traffic flow type models are presented represented by the microscopic traffic flow models Gipps model and Wiedemann model, followed by some sophisticated dynamical system models represented by the Piecewise Autoregressive eXogenous model and the Probability-weighted Autoregressive eXogenous model.

1.2.2 Microscopic traffic flow modeling

Microscopic traffic flow models show interesting properties that can be used to reproduce driver personalized longitudinal dynamics. Among the available types of traffic flow models presented in the previous section, the action-point and psychophysical models are seen as the most interesting and applicable They are detailed bellow.

Both of these model can be classified as expert models, or statistical data analysis non-linear hybrid system models. They can both reproduce drivers dynamics in classic driving situations. They are composed of modes representing typical driving situations, and have tuning parameters representative of the main characteristics of driving behavior. The transition between modes is based on fuzzy logic.

1.2.2.1 Collision avoidance model

The main collision avoidance model in microscopic traffic flow field in the Gipps model [41], created in 1981 by PG. Gipps. This model is discrete in time and continuous in space. This models is categorized as a collision avoidance or safety distance model. It means that a minimization is done between a velocity term representative of free driving and a safety braking term representative of car-following behavior designed based on safety criterion [3].

The basic concepts behind this model are that each driver plans its driving speed for the next time step after reaction delay t_{reac}, such that a safe stop can be achieved even in the event of a sudden braking of the leading vehicle. This model has been developed based on data measured on highway with average congestion levels [66].

According to the original publication [41], the model has been created with the following goals:

- The model should mimic the behavior of real traffic flow
- The model parameters should correspond to obvious characteristics of drivers and vehicles, so to be easy to calibrate
- The model should be calculated at a frequency equal to the inverse of the drivers action delay time.

Gipps model equation system is described in Equation 1.1:

$$v_n(t+t_{reac}) = \min\{v_a, v_b\}, \text{ with}$$
$$\begin{cases} v_a = v_n(t) + 2.5a_{(n)max}t_{reac}(1-\frac{v_n(t)}{v_{n0}})\sqrt{0.025+\frac{v_n(t)}{v_{n0}}} \\ v_b = b_n t_{reac} + \sqrt{b_n^2 t_{reac}^2 + b_n\left[2(s_{n0}-\Delta s(t)) + v_n(t)t_{reac} + \frac{v_{n-1}(t)^2}{b_{(n-1)max}}\right]} \end{cases} \tag{1.1}$$

where $v(t)$ is the vehicle number n velocity, $\Delta s(t)$ the relative distance between vehicles, and the other parameters as in Table 1.1. As several models will be proposed in this thesis, the original notations of this model is preserved. Thus Gipps model parameters notations have meaning only in this section.

Table 1.1: Gipps model notations

Desc.	Desired velocity [m/s]	Stopping distance to lead vehicle [m]	Reaction time [s]	Maximal acceleration $[m/s^2]$	Desired deceleration $[m/s^2]$	Maximal deceleration $[m/s^2]$
Variable	v_{n0}	s_{n0}	t_{reac}	$a_{(n)max}$	b_n	$b_{(n-1)max}$

The acceleration equation v_a of the Gipps model is based on data fitting, and the deceleration equation v_b is an interpretation of driving maneuvering.

- The first equation of the model has been determined to fit the acceleration/velocity envelope of moderate traffic arterial road data.
- The second equation of the model has been designed to enable a stopping safety distance, with a driver reaction time t_{reac} and a delay safety margin θ.

Stopping location of the vehicle is based on the deceleration coefficient b. See Equation 1.2:

$$s^* = s(t) - \frac{v(t)^2}{2 * b} \tag{1.2}$$

where s^* is the stopped location, $s(t)$ the initial braking position, $v(t)$ the initial braking velocity, and b the braking deceleration (negative value).

By adding the driver reaction time t_{reac}, Equation 1.2 becomes:

$$s^* = s(t) - \frac{v(t + t_{reac})^2}{2 * b} + \frac{v(t) + v(t + t_{reac})}{2} * t_{reac} \tag{1.3}$$

The value s^* in Equation 1.3 can represent the minimum following distance. Gipps model adds a safety delay θ during which the vehicle does not start to slow down. The new formulation is showed in Equation 1.4.

$$s^* = s(t) - \frac{v(t + t_{reac})^2}{2 * b} + \frac{v(t) + v(t + t_{reac})}{2} * t_{reac} + v(t + t_{reac}) * \theta \tag{1.4}$$

where θ in a safety delay.

The purpose of this formulation is to avoid impact, thus the model should fulfill the inequality 1.5:

$$s^*_{n-1} \geq s^*_n + s_{n0} \tag{1.5}$$

where n corresponds to the vehicle number and s_{n0} to the desired stopping distance to the front bumper of the lead vehicle.

Equation 1.5 leads to Equation 1.6:

$$s_{n-1}(t) - \frac{v_{n-1}(t)^2}{2 * b_{n-1}} \geq s_n(t) - \frac{v_n(t+t_{reac})^2}{2 * b_n} + \frac{v_n(t) + v_n(t+t_{reac})}{2} * t_{reac} + v_n(t+t_{reac}) * \theta + s_{n0} \tag{1.6}$$

b_{n-1} cannot be assessed directly, and is replaced by a maximal observed deceleration $b_{(max)n}$. $\theta = \frac{t_{reac}}{2}$ is a sufficient model stability condition if $b_{(max)n} \leq b_{n-1}$ and $b_n \leq 0$ [67,68]. Obtaining $v_n(t+t_{reac})$ involves resolving a second or inequality. The negative velocity term is rejected.

Equation 1.6 then becomes Equation 1.7:

$$\begin{aligned} & v_n(t+t_{reac}) \leq b_n t_{reac} + \sqrt{b_n^2 t_{reac}^2 + b_n \left(2\left[s_{n0} - \Delta s(t)\right] - v_n(t) t_{reac} - \frac{v_{n-1}(t)^2}{b_(max)n}\right)} \\ & \text{with } \Delta s(t) = s_{n-1}(t) - s_n(t) \end{aligned} \tag{1.7}$$

Δs represents the range (distance) between the leading vehicle and the following vehicle.

Finally, in Gipps model, Equation 1.1, v_b is considered to be the high boundary of $v_n(t+t_{reac})$ in Equation 1.7.

Gipps microscopic traffic flow model is used in commercial traffic flow simulation software, and it has been discussed, calibrated, and survey by numerous scientific papers [3,34,66,67,69,70]. This number of studies enables to have a good understanding of its functioning. This model is used in Section 3 and Section 5.

1.2.2.2 Psychophysical model

Psychophysical microscopic traffic flow models are a class of models based on the understanding of the human behavior. Instead of finding a global equation representing the average behavior of the dynamics of a driver-vehicle entity, the model is composed of modes representatives of the different driving tasks. In this section the Wiedemann model is introduced, as an example of trade-off between psychology inspired driver modeling and simplicity due to the traffic flow modeling usage.

The Wiedemann model has been create by R. Wiedemann in 1974 [42]. It has been object to numerous scientific researches, calibrations studies, and is now integrated in a commercial traffic flow simulation software [3, 71–73]. This model considers that the driver is influenced by two main factors: the distance to the leading vehicle (range) $\Delta s = s_{n-1} - s_n$ or R, and the relative velocity (range-rate) to the leading vehicle $\Delta v = v_{n-1} - v_n$ or RR. Based on this two-dimensional space, hyperplanes are defined as mode boundaries, and vehicle dynamics lows are defined for each model based on statistical observation of the driver dynamics. An interesting aspect of the original formulation of this model is that the author defines parameters of these mode separations and modes equations as random variables. The functioning of the model is illustrated in Figure 1.10.

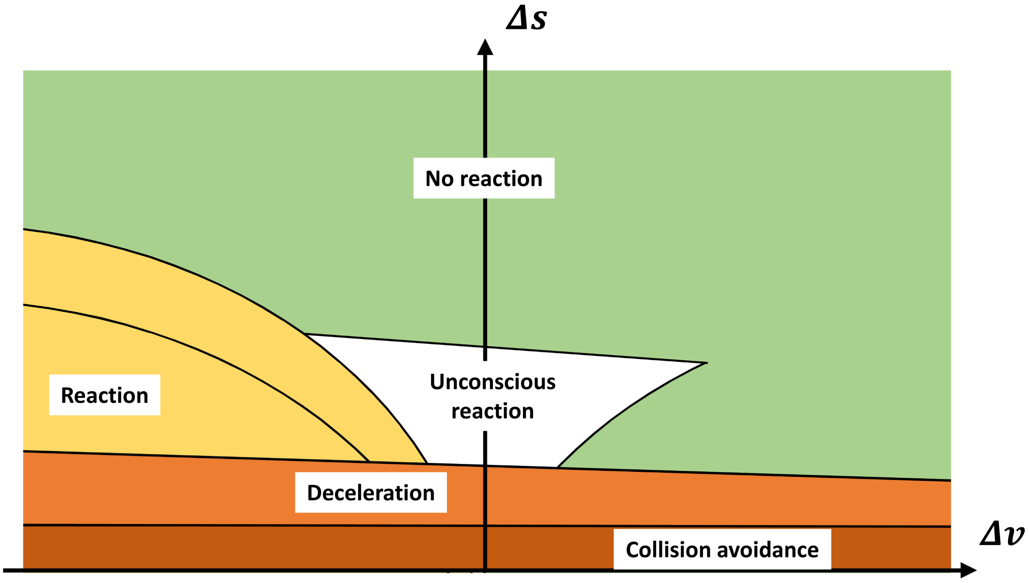


Figure 1.10: States diagram of Wiedemann car-following model.

Wiedemann model is constituted of five main driving modes representative of four driver states (see Figure 1.10). The "no reaction" mode correspond to free driving. The driver drives without taking information from the other vehicles in the flow. At low enough range, if the range-rate decrease, the "reaction" mode corresponds to a smooth deceleration task of the driver to adjust its velocity. If the range to the leading vehicle is too low, the "deceleration" mode represents a driver task to decelerate the vehicle. If the range is very low, a "collision avoidance" mode is activated to express the strong braking task of the driver.

Mode separation and driving dynamics of Wiedemann model are defined by non-linear equations, adjusting the vehicle acceleration depending on the environment. These equation have been defined based on observation of driving data, and correlation analysis between the influence of the environment to the driver response. Modes parameters are described as random variables, and are sampled at each mode jump so to represent the human stochastic nature. Unfortunately, due to the high number of parameters involved in this model (11), and the non-linear nature of the mode separation and modes dynamics equations, this model is difficult to identify directly from short-duration driving measurement data. Moreover, stability analysis and model output discontinuity are hard to overcome. Nevertheless, the Wiedemann model is a good inspirational source for behavior personalized driver modeling.

After this literature review on general driving behavior research, the following section proposes a summary of the general aspects of the human driving behavior based on the literature and on the author's experience, and an overview at the importance of data collection for the understanding of the human driving behavior.

1.3 Essential elements on the human driving behavior

This section proposes a summary on the human driving behavior, a discussion on naturalistic driving data collection in a real-world environment, and finally a selection of the driver modeling approach. More information about the collected data is provided in Chapter 2.

1.3.1 Perspective on the human driving behavior

As explained previously, the human driver includes a number of characteristics that have to be modeled. According to former studies (see section 1.2) and to the analysis of the numerous measurement campaigns done for this research work (see sections 2.2 and 2.3), it can be found that the key points of the human driver behavior are the following:

(a) The driver's decisions and commands are related to the environment after a reaction delay.

(b) The driver's decisions and commands are continuous in time.

(c) The driver's behavior is not perfectly consistent: two actions at close in time might not lead to the exact same driver response.

(d) The driver's behavior can significantly vary over long driving durations.

The characteristics **(a)** and **(b)** are related to the dynamical nature of driving. Driving is a process based on past-time and current environment information and on the past-time and current internal driver states. As explained in the literature review, commands requested by the driver are prone to three main different types of delays: the information acquisition delay from the environment, the brain processing delay, that correspond to the command of the next operation based on the current driver state and the external information, and the physical command operation delay, that correspond to the ability of the driver to apply the desired operation to the vehicle by steering and pushing the pedals. These delays are the reason of the past-time relationship between an control action required by the driver at a time t and the observation of the outside world at time $t - \tau$, τ representing a simplified reaction time of the driver. Moreover, driving is a continuous in time process. The driver's actions evolution are based on the driver's ability to adapt his current mindset. This means that each new action taken by the driver is based on the decision he is taking based on the external information, and on the current action he was doing. As such, it can be said that the driver's operations are related to the environment information after a reaction delay, and that the driver's vehicle command is related to the previous driver's commands.

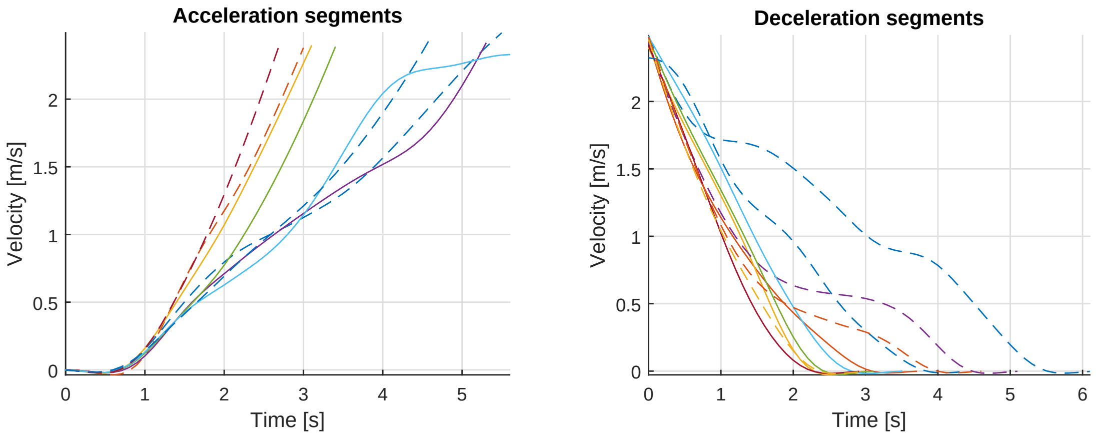


Figure 1.11: Example of collected data of start and stop situation at traffic lights. One driver, one vehicle, total recording duration 20 minutes. Dashed lines indicate free driving, continuous lines car-following situation.

The human driver characteristic **(c)** has an influence on the method of the model validation. In the case of the modeling of a systematic process, direct comparison of the model output to the reference data can be done. In the case of the human driver modeling, as it can be observed in Figure 1.11, human drivers tend to have a significant dispersion in the making of simple driving tasks. A statistical approach or a reference index has to be selected to be able to represent the performance of the generated model. In the case of this study, focus is done on the selection of a reference index: the vehicle energy consumption. This indicator lead to the generation of a value averaged over time, and representative of the vehicle usage on varied scenarios.

The final proposed item **(d)** regulates the amount of data that can be used to identify a model, as long duration data measurement are prone to behavioral changes. The data packets used for model identification have do be kept short in time.

Dependency of the driver behavior to the driven vehicle has also been studied in this thesis. A data measurement campaign (see Section 2.3.2) including three vehicles types and four different drivers had been conducted. No significant difference of recorded driving dynamics could be related to the vehicle type. This study lead us to the conclusion that after a duration, corresponding to the vehicle's dynamics understanding by the driver, the driver can compensate the vehicle's dynamical response. As driver's driving intention cannot be differentiated from the the vehicle's dynamical response in our recorded data, in this study the driver-vehicle behavior is modeled as a single entity [16].

1.3.2 Measurement data collection

To be able to analyze driving behavior, measurement data collection has to be done. The following challenges are faced in the naturalistic data collection:

(a) The driver should have a natural driving behavior.

(b) The measurement equipment should not be intrusive.

(c) The scenarios should be stable in time.

(d) The external factors data should be collected.

To fulfill **(a)**, the examinee should be comfortable with driving. A beginner driver, in learning process, is not a good candidate for our work. The examinee should experience the driven course and the driven vehicle for a certain amount of time. Once that the examinee does not need to focus on directions and when he feels comfortable with the vehicle dynamics, the recording of experimental data can be start. In the lead experimental campaigns (see Chapter 2), 20 to 40 minutes were necessary for the examinee to get comfortable with the driving environment and the vehicle dynamics.

In **(b)**, to be able to get naturalistic measurement data, measurement equipment should not disturb the examinee's behavior. The data collection equipment should not be in the driver's field of view, nor should the driver feel intrusively watched over. To do so, our equipped vehicles had environment sensors out of the field-of-view of the examinee, and apart from one vehicle, no camera was pointing at the driver. Moreover, the examinee were left alone in the vehicles.

The item **(c)** is related to the confidence interval of the measured data. A stable environment is required to be sure that the measured data is only correlated to the driver behavior. Environment stability is defined by the consistency of the environment impact on the measured vehicle dynamics duration over time. This element dictates the measurement duration and the experiment location. Short measurement durations somehow ensure a situation consistency over time. The experiment location has to be chosen so to be representative of a type of environment, while having periods of the day with consistent traffic flow density and composition. To ensure that these points where fulfilled in our experiments, locations were thoughtfully investigated, and recommendations where asked to traffic-flow modeling field scientists.

The external factors of the item **(d)** are represented by the road type and topology, the traffic flow, the examinees' condition and the sensor's limitations. The road type is related to the applied driving rules, especially the maximal driving speed, and the road topology represents the road geometrical aspects: turning angle, width, lane marking, traffic signs. The traffic flow represents the other vehicles on the road. The examinees' condition represent the mental state of the recorded drivers. Finally, the sensor's limitations represent the accuracy of the collected data, expressed as the amount of noise, and the validity of the data in complex situations (for example in leading vehicle distance measurement).

1.3.3 Importance of the selection of the driving situation

This thesis is focused on the reproduction of human driver's behavior, with a focus on personalized energy consumption. Based on this target, it has been important to specify clearly the context of the study and the represented situation.

The data recorded for this study being located in Japan, we will focus on Japanese roads topology. Japan has a specific road topology, laws and demographics. Most of the speed limits are kept low (40 kph within town, 50 to 70 kph for main multiple lanes road in town, 80 to 100 kph for highways), and roads at speed are mostly composed of strait lines within cities (see Figure 1.12). It could be observed that most of the collected data has low lateral acceleration (see Section 2.3). Moreover, due to the high density of the traffic flow in metropolitan areas, most of the collected data is falls under the car-following situation. Due to the modeling difficulty of low speed maneuvering, and its low impact on vehicle energy consumption, only driving above five meters per second has been kept. Based on these observations, the work done in this thesis is focused on car-following situation with no consideration of road topology on driving maneuvering. This point of view enabled us to analyze and model accurately large city streets and highway situations.

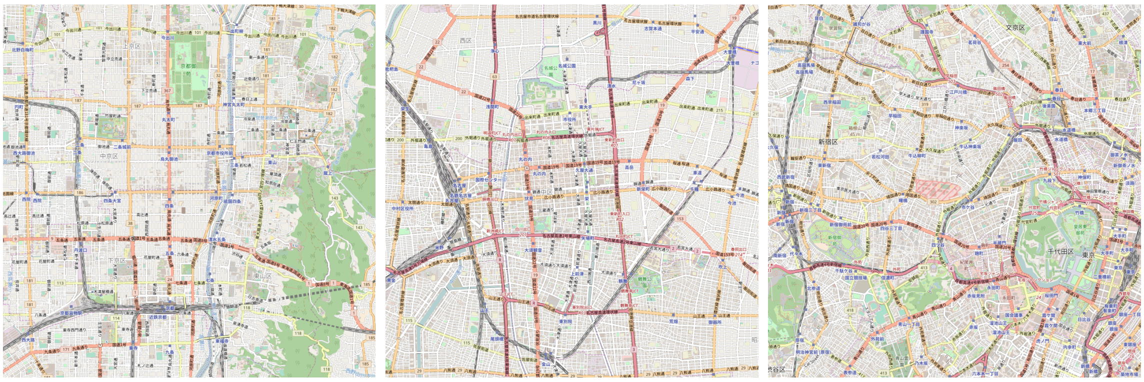


Figure 1.12: Example of Japanese city centers. From left to right, maps of: Kyoto, Nagoya, Tokyo. © OpenStreetMap contributors.

We believe that the structure of the driver-vehicle model should be comprehensive, and as each individual driver behavior tends to be influenced by different internal and external factors, the model should be flexible enough to learn and adapt to every driving sensibility [2, 16, 34, 38]. Thus the modeling approach should be compatible with the dynamical aspects of driving task, and parameters should be identified directly from measurement data to fit as much as possible to drivers personal behavior. Based on these elements, the modeling process used in this research is hybrid dynamical system models (see Section 2.4). The model inputs are carefully selected based on correlation analysis between the represented situation observable data and the desired model output. The modes of the hybrid model have been selected in such a way to represent the main states of the represented situation. By combining these two design rules and by keeping the hybrid dynamical model order low enough, it has been made possible to model comprehensively the driver-vehicle in car-following situation.

In the following section, the driving behavior modeling works studied during this doctoral research are detailed.

1.4 Goals and applications of the thesis

The three years of doctoral research lead to a thorough work on the understanding the human driving manner thanks to the measurement of driving data collection, to work on the modeling of personalized driving behavior, and finally it enables us to propose a method to analyze the stochastic nature of human driving.

The first achievement of this research work has been the development of a general framework for personalized advanced driver assistance system (ADAS). The second research outcome has been the development of a method for the evaluation of driver behavior personalized energy consumption. Finally, the last research deliverable has been a model identification method for understanding the driver's behavior through the study model parameters evolution over time.

In the following sections, summaries of the main research topics are proposed.

1.4.1 Advanced driver assistance system

Due to the financing of the doctoral thesis, the research work has quickly been oriented toward application propositions. Thus, the most straightforward approach has been to benchmark the abilities and range of applications of famous driver models, and of hybrid dynamical systems, on a simple application.

The first paper presented in the thesis proposes a framework for driving behavior personalized adaptive cruise control (ACC) [9]. The created framework enables to use any car-following model, and it is compared to classical ACC models used by the car industry and by commercial traffic flow models. As a conclusion, it can be observed that the use of hybrid dynamical system in the scope of ADAS is promising. It provides more advanced driver adaptation than classical models, and it is easily implementable as an online system in a real vehicle.

1.4.2 Energy consumption evaluation

Following to this first part, work has been focused on the understanding of hybrid dynamical system models, on the collection and processing of the experimental data, and on the research of relevant inputs and outputs to model driving behavior.

The second paper proposes an approach to reproduce driver personalized energy consumption in the car-following task [10]. The study is based on the works done on the multi-mode dynamical system Probability Weighted ARX model, extended toward driver-personalized modeling abilities. Data used for model parameters identification has been recorded by using a driving simulator and by doing real world experiments, with varied driver types. As a conclusion, the driving dynamics could be reproduced accurately, and the generated energy consumption values were representative of the driving behavior.

1.4.3 Driver behavior analysis

Finally, the last part of the research work has been focused on proposing a novel way to analyze and understand stochastic human driving behavior. This research lead us to benchmark existing ways to identify multi-mode model parameters, to analyze trends in data, and to combine time-dependent filtering and sub-optimal optimization.

The last paper proposed a new method for data-based model identification, where the parameter identification process can uncouple data frequency ranges [12]. The goal of this approach is to be able to identify the time evolution of model parameters, to then understand the behavior evolution of drivers over time. The proposed approach is based on a Markov Chain Monte Carlo and a genetic sampling method called Sequential Monte Carlo to estimate Bayes' rule. As a result of this study, the time-dependent parameters of PWARX models could be identified successfully, and the application to driving behavior analysis is discussed in [11].

1.5 Organization of the thesis

This thesis is organized in six main parts:

- Chapter 1: Introduction of the thesis. The first chapter is a presentation of the research background, including the literature review, and the perspective of the author on driver modeling. It also includes a short description of the main achievements of this doctoral thesis.

- Chapter 2: Description of the data based approach, of the measurement campaigns and of the multi-mode dynamical system models. The second chapter explains the purpose of data collection, the experiments and the data use for model identification. It also describes the two multi-mode dynamical system models used in this research.

- Chapter 3: Driving behavior personalized adaptive cruise control. This chapter is the first conference paper published by the author of this thesis (see Section 1.4.1).

- Chapter 4: Driving behavior personalized vehicle energy consumption evaluation. This chapter is the first journal article published by the author of this thesis (see Section 1.4.2).

- Chapter 5: Filtered Bayesian mode identification approach for driving behavior analysis. This chapter is the second journal article published by the author of this thesis (see Section 1.4.3).

- Chapter 6: Conclusions of the thesis.

Chapter 2

Data centric approach and multi-mode dynamical systems

2.1 Introduction

The goal of this thesis is to understand and to model the human driving behavior based on measurement data. The emphasis done on the data is due to its nature to shape the entire research. Studies based on statistical understanding of data or on psychophysical understanding of the human behavior can propose interesting models, but they are subject to a significant bias due to the author's understanding of the data (see Chapter 3).

To reduce this modeling bias, we opted for hybrid dynamical modeling, identified from measurement data. To lead to this approach, large data sets have been used to select a mathematical modeling framework, including decision making processes and dynamical system modeling, and then smaller data sets representative of individual drivers' behavior have been used to automatically identify these models. The resulting driving behavior models, hybrid model composed of a decision making process and a dynamical system models, can reproduce the recorded data as well as allowed by the identification algorithm and the model structure.

This approach enabled us to generate personalized driver models automatically, without requiring an individual understanding of each driving style. Due to the variability of the factors that determine driving manner, the flexibility proposed by this modeling approach could be difficult to achieve with other existing modeling methods.

As such, the data used in this approach is used for:

(a) Analysis of the driving behavior,

(b) Identification of model parameters,

(c) Validation of the model.

The item **(a)** involves two processes: selection of the appropriate model type and selection of the model inputs and outputs. This step is critical and it is the main possible author's bias of this approach. Selection of the model type has to be done based on the nature of the modeled process, and selection of the model structure (e.g. model order, decision making process type) can be done automatically by defining modeling scores. Selection of the model outputs has to be done based on the nature of the modeled situation, and selection of the model inputs can be done based on a inputs/outputs correlation study on a large data sets. In the case of car-following driving, the modeled situation being dynamical, the model has to be dynamical, and the presence of different driving states (driving situations) led the author to use hybrid type models (see Chapter 4).

The item **(b)** is related to two main elements: data processing and identification process. Data processing is important due to the automatic nature of the model identification. In the case of real world data, a particular attention has to be done on the data outlayers removal and on the signals filtering process. Data outlayers represent exceptional data, not representative of the desired model situation, that can induce wrong behavior of the model on certain situations. Signal filtering enables to preserve the representative dynamic of the recorded signal (e.g. reducing the level of noise in the learning data). These data processing items have a significant impact on the convergence rate of the identification process and on the quality of the final identified model. The identification process represents the mathematical approach used to find the model parameters to reduce the difference between the recorded data and the model output. Numerous approaches exist for system identification (see Chapter 5), and detailed analysis of some important criteria have to be done to select the appropriate method. To select the method, the user has to know if an optimal or suboptimal method has to be used, if the problem is globally convex or has numerous local optima, the problem size, and the required identification accuracy. Based on these elements, it is possible to select a model identification process with efficient identification duration and appropriate parameters accuracy.

Finally the item **(c)** is about the model validation. In data-based approaches, validation of the generated model can be done automatically. In machine learning field, a randomly picked set of the measured data is usually used for the model identification (training set), and the remaining part of the data is used for the model validation (test set). In the case of dynamical systems, a random selection of the data for learning and validation data cannot be done, due to the fact that time has to be considered in the creation of data segments. Moreover, hybrid dynamical models require an initial state, thus short time duration segments should be avoided to reduce the chance of overestimation of the model performance. In the case of car-following modeling, it could be observed that recordings time duration should be short to avoid having a large variation of the driver's behavior. As a result, driving model validation is more complex than in classical machine learning problems. In this study, the model validation is done by using a unique identification data set (training set identical to the test set), and by doing relative comparison of two identified models outputs on a single scenario (see Chapter 4). This approach allows to verify the good average performance of the identified model on known data, and its ability to represent relative behavior of different parameter sets on a single scenario.

Driving simulator and real-world measurements have been undertaken in this research. Measurement types, strong and weak points of the measurement methods, and usage of the data are detailed in the following sections.

2.2 Driving simulator measurements

The driving simulator (DS) enables to virtually reproduce a driving environment where all environment variables are controlled. The DS enable to get driving data without sensor noise nor undesired traffic flow interactions, and with scenarios that can be perfectly repeated. This last feature is important for behavior analysis, to be able to measure a single driver multiple times on a single scenario, and to evaluate the behavior of several drivers on a similar scenario.
Nevertheless, usage of the DS has some drawbacks, starting by the absence of driver's feeling of the vehicle acceleration. The lack of acceleration feeling can lead to lower driving consistency, and to velocity oscillatory behavior at while cruising. The second main drawback is the lack of feel of danger, that sometimes lead to more aggressive driving manners.

Based on these considerations, the data provided by the driving simulator experiments has been used for the testing of the parameter identification schemes, and for the validation of the model choices. Nevertheless the DS has not been used extensively for driver's behavior analysis and comparison.

The following sections present the experimental setup and two driving simulator measurement campaigns.

2.2.1 Experimental setup

The driving simulator is composed of three large projected screens covering a lateral field of view of 180 degree. The inside and outside mirrors are created with portable screens. The driver seat is positioned inside a commercial compact vehicle cockpit, and the steering wheel is also extracted from a commercial vehicle (see Figure 2.1).

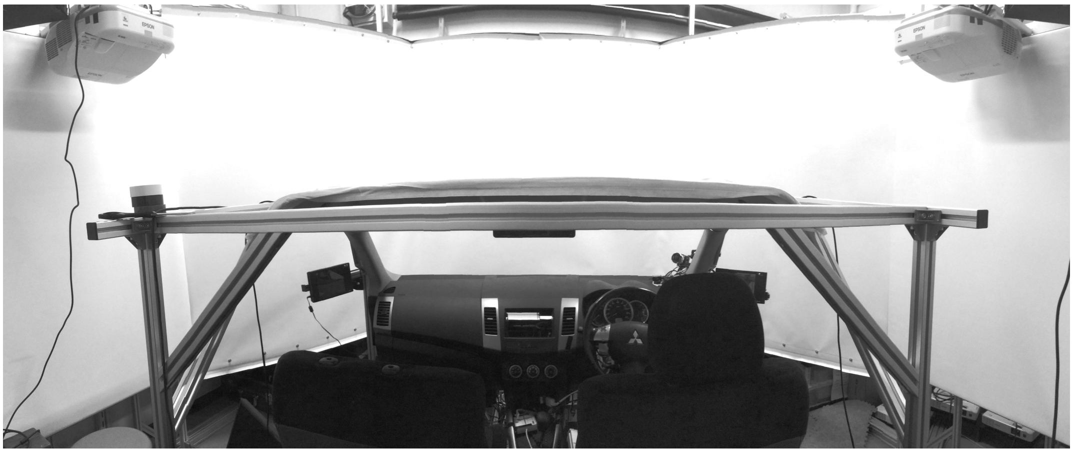


Figure 2.1: Global view of the driving simulator structure.

The driving simulators runs on a custom software setup based on the vehicle dynamics simulator CarSim. The visual environment is generated by the localization of CarSim's vehicle in a custom-designed 3D environment (see Figure 2.2). Sound is generated based on the vehicle's engine state. Data collection of the dynamical information of all the vehicles in the simulation is done at 100Hz.

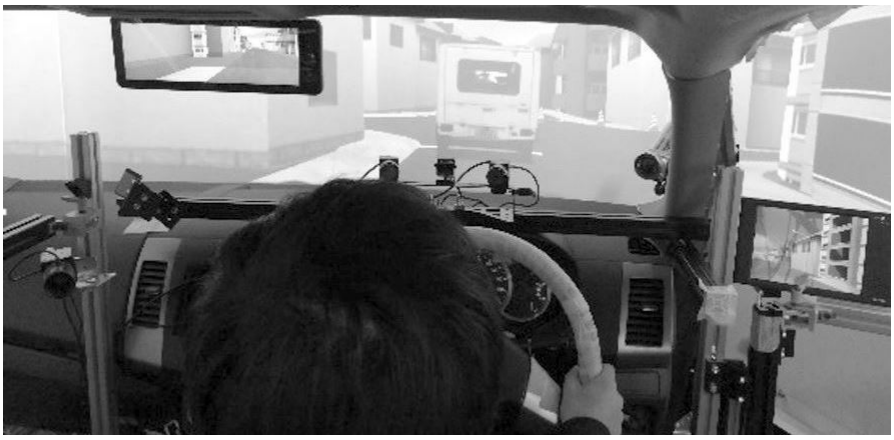


Figure 2.2: Examinee during a DS experiment.

For research purpose, two virtual environments have been used: a low velocity suburban environment and a high velocity highway environment. The suburban experiment has been done jointly with a real world experiment for comparative evaluation of the DS data quality. The highway driving DS experiment has been used to get data for driver model identification and comparative evaluation of driver's behavior. The most realistic results were provided by the highway environment due to the fact that highway driving is less prone to longitudinal accelerations, leading to a situation close to the real-world driving.

All experiments has been done in three phases:

- The examinee gets used to the vehicle for 15 minutes
- The examinee is recorded once
- The examinee is recorded a second time for driving behavior consistency verification.

The following sections provide details about the two DS experiments.

2.2.2 Suburban experiment

A first DS experiment has been done to verify the range of validity of the DS data. This experiment represents as accurately as possible the low velocity suburban experiment (see section 2.3.1). Figure 2.3 shows the road topology similarity between this DS experiment and the real-world experiment in section 2.3.1.

For this experiment, four driver were recorded. The driving scenarios were:

- Following driving. In this scenario, the driver was following a vehicle on the road. The data for the generation of this vehicle has been extracted from the experiment in section 2.3.1.
- Free driving. In this scenario, the driver was driving by himself on the open road.

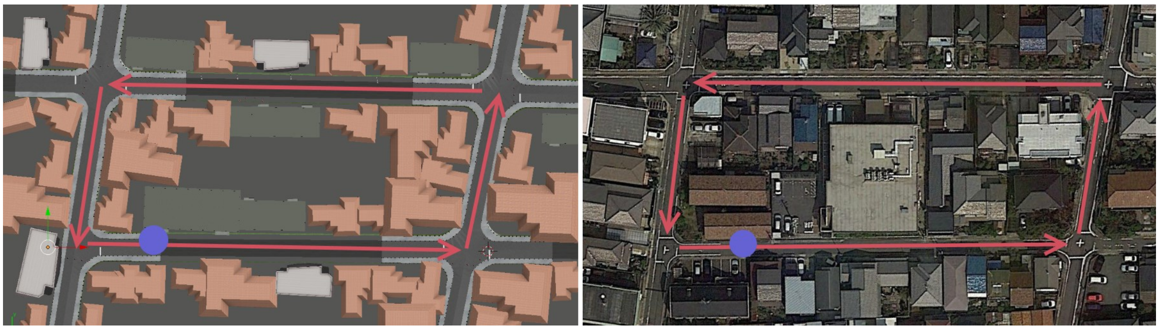


Figure 2.3: Comparison between suburban modeled environment and real environment. The blue spot represents the starting point of the experiment

As a result, the recorded driving dynamics were compared between both experiments. Concerns regarding the importance of physical feeling of the vehicle acceleration and importance of the notion of danger could be confirmed, as acceleration values differed between DS and real world measurements, and drivers were no careful at intersections. Thereby, it has been concluded that the type of scenario expressed in this experimental campaign was not compatible with this type driving simulator, and the measurement data from this experimental campaign has not been used for driving behavior modeling purpose.

2.2.3 Highway experiment

The DS highway experiment has been done jointly with the highway real-world experiment (see section 2.3.3) to collect driving following data for model identification. The highway driving situation seem to be one of the best scenario in DS to get driving data representative of real-world driving situation: the acceleration of the vehicle are low, and the driver spends most of time looking ahead. As explained previously, the main advantages of using this type of experiment on DS are the repeatability of the experiment and the avoidance of undesired external phenomena that would disrupt the experiment. Moreover, the time-efficiency of this type of experiment is much higher than a conventional experiment.

The designed environment was a 7 km long multi-lane highway oval loop, representing a large city airway. The leading vehicle dynamics represented typical driving patterns, such as truck driving, with velocities between 80 and 110 km/k, and passenger vehicle driving, with velocities between 90 and 130 km/h. Drivers were following this a vehicle by redundant sessions of 10 minutes.

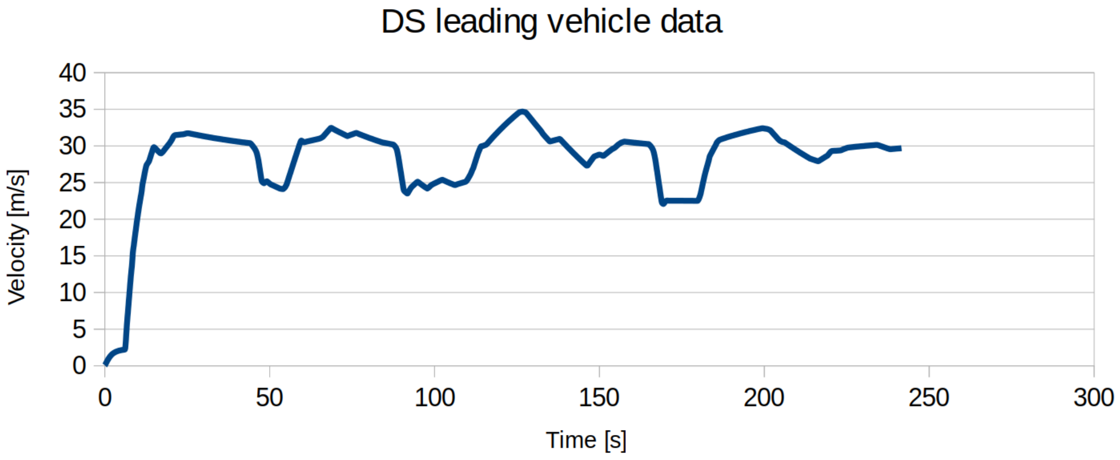


Figure 2.4: Example of highway driving data used for the leading vehicle.

The data collected in this experiment has been used for model identification in the articles [9, 10, 12].

2.3 Real-world measurements

Real-world experiments represent experiments done in open world environment in or around Nagoya city. These experiment have been done to analyze the human driving behavior, to select modeling frameworks, and to identify parameters of the selected models.
Three main real-world measurement campaigns have been done for this doctoral thesis:

- A suburban low-velocity experiment. The goal of this first experiment was to learn about the process of doing real-world experiments, to get a basic understanding of human driving dynamics in a residential area, and to be able to do a real-world/DS comparative study.

- A multiple-drivers multiple-vehicles multiple-environments experiment. The goal of this experiment was to create a comprehensive evaluate the influence of the driver, the vehicle and the environment on the driving behavior.

- A highway driving experiment. The goal of this experiment was to get good quality recording data, with repeatable leading vehicle velocity patterns on simple environment, with the aim to use this collected data for model identification.

These real-world experiments took place in environments representative of the most usual driving situations, ensuring that the researcher had a proper insight of driving behavior in common situations.
In the following sections, the aforementioned experiments are details, including information about the experimental plan and setups, the type of collected data, and the experiments results and interpretations.

2.3.1 Suburban experiment

The goal of this first experimental campaign was to learn about the process of doing real-world experiments, to have a basic understanding of human driving dynamics in a residential area, and to compare real world and DS measurement data.

For this experiment, a micro-mobility electric vehicle, the Toyota Autobody COMS, has been instrumented by the members of Suzuki laboratory. The vehicle has been equipped with:

- A front robotic-type Lidar, for leading vehicle range detection.

- A Mobileye system for on line localization, traffic signs detection and leading vehicle type detection.

- A GPS sensor for absolute position localization.

- A steering angle sensor and pedal position sensors for precise turning and fine control measurement.

- Right and left wheel encoders for precise longitudinal dynamics measurements.

The data from this measurement was collected though a compact data acquisition module connected to an integrated PC.

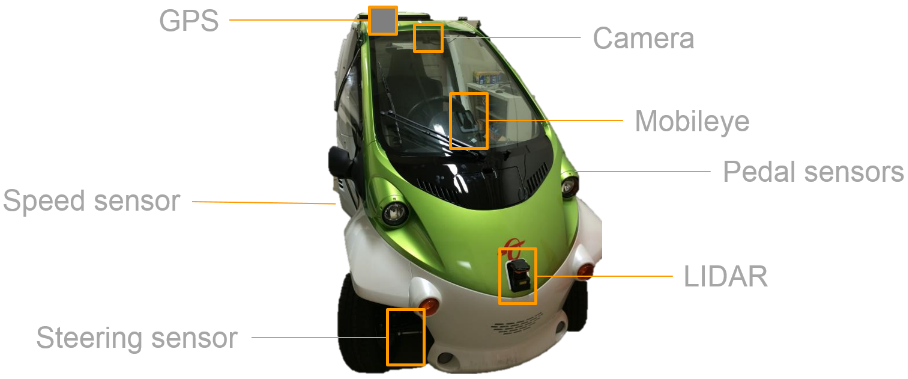


Figure 2.5: Micro-mobility electric vehicle instrumentation.

In this experiment, five examinees have been recorded on a short track, with various driving situation. The examinees have been recorded free driving and while following a leading vehicle. They also had to drive freely with eco-friendly and aggressive manners (see Figure 2.6).

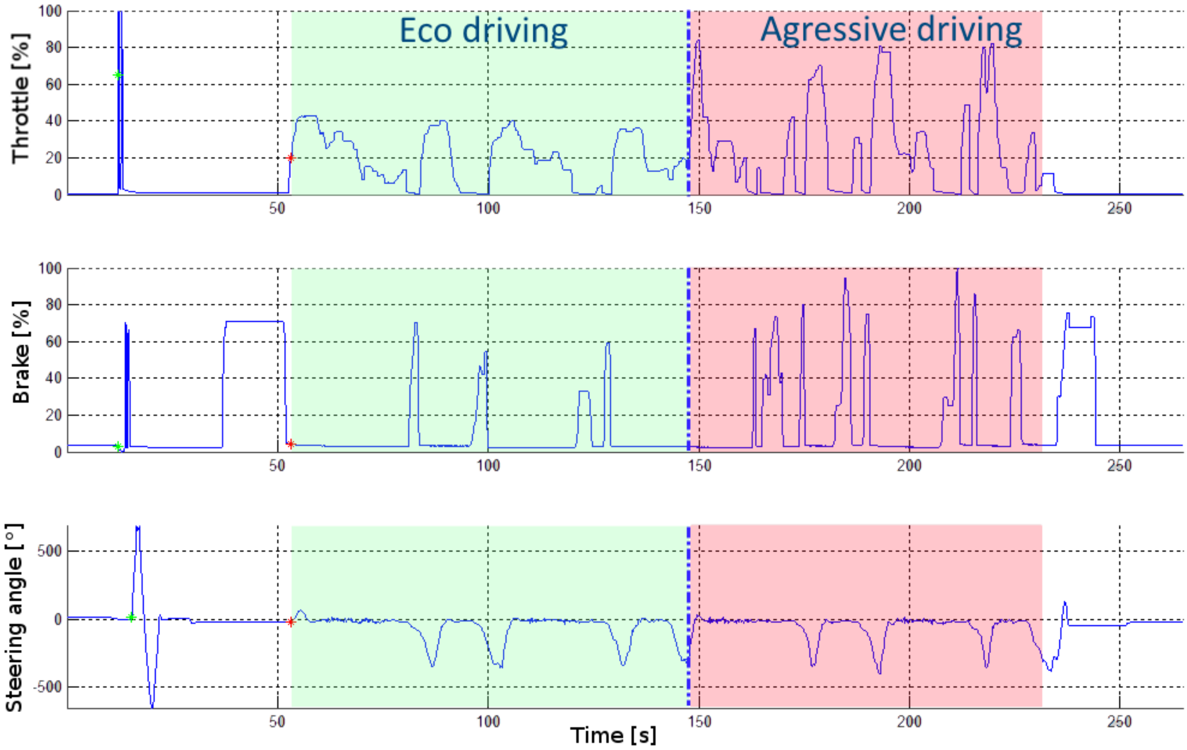


Figure 2.6: Observation of the difference between an eco-friendly behavior and aggressive behavior on the same path in a Toyota Autobody COMS.

This experiment could be used successfully to compare DS and real-world driver dynamics, as seen in section 2.2.2. This measurement has also been rich of understanding about the making of an experimental setup. In this experiment, defaults in leading vehicle sensing did not enable us to use this data for driving following model identification. The issue has been identified as being the Lidar setup, and this sensor has not been used for the following measurements.

2.3.2 Behavior comprehension experiment

The goal of the behavior comprehension experiment was to understand the impact of the vehicle and the environment on different driving behaviors, and to get a global perspective on driving behavior in typical driving locations around Nagoya city. To do so, four different drivers have been driving three vehicles on three varied environments over more than a week, resulting in 36 different recorded situations and 60 data packages.

In the following paragraphs, first the experimental setup and planning is explained, then the type of measurement data gathered are shown and interpreted, and finally conclusions are drown.

2.3.2.1 Planning and experimental setup

Three main elements have been considered for the design of this experiment:

- The experimental setup should not influence the driving behavior.
- The drivers should be accustomed to the driving environment and to the driver vehicle.
- Driving durations should be kept short to avoid driver's tiredness that could lead to behavioral change.

These elements lead to the installation of non intrusive measurement instruments, to insure the naturalistic nature of the recorded data, some preparatory session to help the drivers to feel accustomed to the driving environment, and driving session not longer than 30 minutes.

The three driving environments were representative of city-center driving, mixed-urban driving and countryside driving. All routes were designed to last between 20 to 25 minutes depending on the density of traffic flow.

The mixed-urban route was composed of a large mix of road types, from single lane single way to four lanes arterial roads. It is representative of the variety of road that could be used by a commuting driver in Japan. These situations are depicted in Figure 2.7.

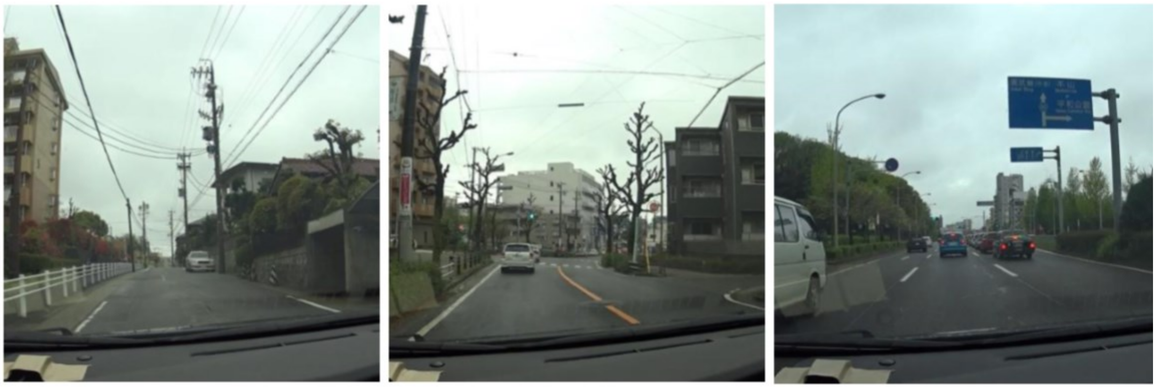


Figure 2.7: Overview of the mixed-urban route.

The countryside road was composed of single lane double way without road lines, and two lanes roads. It can represent the type of road used by persons living outside major cities in Japan. These situations are depicted in Figure 2.8.

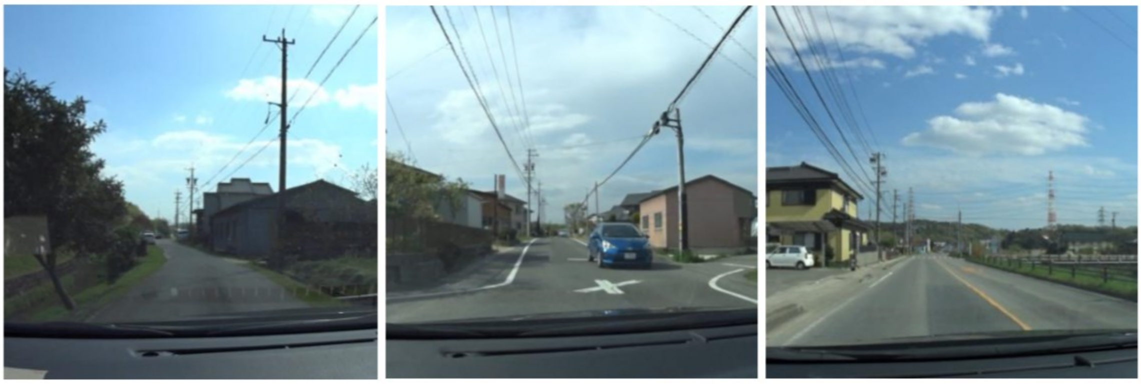


Figure 2.8: Overview of the countryside route from on-board camera.

The city-center route was composed of various roads from single lane one way roads to arterial roads. It represents most driving situations in a Japanese city-centers. These situations are depicted in Figure 2.9.

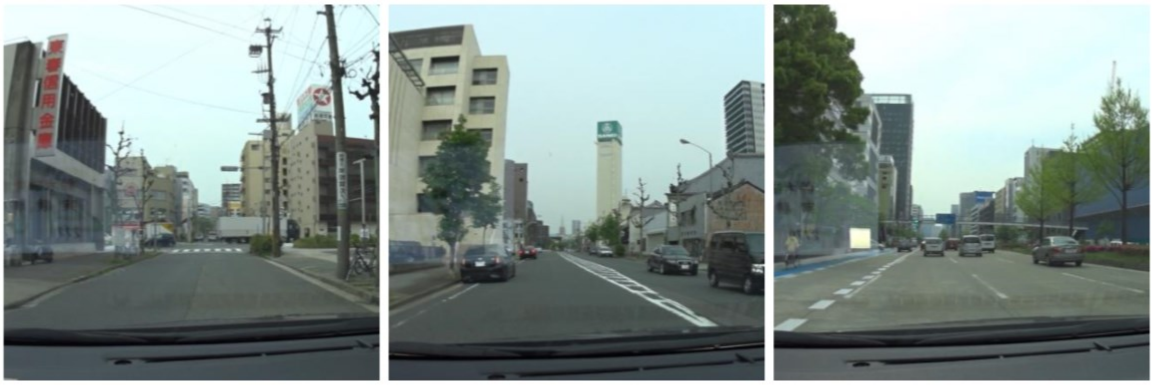


Figure 2.9: Overview of the city-center route from on-board camera.

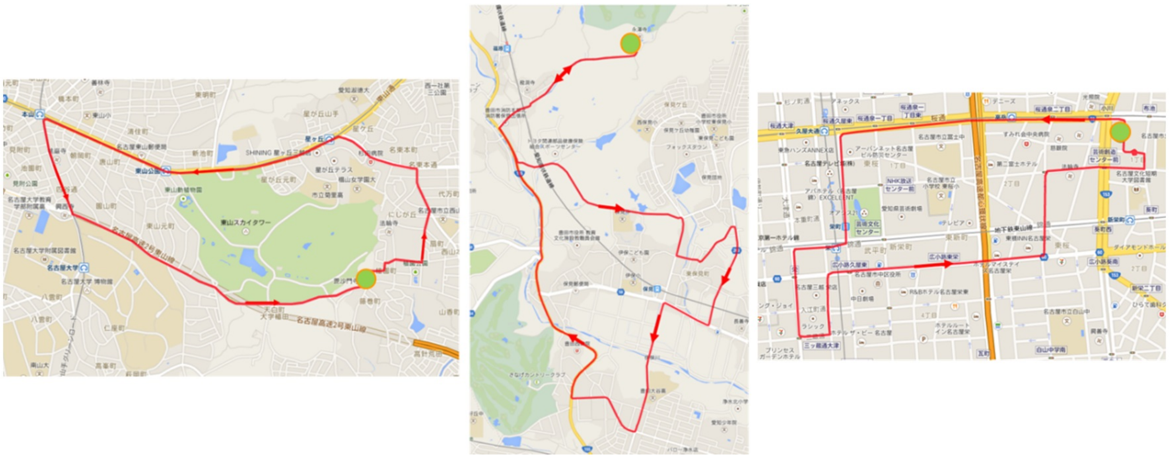


Figure 2.10: Routes maps. From left to right, the mixed-urban, the countryside and the city-center routes from on-board camera.

Table 2.1: Routes basic information.

	Mixed urban area Higashiyama	Countryside area North Toyota	City center area Sakae
Length	6.85 km	11.5 km	4.06 km
Starting point	35.15395, 136.98818	35.15499, 137.13238	35.17356, 136.92022

Figure 2.10 provides the route map of the selected driving environments, and Table 2.1 their basic characteristics.

The vehicles of this experimental campaign have been selected to represent the main types of drivetrain available on the Japanese market. Thus choice has been done to use an automatic gearbox internal combustion engine vehicle, the Toyota Auris, a hybrid vehicle, the Toyota Prius, and a vehicle used in pure electric mode, the Toyota Plug-in Prius (see Figure 2.11). This last vehicle is not a pure electric vehicle, but pure electric mode can be forced, and the battery allows more than 15 kilometers of electric range, and normal acceleration abilities.

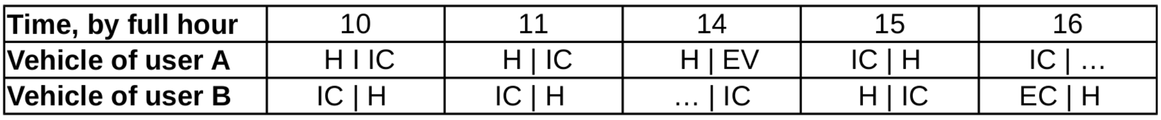
Figure 2.11: Vehicles used for the experiment.

The population of this experiment is composed of four examinee with large age and driving style variations. All of the drivers were used to driving, to enable good consistency along the driving experiments. Description of the examinee is given as follows:

- Age: 30, Gender: Female, Driving for more than 5 years.
- Age: 25, Gender: Male, Driving for more than 5 years.
- Age: 35, Gender: Male, Driving for more than 15 years.
- Age: 60, Gender: Male, Driving for more than 30 years.

The planning was composed for 6 full days of experiment, as one day was required to record two drivers on one environment on the three vehicles. The planning of one day of recording is shown in Table 2.2. The morning sessions were used for route learning and driver adaptation to the vehicles. Afternoon sessions were dedicated to data recording. The examinee could drive the electric vehicle only once due to the capacity limitation of the vehicle's batteries.

Table 2.2: One-day experiment timetable. IC stands for internal combustion engine vehicle, H for hybrid vehicle and EV for electric vehicle.



Time, by full hour	10	11	14	15	16
Vehicle of user A	H I IC	H | IC	H | EV	IC | H	IC | ...
Vehicle of user B	IC | H	IC | H	... | IC	H | IC	EC | H

Two types of sensing platform were installed in the vehicles:

- Prius:
  The prius has been equipped with a millimeter-wave radar in the front, small cameras to record low definition video of the environment, a CAN acquisition system, a Mobileye sensor, a GPS measurement unit and a CAN acquisition system.

- Plug-in Prius and Auris:
  These vehicles have been equipped with dual front facing cameras, a Mobileye sensor, an acceleration sensor, an GPS sensor and a CAN bus acquisition system.

The main difference between these setups is that only the Prius is equipped with a good quality relative distance measurement tool. Mobileye system provides interesting line positioning information, but very partial information on the road situation. As this experiment is focused on the comparison on vehicle dynamics data, all vehicle were equipped with similar acceleration and velocity sensing abilities. The internal combustion engine vehicle's setup is shown in Figure 2.12.

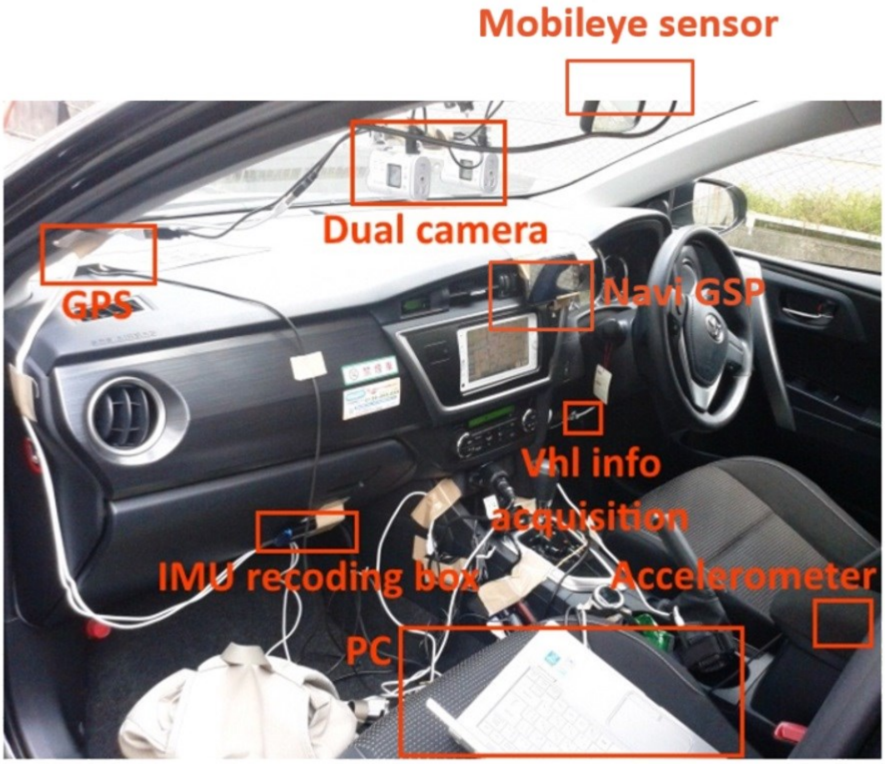


Figure 2.12: Internal combustion engine vehicle sensing setup.

2.3.2.2 Results and interpretation

In this measurement campaign, data has been recorded on 3 environments with 3 different cars with 4 different drivers. The number of repetitions for each recording is 2. Based on this data, interpretation of the influence on the environment and the vehicle on the driving behavior has been done. Nevertheless, the population of drivers is not large enough to use statistical methods to analyze the driving styles.

A few analysis have been done to try to understand:

- The influence of the vehicle type on the driving behavior.
- The influence of the environment on the driving behavior.
- Low velocity driving behavior.

Influence of the vehicle type on the driving behavior

Analysis of the influence of the vehicle type on the driving dynamics has been done from the observation of the histogram of the velocity, histogram of the longitudinal acceleration, and an acceleration/velocity 2d histogram mapping. This statistical

approach that does not directly analyses the driving signal but the frequency of occurrence of the situation.
For example the velocity space has been divided in 5km/h cells, and the amount of data in each cell is analyzed.
This analysis has been repeated on 4 drivers in 2 different environments.

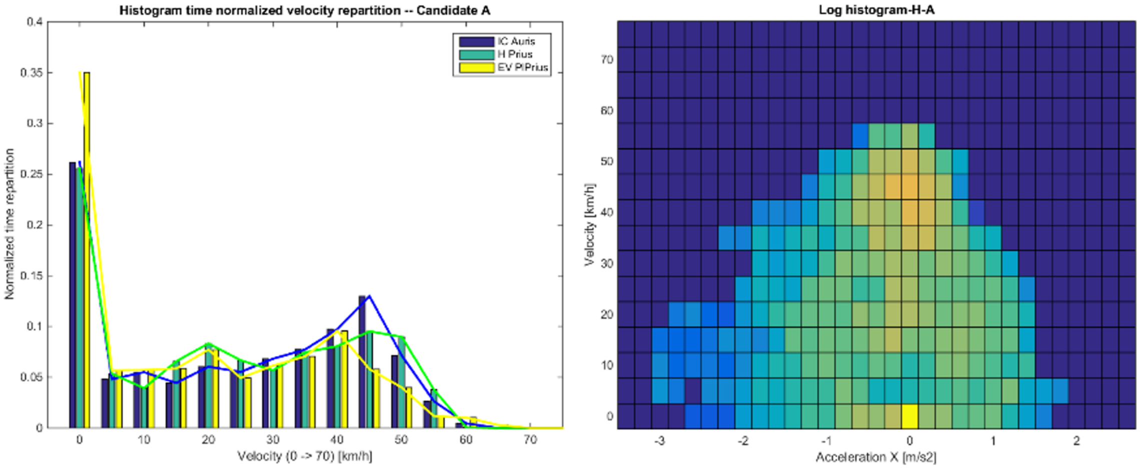


Figure 2.13: Example of histograms used for this analysis. On the left velocity histogram of a driver on a route for 3 different vehicles. On the right, acceleration/velocity 2D histogram.

Data observation lead to the following interpretation:

- EV vehicle seemed to be driven slightly slower.
- IC vehicle usage lead to slightly higher accelerations at low velocity. This much be due to the more brutal behavior of the automated gearbox compared to the electric and hybrid powertrains.

Conclusion: The type of driven vehicle did not influence much the driving dynamics and behavior of the examinees. Once the learning phase is finished, drivers tend to apply their own driving style on all selected vehicles.

Influence of the environment on the driving behavior

The same approach has been used to understanding the impact of the driving environment on the driving behavior. Longitudinal velocity, and longitudinal and lateral acceleration have been investigated. General statistics have also been extracted from the data (see Table 2.3). The used data is based on 3 environment and 4 drivers with the hybrid Prius vehicle.

Table 2.3: Statistical of driving dynamics in the studied environments.

	City-center	Mixed-urban	Countryside
Stopped time	47%	27%	18%
Average velocity	11 km/h	23 km/h	29 km/h

The conclusions are:

- In Japan, car are stopped 50% of the time in city center, 30% of the time in the mixed urban environment, and 15-20% in the countryside environment.
- There is less low accelerations in city center. This must be due to the fact that these low accelerations often happen at higher velocities.
- There is more lateral acceleration in countryside environment, due to the road shape at driving velocity.

Nevertheless, the longitudinal acceleration behavior of the drivers did not seem to be impacted by the type of environment.

Low velocity driving behavior

Take off and end of braking situations are zones sensible to the driver behavior, and to the vehicle powertrain. These phases certainly have a strong influence the way a hybrid or start&stop system works. This investigation has been done to verify if implementation of a specific model for low velocities is required in driver modeling. The sample data includes 3 types of vehicles, with the 4 drivers on 2 different environments. Only the behavior in the longitudinal direction is studied.

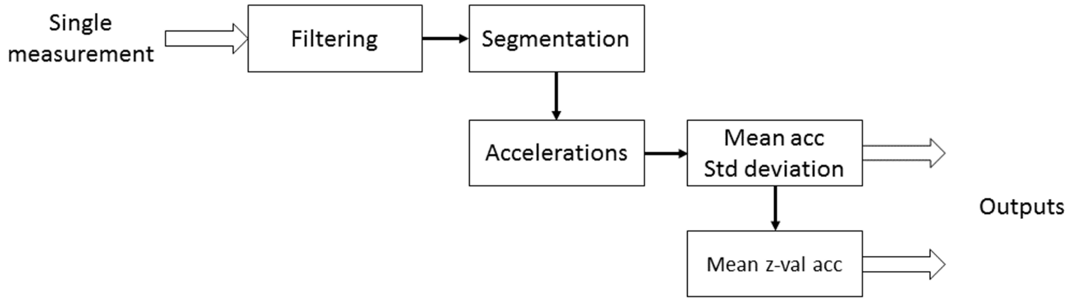


Figure 2.14: Data processing used to automatically extract low velocity driving statistics from the data recordings.

Each measurement recording has been filtered to remove sensor noise and segmented (see Figure 2.15) to represent acceleration and deceleration phases. Then statistical values could be extracted based on these acceleration segments. This process is shown in Figure 2.14.

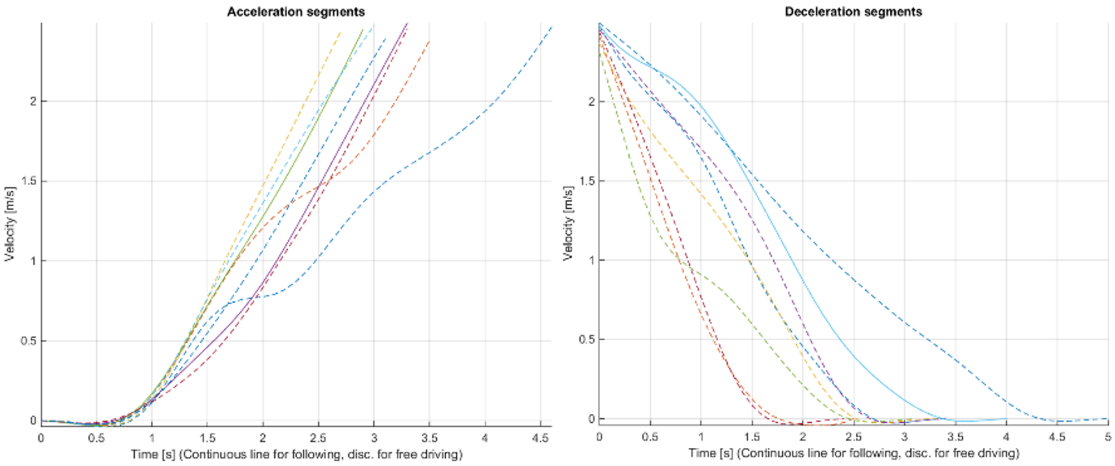


Figure 2.15: Example of driving segments automatically extracted from driving data.

The results of this analysis have been done based on the analysis of the average acceleration focused on several cases: users, vehicles and environments.
The values show in the following graphs are:

- Mean: mean of the average values of segments on a single experiment.
- Mean$-|Zsc| > 1$: mean of the average values of segments on a single experiment after removing the data segments whose mean value Z-score was over one.
- Mean/item: mean of the means for all the experiments cases used for the analysis of the item.
- Zmean/item: mean of the "Mean$-|Zsc| > 1$" for all the experiments cases used for the analysis of the item.

Removal of data with Z-score over one is done to avoid to consider data outliers segments. The values analyses from this experiment are the Zmean/user, hallowed in yellow. Then are supposed to be representative of the average driving behavior on the studied item.

- Analysis by driver

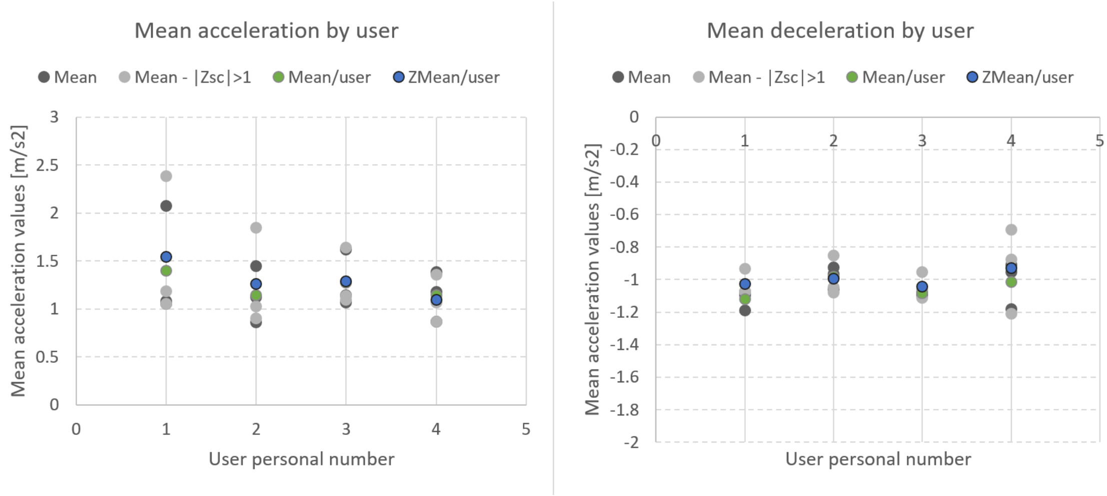


Figure 2.16: Mean acceleration and deceleration statistics per driver.

As observed in Figure 2.16, the driver type has a high influence on the average acceleration and deceleration. It is easily possible to identify aggressive and soft drivers.

- Analysis by environment

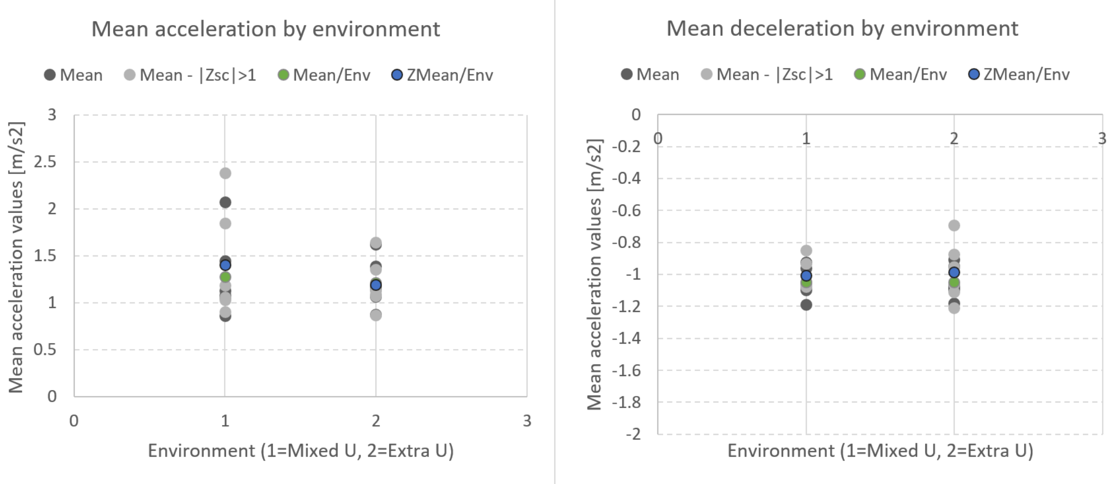


Figure 2.17: Mean acceleration and deceleration statistics per environment.

As observed in Figure 2.17, it is not clearly possible to observe an influence of the environment on the low velocity driving dynamics.

- Analysis by vehicle type

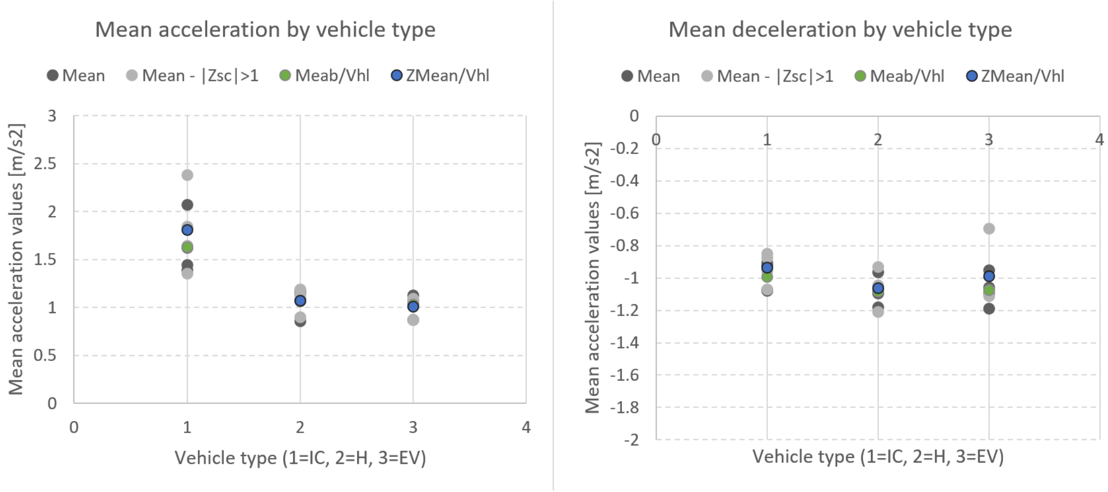


Figure 2.18: Mean acceleration and deceleration values per vehicle type.

As observed in Figure 2.18, only the IC vehicle acceleration phase shows specific values. This phenomenon has already been observed on the Influence of the vehicle analysis. This is due to the behavior of the automated gearbox. Otherwise, EV and H show similar driving dynamics.

2.3.2.3 Conclusion

From this experimental campaign, it can be remembered that:

- Different drivers show different accelerations and decelerations profiles, and high and low velocities.
- The type of powertrain in the driven vehicle does not influence much the driving style. Once the learning phase over, drivers drove with their own personal driving behavior, as long as the vehicle could provide the desired dynamics. The only observation of vehicle dynamics change due to the powertrain type is a low velocity, and was due to the behavior of automatic gearbox.
- The type of environment did not show any impact on the driving style.

2.3.3 Highway experiment

The goal of the highway experiment has been done to get good quality vehicle-following recording data with repeatable leading vehicle velocity patterns on a simple environment for model identification.

To do so, the following experimental setup has been selected:

- The selected environment has been a free highway section with few traffic flow and large possible velocities oscillations.
- The leading vehicle was a large van (for easy radar detection by the following vehicle) equipped with a target velocity profile display system detailed bellow. The selected leading vehicle velocity profile had large velocity and acceleration amplitudes to cover most driving dynamics and thus facilitate model parameters identification (see Figure 2.19).
- The following vehicle was the hybrid Toyota Prius detailed in section 2.3.2.1, equipped with a GPS position sensor, a vehicle dynamics sensor, and a leading vehicle relative distance radar sensor.

Five drivers have been recorded over two days with requested natural, soft and aggressive behaviors. Each recorded lasted about 10 minutes and has been repeated twice, for a total of 30 measurements. Nevertheless a few of these measurements had to be discarded due to the behavior of traffic flow vehicles.

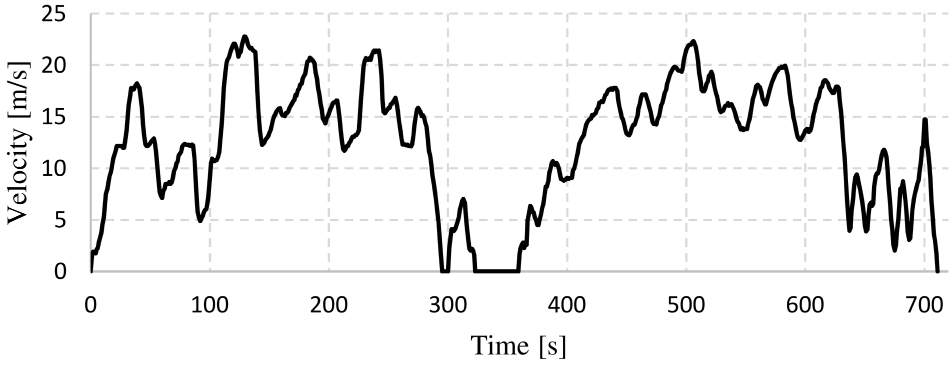


Figure 2.19: Leading velocity pattern used for the Highway experiment.

As explained above, an online target velocity display system has been created for this experiment. This tool enabled to have obtain a well designed leading vehicle velocity profile, to facilitate model identification. This system was composed of a program coded in Matlab using an USB GPS sensor to record and display a target velocity based on the GPS position and the curvilinear abscissa on the recorded path. This program was interfaced to the driver by a remote desktop system displayed on a smartphone on the dashboard of the leading vehicle.

This system is explained schematically in Figure 2.20, and by some images in Figure 2.21.

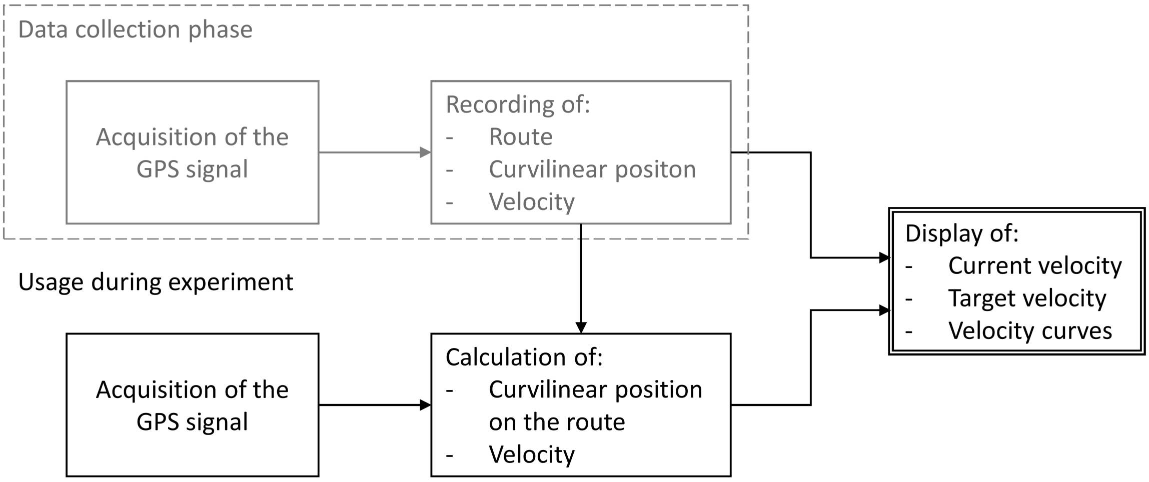


Figure 2.20: Functional diagram of the target velocity display program.

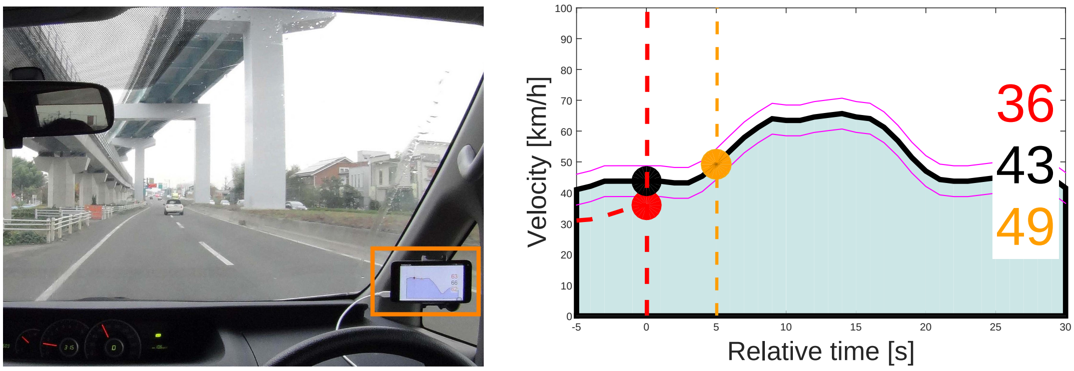


Figure 2.21: Example of target velocity display usage. On the left usage in a car, on the right displayed information. Current velocity in red, target velocity in black and future target velocity in orange.

The data recorded during this experiment has been used in the conference publication [11] and the journal article [12].

Now that the purpose of measurement data collection and the measurement campaigns have been described, the following section details the two multi-mode dynamical system models used in this research for driving behavior modeling.

2.4 Multi-mode dynamical system models

Multi-mode dynamical systems models are an extended class of dynamical systems models. Dynamical system models can be defined as functions describing the geometrical representation of a set of points in a finite geometrical space over time. The extension of this model class to multi-mode system models enable discrete or continuous switch between several dynamical system models. When the mode switching is discrete, multi-modes dynamical models are named hybrid dynamical models. This allows to represent both discrete- and continuous-in-time behavior of a dynamical system, and discrete- and continuous-switching between several models. Applications of multi-mode dynamical system models are broad. They can for example be used for the representation of physical phenomenons involving environment related physics (control of an air/water drone), a mixture of digital and analog electronic components, or robotic systems involving decisions taking mechanisms [74–76]. The family of hybrid dynamical system models includes Hybrid Automata [77], where the model system state is explicitly expressed by a continuous and a discrete signal, partition of the continuous state deciding of the current discrete state (hybrid system mode), and Switching Systems, systems composed of differential equations selected by a switching signal [78].

Depending on the studied point of view, different notations are used for multi-mode dynamical systems. Signal processing and robotics fields tend to define the multi-mode systems in its state-space form with observable and process functions, and to define the model configuration with states variables [75, 79]. In dynamical systems and system identification fields, multi-mode dynamical system models are usually defined by input-output functions systems, and the model configuration is defined by the system output and the model parameters [59, 80, 81].

In this thesis, the multi-mode dynamical models are studied in the context of system modeling from measurement data, thus involving a detailed parameter identification phase. Thus the multi-mode systems are described using their input-output functions system form.

2.4.1 Piecewise autoregressive exogenous model

An archetype hybrid (discrete multi-mode switching) dynamical system model is the PieceWise AutoRegressive eXogenous (PWARX) model. This model has been extensively used in data based system modeling due to its ability to reproduce the behavior of complex dynamical systems by using several simple ARX models. ARX models use past knowledge of the model output (autoregressive input) and external data input (exogenous input) to calculate the model output. Operations between the inputs and outputs are linear.

The general piecewise hybrid dynamical system model framework [80] is can be described as

$$\begin{cases} y(k) = f^1\left(r^1(k), \theta^1\right) + e^1(k) & \text{if} \quad \mu(k) = 1 \\ & \vdots \\ y(k) = f^M\left(r^M(k), \theta^M\right) + e^M(k) & \text{if} \quad \mu(k) = M \end{cases} \tag{2.1}$$

where r is the model input, y is the output data, f^m is a dynamical system model with $m \in \{1, 2, ..., M\}$ the mode index number, $k \in \{1, 2, ..., K\}$ is the discrete time step, $\theta^m \in \Theta^m$ is the parameter vector of the mode m, Θ^m the parameter space of the mode m, e^m is the modeling error, and $\mu \in \{1, 2, ..., M\}^K$ is the mode index vector.

Based on these notations, the PWARX model can be formulated as

$$\begin{aligned} &\hat{y}(k){=}f(r(k)), \\ &r(k){=}[\hat{y}(k-1)\ldots\hat{y}(k-n_a)\ u(k)\ldots u(k-n_b)]^\top, \\ &f(r(k)){=}\begin{cases} \theta^{1\top}\begin{bmatrix} r(k) \\ 1 \end{bmatrix} & \text{if} \quad r(k) \in X^1 \Leftrightarrow \mu(k) = 1 \\ & \vdots \\ \theta^{M\top}\begin{bmatrix} r(k) \\ 1 \end{bmatrix} & \text{if} \quad r(k) \in X^M \Leftrightarrow \mu(k) = M \end{cases} \end{aligned} \tag{2.2}$$

where r is the regression vector (input vector), $\hat{y}$ is the model output, and u is the exogenous input. $(n_a, n_b) \in \mathbb{N}^{*2}$ are the ARX models orders. M defines the number of mode. These last three variables are supposed to be known.

The data partitions X^m are assumed to be bounded convex polyhedra, described by

$$X^m = \{r \in \mathbb{R}^{n_{pw}} | H^m r \leq h^m\} \tag{2.3}$$

where H^m and h^m are the real valued matrix and vector describing the mode partitioning. $X = \cup_{m=1}^{M} X^m$ is assumed to be a bounded convex polyhedron, and

$\forall(i,j) \in \{1,2,...,M\}^2, i \neq j, H^i \cap H^j = \emptyset$.

PWARX models combine the simplicity of linear models identification and the ability to model dynamical systems, thus showing non-linear output properties. Using multiple ARX model, the PWARX model allows to increase the level of nonlinearity of the modeled phenomenon by using ARX models in a mathematical neighborhood of the model state-space. Numerous modes switching strategies have been developed, such as hierarchical segmentation [82], stochastic mode switching [60] or probilities weightening [59] such as explained in the following section. Moreover, this type of model attracted a large attention on the driver-modeling community, especially for driver tasks analysis and driving behavior understand [61, 83–85]. As a composition of dynamical linear models together with modes switching processes, numerous mode identification schemes can apply to PWARX models [86–90]. A comparative review of the identification methods can be found in the reference paper [81].

2.4.2 Probability weighted autoregressive exogenous model

The Probability weighted AutoRegressive eXogenous (PrARX) model is a multi-mode ARX model introducing a mechanism for continuous soft mode switching [59]. The mode switching mechanism is based on logistic regression and on probability estimation methods (see Figure 2.22). This soft mode transition methods avoids to have a binary decision in mode switching as in a PWARX model. Moreover the introduction of soft mode switching enables to get rid the discontinuity in the model output derivative at mode transition for low auto-regression order models. PrARX model parameters can be identified by calculating independently the ARX and the logistic function parameters, or with a method enabling simultaneous identification of both the model ARX model parameter and the logistic function parameters. Moreover, parameters of a PrARX model can be converted into PWARX model parameters.

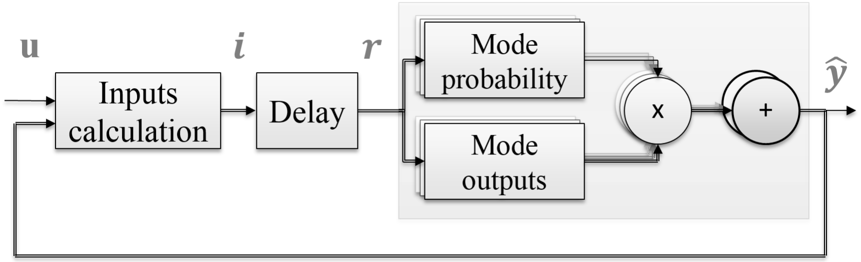


Figure 2.22: Diagram of the PrARX hybrid dynamical system model.

As expressed above, the main difference with the PWARX model is the mode switching mechanism. The PrARX model uses a logistic function to generate weights associated with the modes output. The model output is then calculated by summing of the ARX modes outputs multiplied by their mode probability.

The PrARX model is formally defined by:

$$\begin{aligned} \hat{y}(k) &= f_{PrARX}(r(k)) = \textstyle\sum_{m=1}^{M} P^m \theta^{m\top} \phi(k) \\ r(k) &= [\hat{y}(k-1) \dots \hat{y}(k-n_a)\ u(k) \dots u(k-n_b)]^\top \end{aligned} \tag{2.4}$$

where r defines the regression vector, $\phi(k) = [r(k)^\top 1]^\top$, u is the exogenous input, $\hat{y}(k)$ the model output, θ^m the identified vectors defining the ARX mode $m \in \{1, 2, ..., M\}$, $(n_a, n_b) \in \mathbb{N}^{*2}$ are the ARX models orders, and M defines the number of modes. P^m is the vector expressing the mode probability of the mode m output. P^m is defined by the following softmax function:

$$P^m = \frac{\exp(\eta^{m\top}\phi(k))}{\sum_{m=1}^{M} \exp(\eta^{m\top}\phi(k))} \tag{2.5}$$

where η^m is the identified parameter defining the probabilistic partition of the mode m.

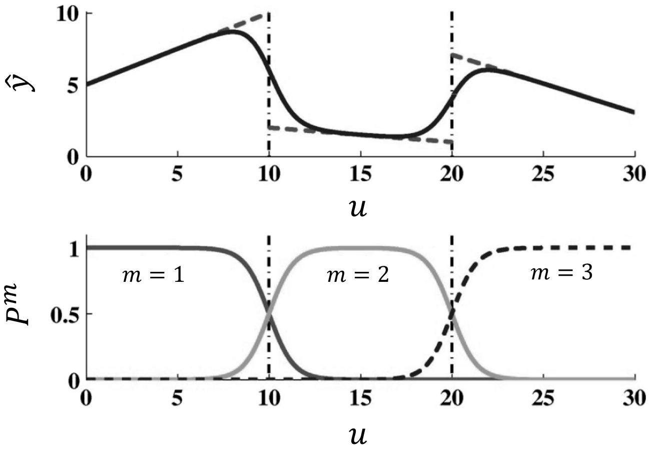


Figure 2.23: Example of single output three modes PrARX model.

The article by Okuda et al. presenting the PrARX model is focused on the analysis of personalized driving behavior. In this article, car-following driving behavior is identified from driving simulator data, classification of the recorded data situations, interpretation of the model parameters and synthetic data reproduction are proposed.

The following chapter proposes a framework for the creation of a personalized adaptive cruise control system, and an applicative comparison between the developed model and the classical cruise control models.

Chapter 3

Behavior personalized adaptive cruise control

3.1 Introduction

Adaptive cruise control (ACC) systems have been studied and implemented on cars in recent years, and they are part of the first steps to vehicle automation [36]. As an extension of classic cruise control, most of these systems have been designed to keep a minimum predefined time headway to the leading vehicle. This functionality can reduce fatigue of the driver, fuel consumption, and traffic flow congestion [91].

In the past years, ACC models and vehicle maneuver planning systems have been created taking into account the limited resources of fail-safe vehicles electronic architectures [92]. In these cases, the ACC assistance is based on a simple risk evaluation index such as time headway, and the user has to manually predefine the value. This type of system is designed considering the average behavior among users, and it is cannot be entirely personalized. Taking advantage of classic low computing power ACCs and behavior learning, researchers created adaptive cruise control systems able to calculate and use the correct time headway as a reference for the control program [91].

The view point taken in this paper is different. Recent evolutions of embedded systems enables us to implement a control method with more complexity and higher computational load. Moreover, due to the increase of passenger vehicles connectivity, heavy computational tasks can be performed in the cloud. Thus personalized parameter identified based dynamical systems can considered as a serious candidate for on-line vehicle control.

In recent years, driving behavior has been approached through multi-mode dynamical systems and investigated by numerous researches. The main idea is to observe

the human as a controller. It can be done by using linear or non-linear control theory [47], stochastic models, hybrid models, or implying neural networks or hidden Markov chains [62–64].
Based on this knowledge, this paper proposes an ACC system able to use system identification based following systems. Emphasis is done on the use of the Probability-Weighted ARX model [59]. This paper proposes a demonstration of the created model dynamics in typical driving situations, and a modeled vehicle dynamics comparison with typical car-following models and an industry grade ACC model.

Section 3.2 presents the concept of the novel ACC system and its basic operating. Section 3.3 details architecture of the model. Sections 3.3.1 and 3.3.2 explain in detail the used vehicle following models: the multi-mode PrARX model, and the GHR car following model. Section 3.4 demonstrates the use of the created ACC structure. Section 3.5 compares the results with classic ACC and traffic flow model, and section 3.6 proposes a possible model use.

3.2 The virtual leading vehicle adaptive cruise control concept

In this first section, the concept, the macroscopic architecture, and the main working situations of the presented adaptive cruise control are exposed.

3.2.1 Model concept

The concept behind this model is to create a simple structure able transform multi-mode vehicle following models in ACC models. The car following model used in this study is the PrARX model [59], which is a multiple ARX model with soft mode transitions abilities. This vehicle following model, once correctly calibrated, can reproduce accurately the driver dynamics, following distance and response time. Nevertheless this ARX model needs to follow a lead vehicle at any time to be able to calculate an output, and this can be done thanks to the here exposed Virtual Leading Vehicle ACC system (Vlv-ACC).

As smart cruise control systems, ACCs have to be able to deal with two main situations. Cruising at constant velocity when there is no leading vehicle in the way, or follow a leading vehicle with the right safety distance. Moreover an ACC must be able to handle correctly transition phases.

To be able to tackle these issues, the concept of virtual leading vehicle (VLV) was created. The virtual leading vehicle replaces the real leading vehicle (RLV) as soon as the real leading vehicle following conditions are not satisfied (see Figure 3.1). The VLV is driven at the cruise control desired velocity, and a set of VLV/RLV switching rules enable to change the leading vehicle smoothly. Thus the following vehicle (FV) driven by a standard car following model gets the ACC functionality.

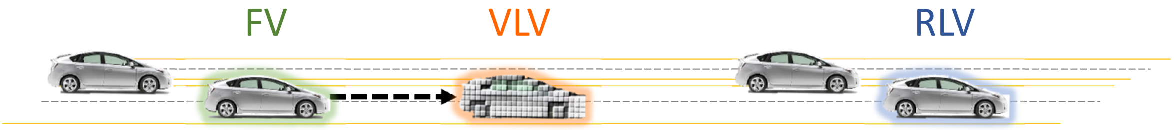


Figure 3.1: Two lanes highway driving. The FV is following the VLV.

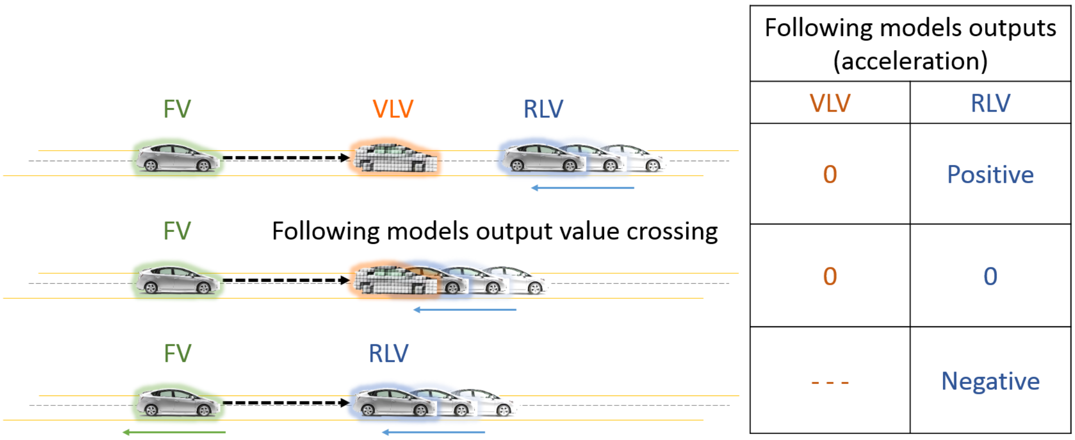
The Vlv-ACC uses the selected following model to follow both the VLV and RLV at any time. Then based on the vehicles relative positions and on the VLV and RLV following models outputs, the switching mechanism can select which target vehicle the FV has to follow. The following rules used to switch between situations only works with sufficiently sophisticated following models. Indeed the selected following model must have a natural reaction to the distance and relative velocity between vehicles.

3.2.2 Model situations definition

In this configuration, two main situations and four switching cases can occur. The two main situations have already been introduced: following the virtual leading vehicle (VLV) or following the real leading vehicle (RLV). The VLV following situation is the basic state when the ACC is activated.

Then the four possible switching situations are the following:

3.2.2.1 Soft vehicle cut-in



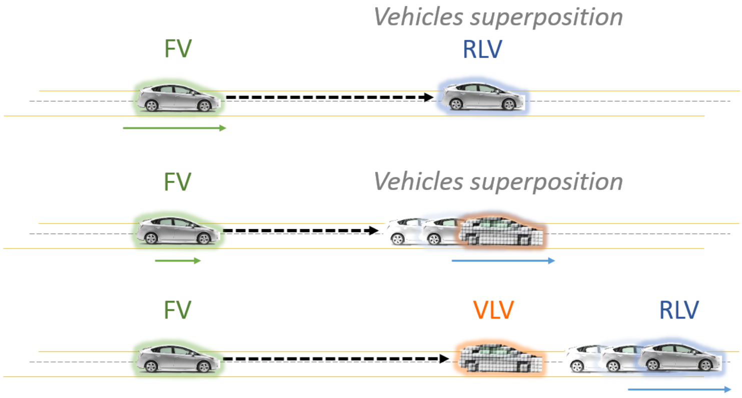
Following models outputs (acceleration)	
VLV	RLV
0	Positive
0	0
- - -	Negative

Figure 3.2: "Soft RLV cut-in" situation. Initially, the FV is following the VLV. A RLV is slowing down, and is smoothly replacing the VLV, when the models outputs (FV acceleration) values intersect.

The "Soft RLV cut-in" switching case (see Figure 3.2) modifies the situation from following the VLV to following the RLV.

Soft RLV cut-in expresses the fact that while following the VLV, a slower than VLV and far away RLV vehicle is in the current lane. After a certain time, this RLV will overlap the VLV and eventually collide the FV. To avoid this situation, both following models output for the RLV and VLV are constantly computed, and when these models outputs (in our case vehicle acceleration) intersect, the transition from following VLV to RLV is done.

3.2.2.2 Soft vehicle out



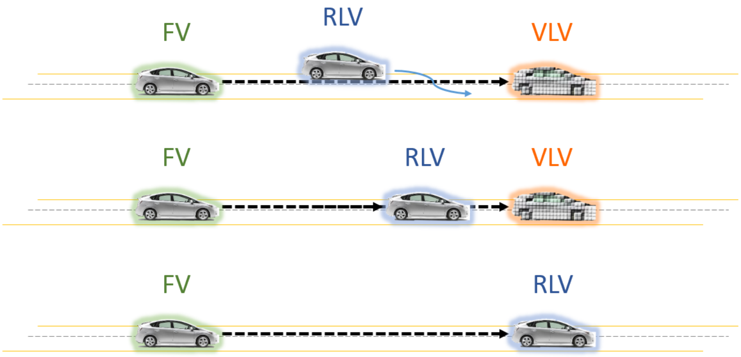
Leading vehicle velocity	
VLV	RLV
Same as RLV	Lower than ACC defined
ACC defined velocity	ACC defined velocity
ACC defined velocity	Higher than ACC defined

Figure 3.3: "Soft RLV out" situation. Initially, the FV is following a RLV. The RLV accelerates. When RLV velocity is higher that the cruise control desired velocity, the RLV is replaced by the VLV.

The "Soft RLV out" switching case (see Figure 3.3) modifies the situation from following the RLV to following the VLV.

Soft RLV out expresses the fact that while following a RLV, the RLV accelerates to a velocity higher than the ACC desired velocity. When the RLV reaches the cruise control desired velocity, the RLV is replaced by an overlapping VLV. Another switching solution is to virtually superpose the RLV and VLV, but force the VLV velocity to the ACC velocity for the following model output calculation. Then the transition can be triggered using the same technique as Soft RLV cut-in, with following models output values crossing.

3.2.2.3 Hard vehicle cut-in



Following models outputs			
T	VLV	RLV	FV
$T < T_0$	a_x	- - -	a_x
$T_0 < T < T_0 + \tau$	$a_x(T_0)$	a_y	$t * a_x(T_0) + (t - \tau) * a_y$
$T_0 + \tau < T$	- - -	a_y	a_y

Figure 3.4: Example of a possible "Hard RLV cut-in" situation. Initially, the FV followed the VLV. A RLV inserts on the VL lane with a high negative differential velocity. a_y has to be lower than a_x to trigger the "Hard RLV cut-in" situation.

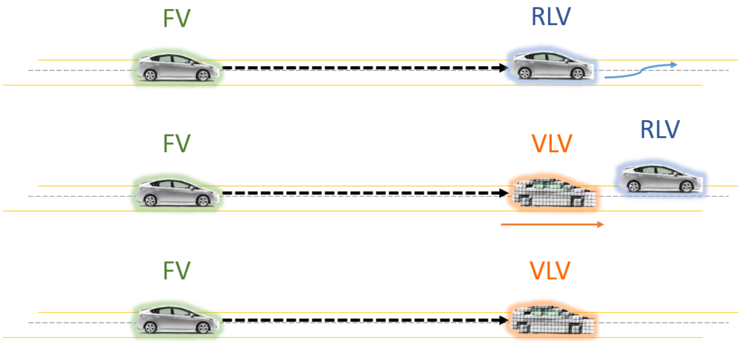
The "Hard RLV cut-in" switching case (see Figure 3.4) modifies the situation from following the VLV to following the RLV.

"Hard RLV cut-in" expresses the fact that a RLV inserts between the FV and the VLV, and its velocity is low enough so the FV driver would need to brake. In this case, output of the following models are switched in a time corresponding to a fraction of the initial time headway, avoiding any safety related issue as well as acceleration discontinuity. The output switch occurs only if the replacing following model output value is lower than the previous one.

3.2.2.4 Hard vehicle out

The "Hard RLV out" switching case (see Figure 3.5) modifies the situation from following the RLV to following the VLV.

"Hard RLV out" expresses the fact that while following the RLV, this vehicle quickly exits from the current lane (lane change, highway exit). The RLV is replaced by the VLV. The VLV velocity is increased from RLV velocity to the cruise control desired velocity with an acceleration corresponding to the average observed driver acceleration during the learning phase.



Vehicles accelerations		
Conditions	VLV	RLV
- - -	- - -	a_{RLV}
$V_{VLV} < V_{ACC}$	$a_{defined}$	- - -
$V_{VLV} = V_{ACC}$	0	- - -

Figure 3.5: Example of 'Hard RLV out' situation. Initially, the FV is following the RLV. When the RLV exits the lane, it is replaced by a VLV at the previously known RLV position and velocity.

The proposed switching conditions are only one possible interpretation of the model. These conditions have shown good results with the hybrid dynamical PrARX following models [59], but can be too broad for some basic car following models.

3.3 Implementation of the proposed adaptive cruise control model

This section is dedicated to the explanation of the model structure from the point of view of implementation. Simulink has been used to build the model. It enables easy replacement of the vehicle following model, and a connection can be done with vehicle simulation softwares.

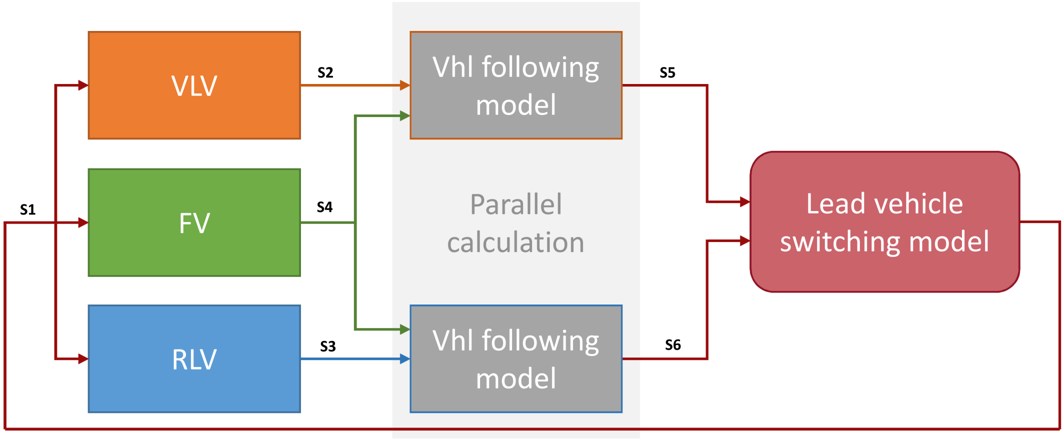


Figure 3.6: Diagram of the adaptive cruise control model architecture.

As explained in section 3.2, the Vlv-ACC runs two instances of the selected vehicle following model in parallel. This is required to be able to have smooth mode switching in case of soft RLV in. This transition condition is based on both following models output comparison. In figure 3.6, the upper and lower vehicle following branch use the same following model. Only the inputs are different. S1 represents the leading and following vehicles locations and dynamics signals, and a virtual leading vehicle reset signal, S2 represents the virtual vehicle location and dynamics signals, S3 the real leading vehicle location and dynamics signals, S4 the following vehicle location and dynamics signals, S5 the virtual leading vehicle and following vehicle positions and dynamics signals at the next time step, and S6 the real leading vehicle and following vehicle positions and dynamics signals at the next time step. Details of the signals are proposed in Figure 3.7.

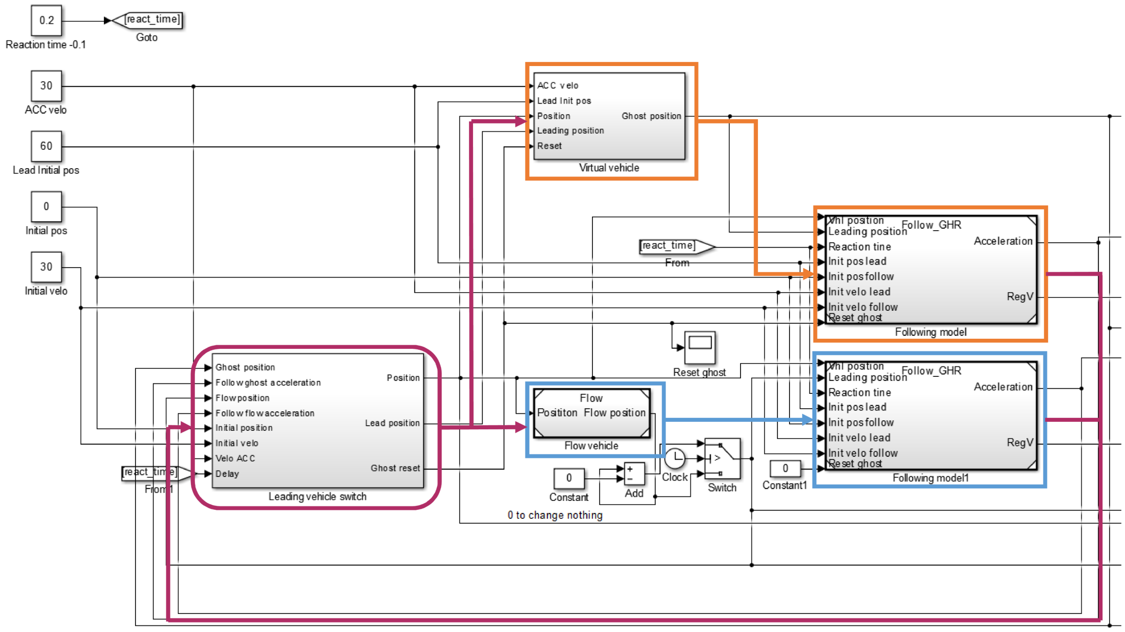


Figure 3.7: Example of Simulink integration of the model.

The upper branch of Figure 3.7 is dedicated to the VLV and to the vehicle following the VLV. The "Virtual vehicle" bloc is used to the generation of the virtual leading vehicle. This bloc inputs are composed of signals required to generate the VLV and an initial position signal. The signals to generate the VLV are the adaptive cruise control velocity, the position of the leading vehicle at the previous time step, the position of the following vehicle at the previous time step and a "reset" signal. If the "reset" signal is activated, the output signal is equal to the "Lead position" signal. If the "reset" signal is disabled, the output signal is equal to the "Lead position" signal added to the distance traveled by the VLV vehicle in one time step at the ACC velocity. The initial virtual vehicle position is set based on data correlation between the leading and following relative position at defined velocities and relative accelerations.

The lower branch of Figure 3.7 is dedicated to the RLV. In a real world situation, the RLV information would be extracted from on vehicle sensors such as a Radar, Lidar or camera system, and the flow vehicle information could be communicated to the following model (see Section VIII.b). However, in the case of this simulation, the RLV position is read from a vehicle position file. IPG Carmaker with on vehicle sensors and traffic flow would also be a solution.

3.3.1 Behavior personalized modeling by probability weighted ARX model

The multi-mode dynamical system model used to demonstrate the ability of the ACC system is the Probability Weighted autoregressive exogenous (PrARX) model 2.4.2. This model is sued to reproduce the driver-vehicle entity dynamics.

Reproduction of the driving behavior depends not only on the structure of the model (here PrARX), but also on the selection of explanatory variables of the model. The selected input vector is composed of acceleration of the driven vehicle, distance between the driver and leading vehicles (range), relative velocity between the vehicles (range-rate), and the inverse of the time headway (velocity/range) [93]. This set of inputs enables to give vehicles dynamic information and surrounding vehicles relation characteristics.

The output of the model is the desired acceleration of the vehicle.

A delay of 300ms is applied between the input and the output of the model, to insure a cognition time matching the average human brain capacity [16]. The driven parameters identification process is done by optimizing independently the ARX modes parameters based on pre-clustered data. Supervised clustering of the learning data enables robustness in the parameter identification, and the mode separation parameters are calculated uses multinomial logistic regression. The data used for parameter identification has been measured in the driving simulator (see section 2.2).

The selected mode data clustering is:
- A mode for the 20% most negative accelerations.
- A mode for the 15% highest accelerations.
- The last mode for the rest of the data.

This combination provided a stable and safe reproduction of the vehicle/driver dynamics.

3.3.2 Gazis-Herman-Rothery car following model

The Vlv-ACC is a flexible platform able to receive different types of car following models. While some very basic following models can need some adjustments on the ACC switching conditions, most of the classic car-following models work without any problem. To show the flexibility of the Vlv-ACC, the Gazis-Herman-Rothery (GHR) following model [3] was implemented. This model is a classic car-following model for microscopic traffic flow modeling.

The Gazis-Herman-Rothery (GHR) car-following model is one of the most well-known models and was developed in the early sixties at the General Motor Research labs in Detroit. It is based on the assumption that the drivers acceleration is proportional to the velocity difference and to the following distance. The model is expressed in the simple following formula (3.1).

$$a\_ghr_n(t) = cv_n^m(t)\frac{\Delta v(t-T)}{\Delta x^l(t-T)} \tag{3.1}$$

a_ghr_n represents the n^{th} vehicle acceleration, v_n the n^{th} vehicle velocity, Δv the differential velocity between the vehicle n and the vehicle $n-1$, and Δx the distance between the vehicle n and the vehicle $n-1$. C, m and l are the driver parameters. Due to the variety of models in this thesis, these parameters notations is kept from the original paper [3], and their meaning only has value in this section.

Then the following correction formula is applied to the GHR model:

$$\begin{aligned} a_n(t) = a\_ghr_n(t) + & \left(\tfrac{\Delta x(t-T) - d_{follow} - q}{r}\right) * \tanh\left(\left(\tfrac{\Delta x(t-T) - d_{follow}}{p}\right)^2\right) * \\ & \left(1 - \left(\tanh\left(\tfrac{\Delta v(t-T)}{s}\right)\right)^4\right) \end{aligned} \tag{3.2}$$

where d_{follow} is the desired average following distance, and p half of the dead-zone size on the relative distance axis, q the offset of the correction curve on the relative distance axis, r the proportional coefficient value on the relative distance, and s the limiting coefficient based on the relative vehicles velocity.

The first term of the correction formula (3.2) enables to keep a correct relative distance between the vehicles by modifying the acceleration value. The second term represents the zone of action based on the relative distance, and the third term represents the zone of action based on the relative velocity. The last two terms enable to avoid conflict between the GHR car following model and the relative distance correction during high dynamical phases.

As this model was developed in the sixties, it has been used in numerous studies and several parameter sets have been determined [3] to represent global driver behavior. In order to improve accuracy, we a set of parameters determined by Ozaki in 1993 was used. This set includes different parameters for acceleration and deceleration situations. The additional term in (3.2) has been added to the equation to force the following model to stay within a certain range of distance behind the leading vehicle. Indeed, the original GHR model cannot correctly handle the leading vehicle relative distance discontinuity. This term is only active during low dynamics phases, if the differential velocity between the vehicles is low. Thus it does not interfere with the GHR model during acceleration and braking phases (see Figure 3.8). The selected distance is 60m, and is a coherent value regarding the driving data collected in section 3.3.1. Selected parameters of the model can be observed in table 3.1.

Table 3.1: GHR model values, from Ozaki (1993) model calibration [3]

Variable	c	m	l	d_{follow}	p	q	r	s
Value if $a_n \leq 0$	1.1	0.9	1	60	10	10	50	1
Value if $a_n > 0$	1.1	-0.2	1	60	10	10	50	1

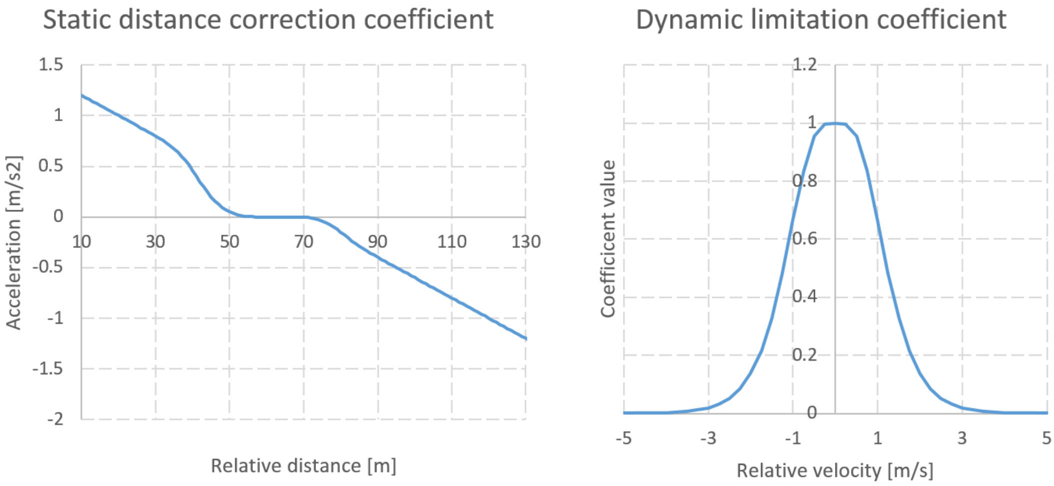


Figure 3.8: Corrective coefficients to the GHR model.

3.4 Comparison of models in the Vlv-ACC framework

This section is dedicated to the comparison between the PrARX model and the GHR model, both implemented in the Vlv-ACC framework. Three main driving situations are proposed. The following driving situations are representing classic use, with soft switching cases on Figure 3.9, hard vehicle in Figure 3.10, and hard vehicle out situation in Figure 3.11.

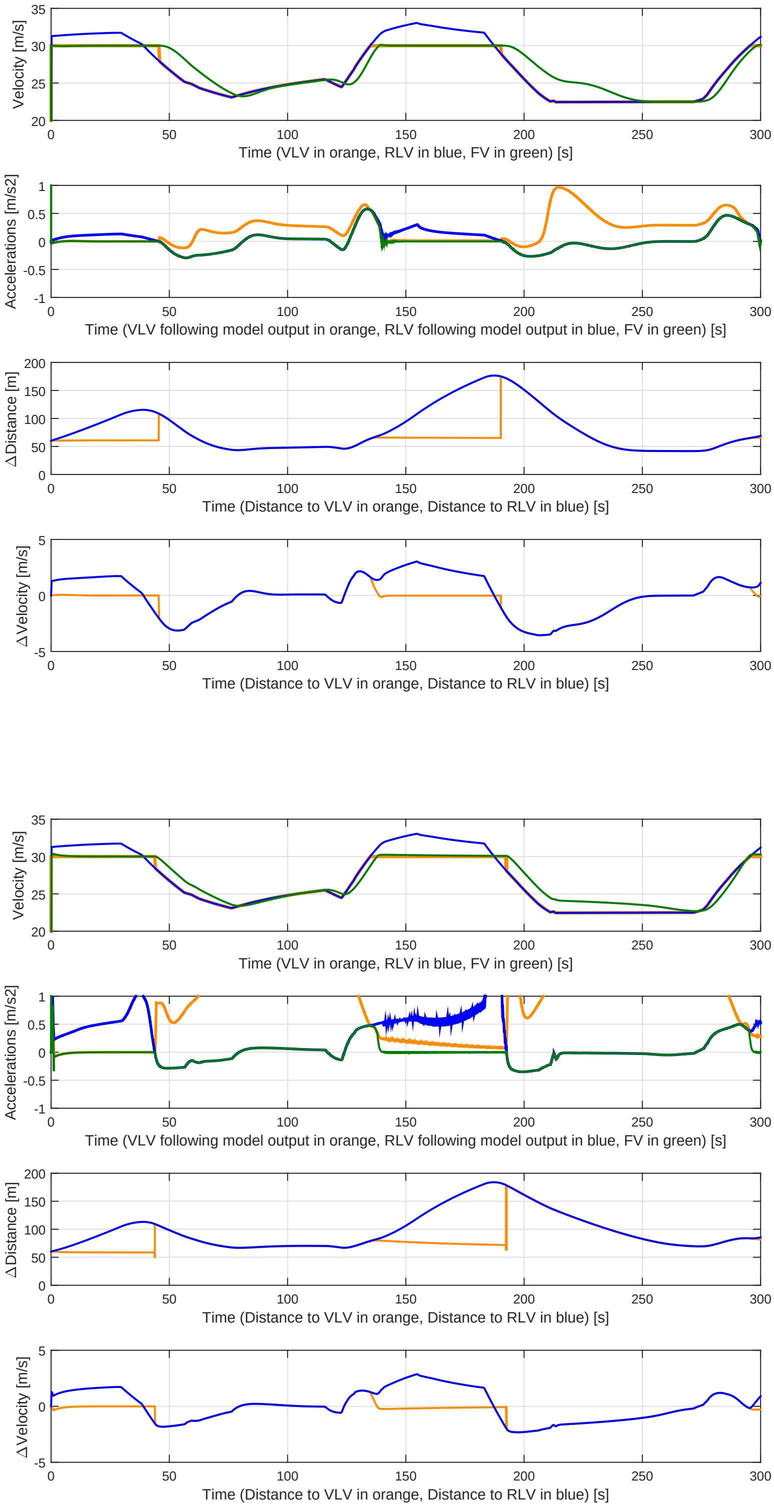


Figure 3.9: 30 m/s Vlv-ACC example with PrARX car following model on the top, with modified GHR car following model on the bottom.

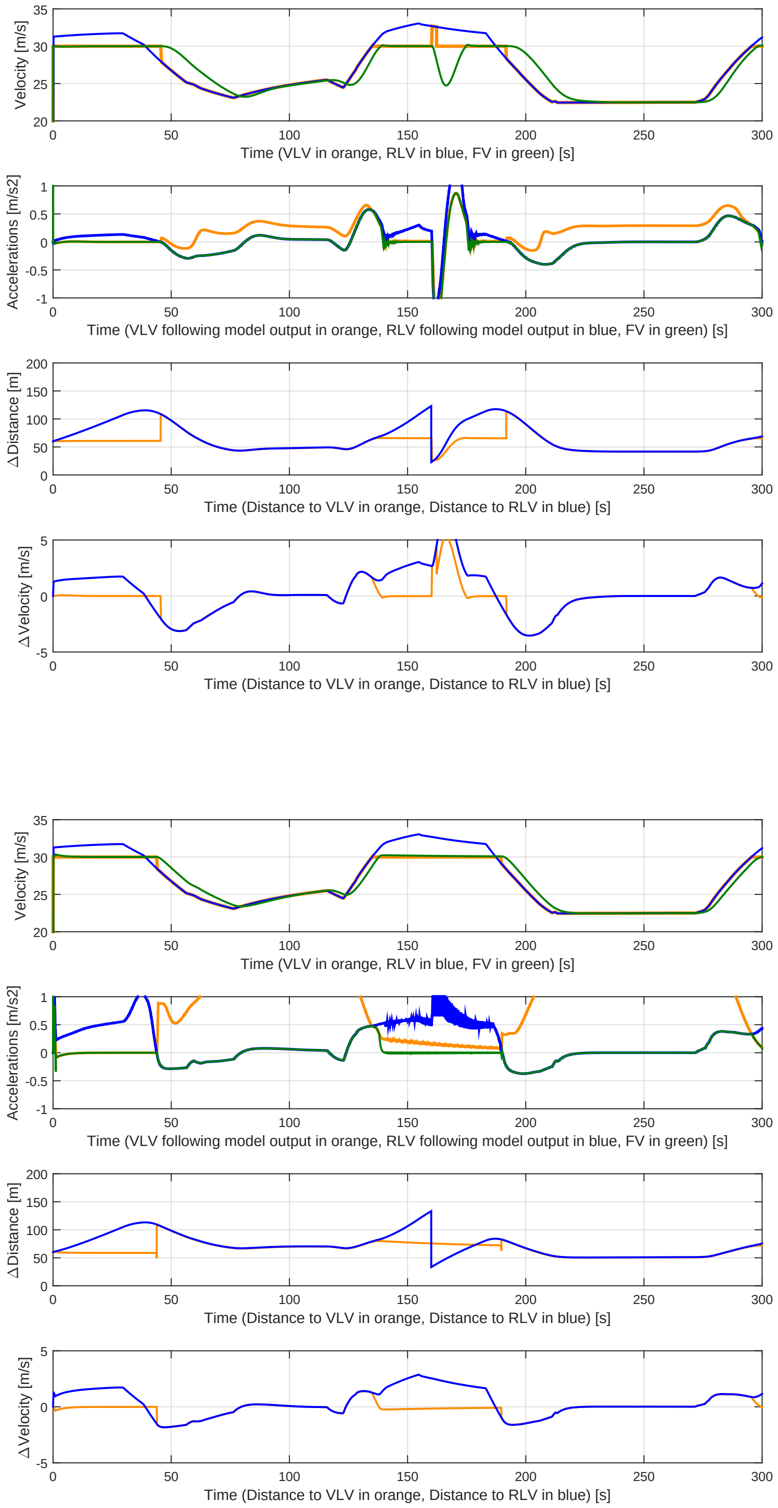


Figure 3.10: 30m/s Vlv-ACC with PrARX following model on the top, with modified GHR model on the bottom. 'Hard RLV cut-in' event at $t = 160$s.

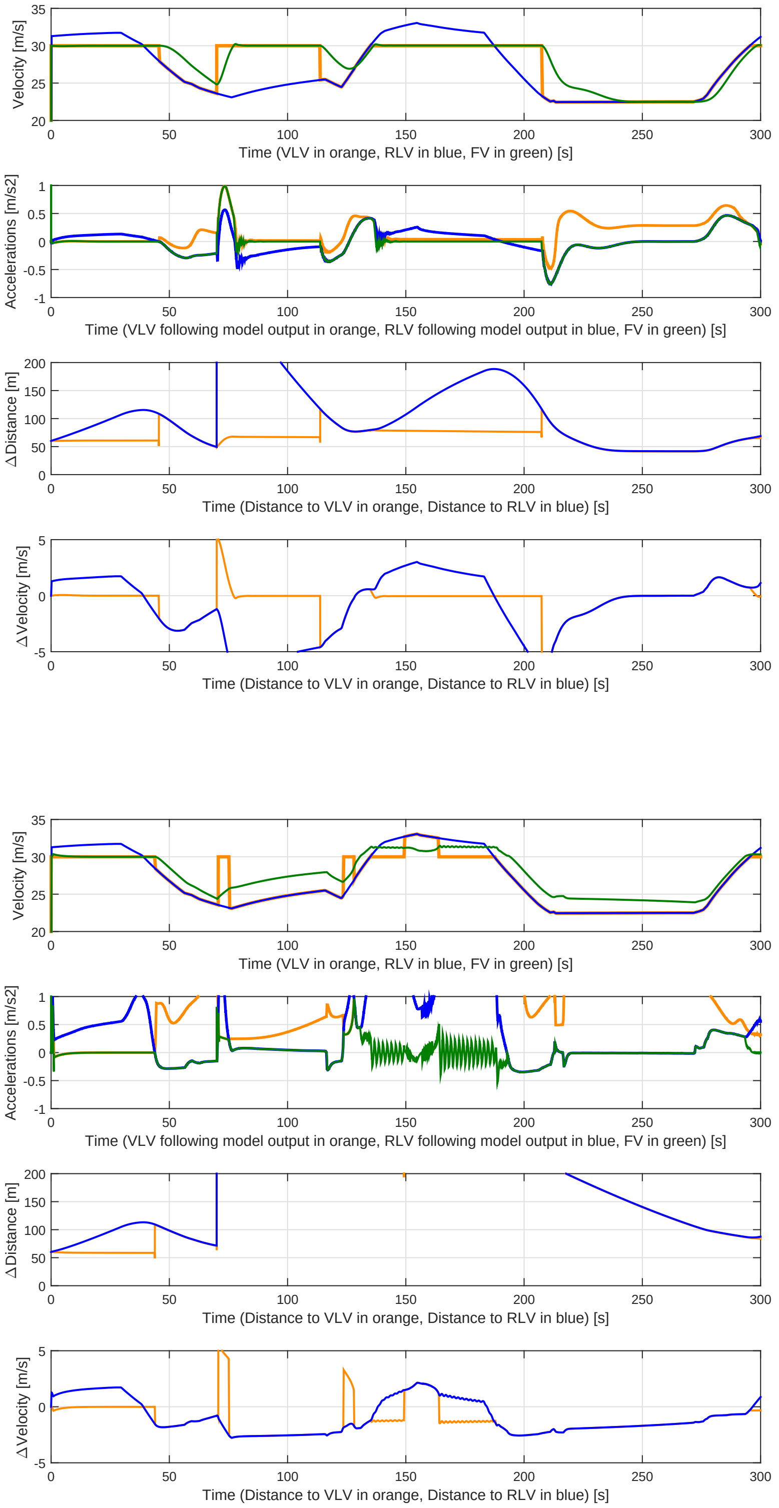


Figure 3.11: 30 m/s Vlv-ACC with PrARX following model on the top, with modified GHR model on the bottom. 'Hard RLV out' event at t=70s.

We can see on Figure 3.9 that both following models are able to handle soft switching conditions without any issue. For $t = [0, 40]$ and $t = [130, 190]$ seconds, ACC model is following the virtual vehicle, otherwise, it is following the real leading vehicle. The lead following vehicle can be understood by looking at the relative distance graph. The model response in green on the acceleration curve differs slightly but the global behavior of the vehicles are similar. There is no major discontinuity due to the lead vehicle switch, the obtained results dynamics can be considered as satisfying. The noise observed on the GHR results are due an instability in the GHR real vehicle following model output when the RV is following the VLV. This is due to a feedback delay issue in the Simulink implementation and does not alter the results of the study.

Figure 3.10 shows the response of the models to the insertion of a RLV during a VLV following phase (cut-in), at $t = 160$s. We can see that the PrARX based Vlv-ACC responds correctly, by decelerating the vehicle to keep a correct safety distance. In the case of the GHR based Vlv-ACC, we can see that no measure is taken by the model to insure a correct safety distance. It shows that the RLV insertion has not been detected, and the Hard RLV in case has not been triggered. This is due to the conception of the GHR model, based on the differential velocity divided by the differential distance. This model cannot handle lead car position discontinuities to adapt the following vehicle behavior. The GHR model is not a good candidate with the selected state switching rules.

Figure 3.11 shows the exit of a RLV at t=70s. This case can represent the departure of the leading vehicle from the highway. The PrARX based model react in a logical manner, by following the VLV. However the GHM based model does not handle the situation correctly. This is due to the fact than despite the distance separating the following car and the RLV, the model output is lower for the RLV following than the VLV following. Thus the models switches back to RLV following mode and the result becomes meaningless.

We could observe in the previous comparison that the logic switching has to be adapted to the selected following model. The switching logic has been developed based on human reactions. The PrARX car following model can be correctly reproduce human reactions, and therefore lead to a coherent model behavior. Nevertheless, the GHR model is not adapted to this iteration of the Vlv-ACC. Adaptation to the cruise control logic would have to be done to insure correct situation selection.

3.5 Comparison between a Vlv-ACC model and standard cruise control models

In this section, the comparison between the PrARX based Vlv-ACC and two classic ACC models is done. The goal is to show the main following characteristics of the PrARX model next to industry standard models.

3.5.1 Comparison with Gipps model

The Gipps model, presented in section 1.2.2.1, is a classic collision avoidance model type [3, 72]. It means that this type of model seeks to find a safe following distance to the lead vehicle.

$$v_n(t + t_{reac}) = \min\{v_a, v_b\}, \text{ with}$$

$$\begin{cases} v_a = v_n(t) + 2.5a_{(n)max}t_{reac}(1 - \frac{v_n(t)}{v_{n0}})\sqrt{0.025 + \frac{v_n(t)}{v_{n0}}} \\ v_b = b_n t_{reac} + \sqrt{b_n^2 t_{reac}^2 + b_n \left[2(s_{n0} - \Delta x(t)) + v_n(t)t_{reac} + \frac{v_{n-1}(t)^2}{b_{(n-1)max}}\right]} \end{cases} \tag{3.3}$$

where $v_n(t)$ is the velocity of vehicle number n, $\Delta x(t)$ the relative distance between vehicles, and the other parameters as in Table 3.2. Gipps model parameters notations have meaning only in this section.

Table 3.2: Gipps model parameters.

Desc.	Desired velocity [m/s]	Stopping distance to lead vehicle [m]	Reaction time [s]	Maximal acceleration [m/s2]	Desired deceleration [m/s2]	Maximal deceleration [m/s]
Variable	v_{n0}	s_{n0}	t_{reac}	$a_{(n)max}$	b_n	$b_{(n-1)max}$
Value	30	5	0.3	1.5	-1.5	-2.5

We can observe in Figure 3.12 that the Gipps model represents correctly the driver behavior. The response time, the following distance and the acceleration are close to the PrARX Vlv-ACC. Nevertheless the model output have the disadvantage to be discontinuous. Thus filtering should be applied, implying an undesired increase in the model response time.

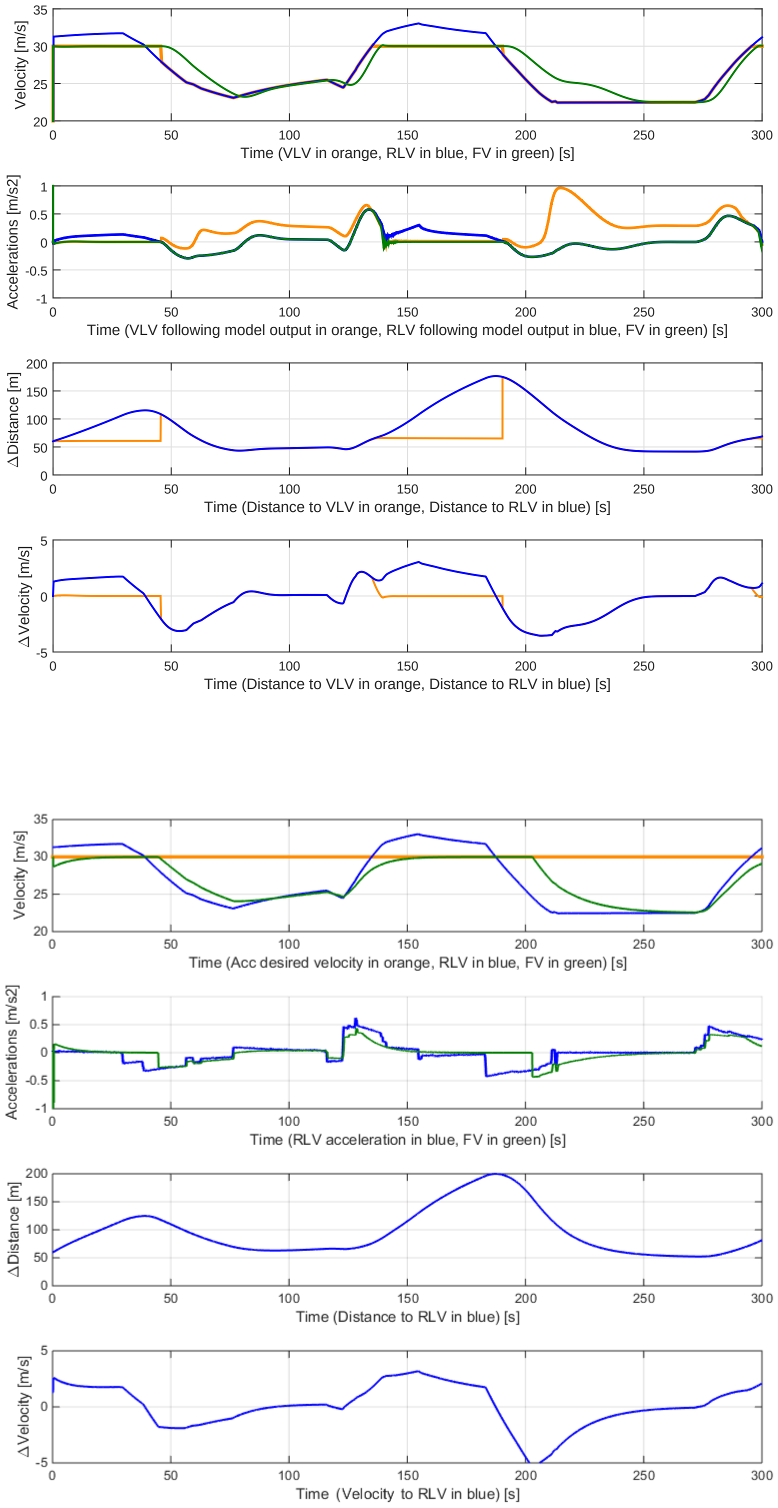


Figure 3.12: Following situation with the PrARX Vlv-ACC on the top, and with the Gipps model on the bottom.

3.5.2 Comparison with IDM model

IDM is a time-continuous car-following model, developed to improve older models as Gipps model [94, 95]. It has been used as base for the implementation of an ACC in a Volkswagen vehicle in the INVENT project.

$$\begin{aligned} \dot{v}(s, v, \Delta v) &= a \left[1 - \left(\tfrac{v}{v_0} \right)^4 - \left(\tfrac{s^*(v, \Delta v)}{d} \right)^2 \right] \\ s^*(v, \Delta v) &= s_0 + v t_{reac} + \tfrac{\Delta v}{2\sqrt{ab}} \end{aligned} \tag{3.4}$$

with v the vehicle velocity, Δv the relative velocity with the leading vehicle, s the relative distance with the leading vehicle. Parameters in Table 3.3 have been identified to match the PrARX model behavior. IDM model parameters notations have meaning only in this section.

Table 3.3: IDM model parameters.

Description	Desired velocity [m/s]	Jam distance [m]	Safety time [s]	Maximal acceleration [m/s2]	Desired deceleration [m/s2]
Variable	v_0	s_0	t_{reac}	a	b
Value	30	2	2	1.4	2

It can be observe on Figure 3.13 that despite the correct setting of the IDM parameters, the model has a very different behavior from the identified PrARX based Vlv-ACC. The vehicle accelerations are smoother but act later, focusing on the preservation of the correct safety distance. It can clearly be seen at [200-250] seconds, where the PrARX based model slows the vehicle down depending on the distance and relative velocity to the lead vehicle, while the IDM model only acts to reach the ideal safety distance. It is realistic to believe that the early braking behavior of the PrARX Vlv-ACC model enables the driver to understand the behavior of the ACC, and thus to have a good confidence in the system.

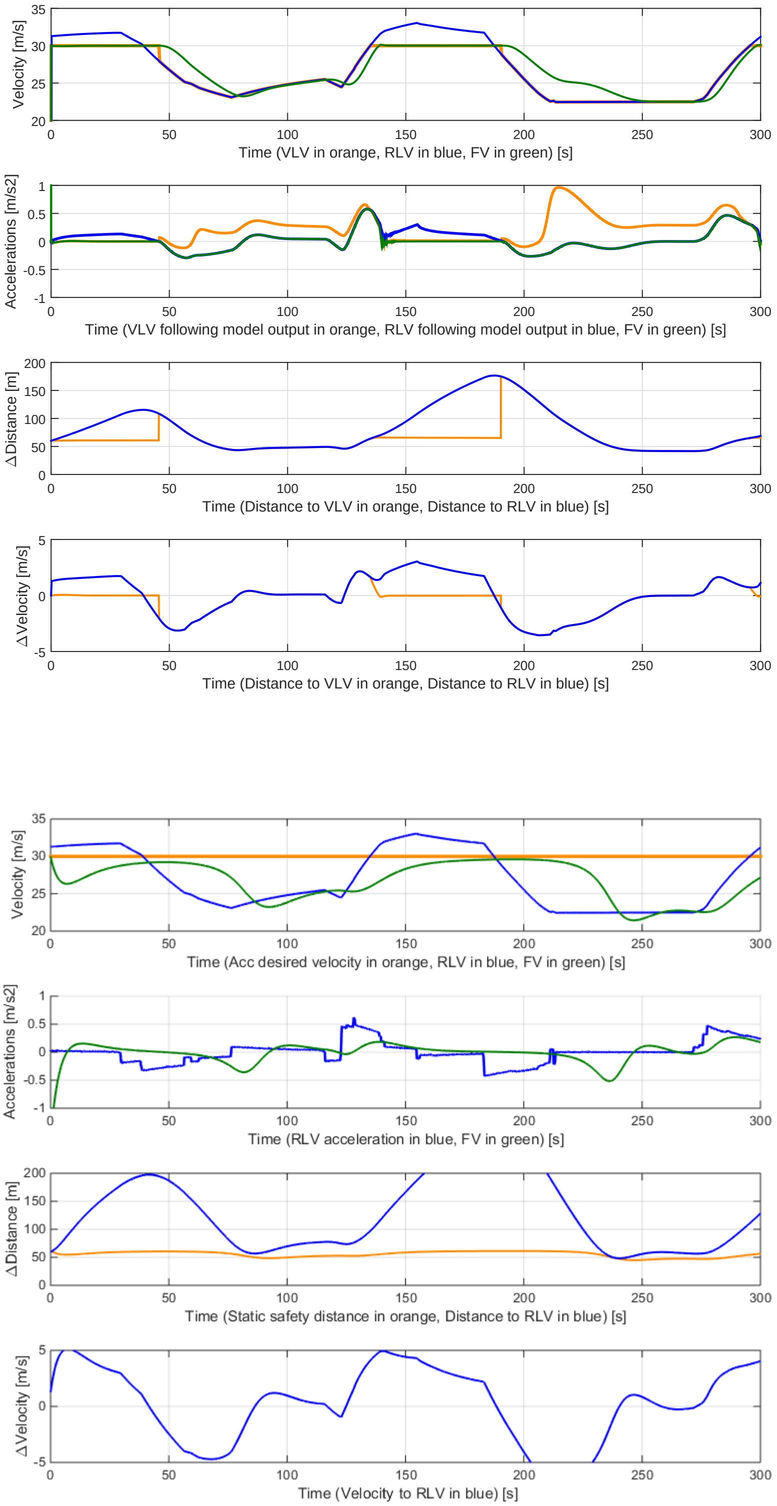


Figure 3.13: Following situation with the PrARX Vlv-ACC on the top, with the IDM algorithm on the bottom. Static safety distance represents s^* without the relative velocity term.

3.6 Real-world application

In this section, possible implementation in a real vehicle is exposed.

3.6.1 Computation complexity

To be able to use the PrARX Vlv-ACC, tree main steps have to be followed. At first the driving data has to be measured, then the driver has to be identified, and finally the PrARX model is able to reproduce the driver behavior. Both measurements and reproduction of the drivers behavior are very light processes in terms of computation power and can be executed in a low-power ECU of the car. The driver parameters identification algorithm is heavier, due to a process of optimization using a gradient descent algorithm. Parameter identification can be done on a remote server through network communication (see Figure 3.14). In case there is a need to adapt parameters definition on-line, a low-power version of the algorithm has also been developed [96].

Each iteration of following behavior estimation takes about ten micro-seconds, while the full identification operation can take a few minutes on a modern computer. If the algorithm runs at 10Hz, we see that there should not be any issue with current vehicle computation speed capacity.

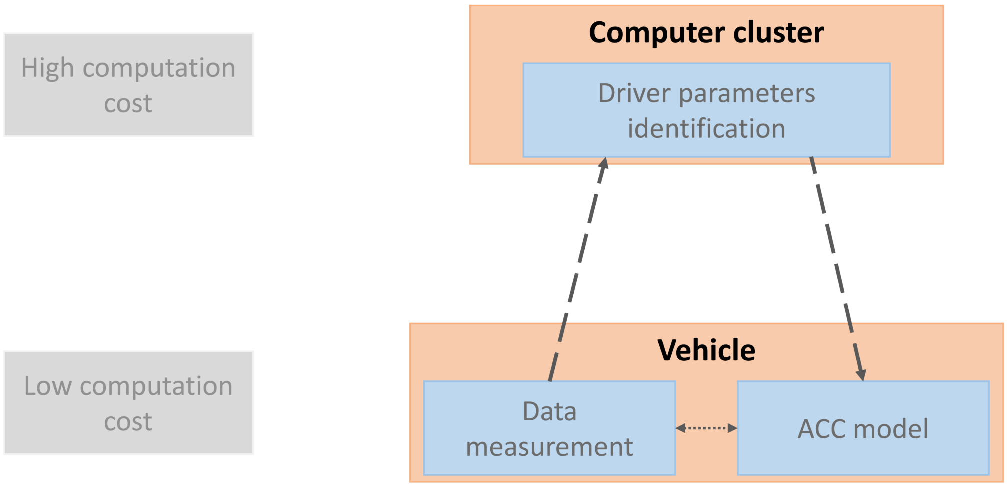


Figure 3.14: Proposition of vehicle implementation for PrARX models parameters identification.

3.6.2 Implementation of the developed cruise control model

Various driver models can be used in the Vlv-ACC architecture. ARX models and other autoregressive systems are sensible to information delay. This delay can act as a gain in the closed-loop system or can lead to instability. Moreover, the selected multi-mode dynamical system model, PrARX, reproduces the combination of the driver and vehicle behavior. Thus, the driven vehicle should not be in the control loop.

To avoid any of this situation from happening, separating the model in two distinct control loops is advised. As shown on Figure 3.15, the real vehicle is controlled based on a simulated vehicle. When using the PrARX Vlv-ACC model, the model calculated vehicle is simply the integration for the Vlv-ACC output, with real vehicle dynamical limits (i.e. max accelerations). Thus it can be used to frame the model dynamics, but does not cost any calculation time. The real vehicle is then controlled from the modeled vehicles position. The required relative distance to the leading vehicle can be corrected to represent the distance between the modeled vehicle and the lead vehicle.

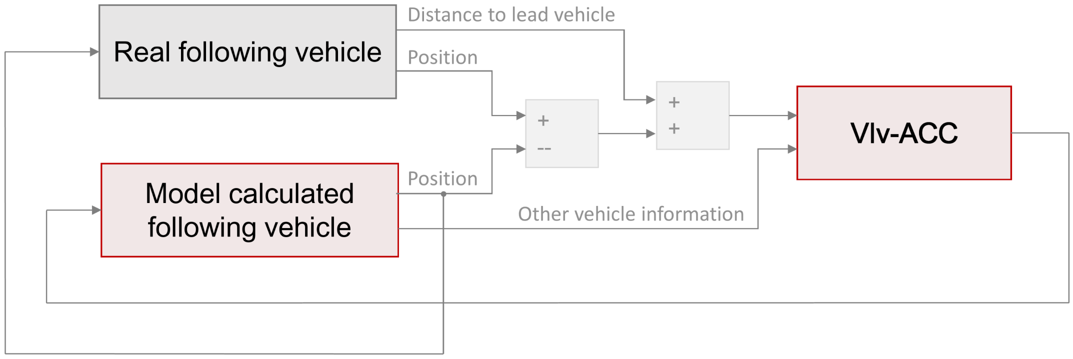


Figure 3.15: Proposition of in-vehicle Vlv-ACC implementation.

The following chapter proposes a novel approach to evaluate driver personalized energy consumption. The method consists of identifying driver-vehicle dynamics using a multi-mode dynamical model and then of reproducing driver-vehicle behavior in vehicle-following task. The energy consumption of the vehicle is estimated from the velocity profile calculated by using the driver-vehicle model. The explanation about the modeling framework and the obtained results are followed by application examples.

Chapter 4

Evaluation of behavior personalized vehicle energy consumption

4.1 Introduction

Analysis and evaluation of mobility solutions energy consumption is one of the key issues to realize environment-aware transportations. Numerous studies have been dedicated to energy consumption analysis of road and rail vehicles, at microscopic and macroscopic scales, in order to reduce energy losses [54, 72, 97–99]. Based on these energy estimation models, many researchers and developers have used optimization theory to establish the control method for vehicle motion control to minimizes the energy consumption of single or networked vehicles [100–106].
Although these works could successfully estimate the energy consumption in each application domain, the accuracy of road vehicles energy optimization was limited due to the lack of precise information on the driving characteristics of each driver. In order to improve the modeling and estimation accuracy of the energy consumption, driving characteristics of each individual driver must be explicitly represented.

From these considerations, this chapter proposes a modeling framework to estimate the energy consumption of a vehicle personalized on the individual behavior of the driver. To achieve this goal, research has been conducted on the selection of an appropriate modeled situation and of a mathematical model to reproduce this situation. Then an input-output combination used to provide the lowest the modeling error is proposed in combination with a model identification method for robust model fitting. Once these elements shown, the model accuracy assessment is done by comparing measured and simulated data based simulations of vehicle energy consumption.

This chapter is organized as follows: in section 4.2, the developed energy consumption evaluation perspective is explained. In section 4.3, the implementation of the PrARX model to represent the driver-vehicle dynamics is described in detail. Modeling method, inputs, output, and the identification process are also explained. Section 4.4 introduces the experimental setups to collect the driving data. Section 4.5 describes the energy consumption evaluation method in detail, and in section 4.6 results of the energy consumption evaluations are discussed for various situations. Finally section 4.7 is dedicated to application proposals.

4.2 Personalized energy consumption evaluation

In this study, the driver and vehicle are considered as a single entity. This choice has been taken based on prior study analysis such as MacAdam work [16], and based on the data analysis done in section 2.3.2. Indeed, we could find that for equivalent type of vehicles with different powertrains, the driving dynamics was mostly driven by the desires of the human, the vehicle acting like a limiting factor. The selected situation is the car-following situation. This situation has been selected due to the fact that it is the most encountered situation in crowded areas, most likely to suffer from vehicle pollution issues.

Figure 4.1 depicts the overall architecture of the proposed energy consumption evaluation system. The proposed system explicitly includes the driver-vehicle model which has two main inputs: a specific set of parameters depending on the driver, and a velocity pattern of the leading vehicle. Definition of the driver-vehicle model is described in section 4.3.1. The ego-vehicle velocity profile is calculated as the output of the driver-vehicle model. Finally, the vehicle energy consumption is estimated by using a detailed car model (in this work, IPG Carmaker is used).

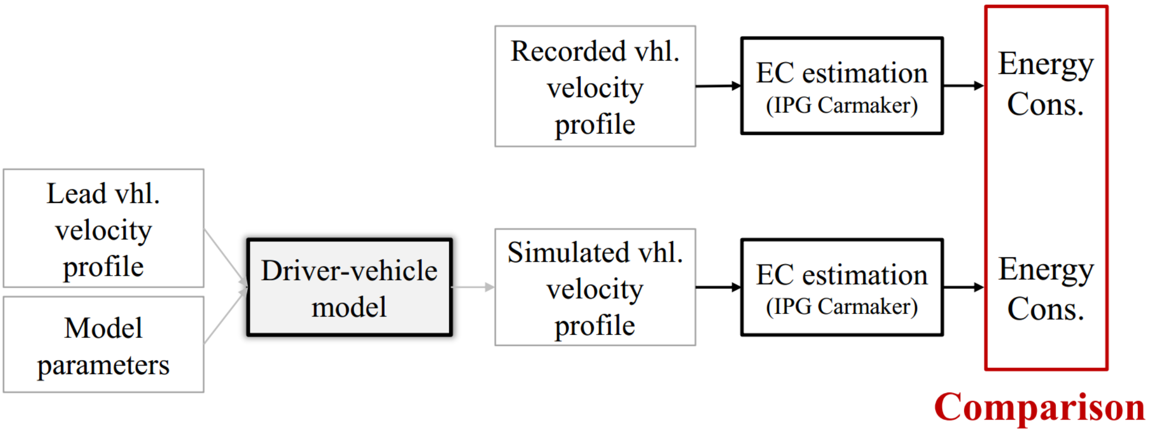

Recorded vhl. velocity profile
EC estimation (IPG Carmaker)
Energy Cons.
Lead vhl. velocity profile
Model parameters
Driver-vehicle model
Simulated vhl. velocity profile
EC estimation (IPG Carmaker)
Energy Cons.
Comparison

Figure 4.1: Driver personalized vehicle energy consumption evaluation framework. Comparison of the estimation of the energy consumption of a vehicle based on a recorded vehicle velocity profile and a simulated vehicle velocity profile.

As shown in Figure 4.1, the driver-vehicle model is validated by comparing experimental energy consumption to simulated driver-vehicle energy consumption. Thus, the proposed framework enables us to evaluate the energy consumption of different drivers, depending on the choice of the leading vehicle velocity pattern and depending on a vehicle powertrain.

Obviously, careful selection of the driver-vehicle model is a central issue in this framework. The driver model should be simple enough to be used in optimization procedures, and precise enough to realize accurate reproduction of the driver-vehicle behavior. Model selection, definition and implementation are detailed in the following section.

4.3 Definition of the driver-vehicle model

This driver-vehicle model is based on the probability weighted autoregressive exogenous (PrARX) model (see section 2.4.2). ARX class model have been selected for their ability to model a large variety of driving styles. The model inputs and output configuration can be designed based on large data-sets correlation analysis, and then each individual driver behavior can be represented by the value of the ARX model parameters [59]. Nevertheless, a single ARX model is not able to represent accurately all driving situations precisely, thus multi-mode ARX models are introduced. The PrARX model is a modified version of piecewise autoregressive exogenous (PWARX) with a mode switching mechanism based on a probability density function. Unlike a

PWARX model, with discrete mode switching, the PrARX model soft (continuous) switching mechanism has been selected to represent the decision process of a real driver. By using the PrARX as a driver model, we aim to be able to represent as precisely as possible a large variety of driving styles, by reproducing the main driving dynamics of each individual driving situation by ARX models, as well as the decision making process by probabilistic mode switching. This study focuses on the analysis of a single following vehicle. Vehicle platooning modeling would require more in-depth analysis of information propagation on the modeled string.

The above sections describe the model inputs, output, model order and the identification procedure. These elements have been uniquely designed to fulfill the requirements of vehicle energy consumption evaluation.

4.3.1 Probability Weighted ARX model setup

The goal of this study is to reproduce personalized vehicle behavior in vehicle-following task. The PrARX model is used to model the integrated behavior of the driver and vehicle dynamics. The measured data of leading vehicle and the model output are used to create the regression vector (see Figure 4.2). The PrARX model is used to predict the outputs of the integrated behavior at $k + \tau$ based on the current input at time k. A delay τ is applied to the model input to represent the drivers' cognitive reaction time (300ms [16, 41]). Then, the output of each ARX model and the mode probabilities (weighting parameters) are calculated by using the PrARX model. The PrARX model with input-delay is given by:

$$\begin{gathered} i_k = f_{inputs}(u_k, \hat{y}_k) \\ r_k = i_{k-\tau} \\ \hat{y}_k = f_{PrARX}(r_k) \end{gathered} \tag{4.1}$$

where u_k is the exogenous input, i_k is the regression vector without delay, r_k is the regression vector, f_{inputs} is the function to calculate the PrARX model inputs, and y_k represents the PrARX model output.

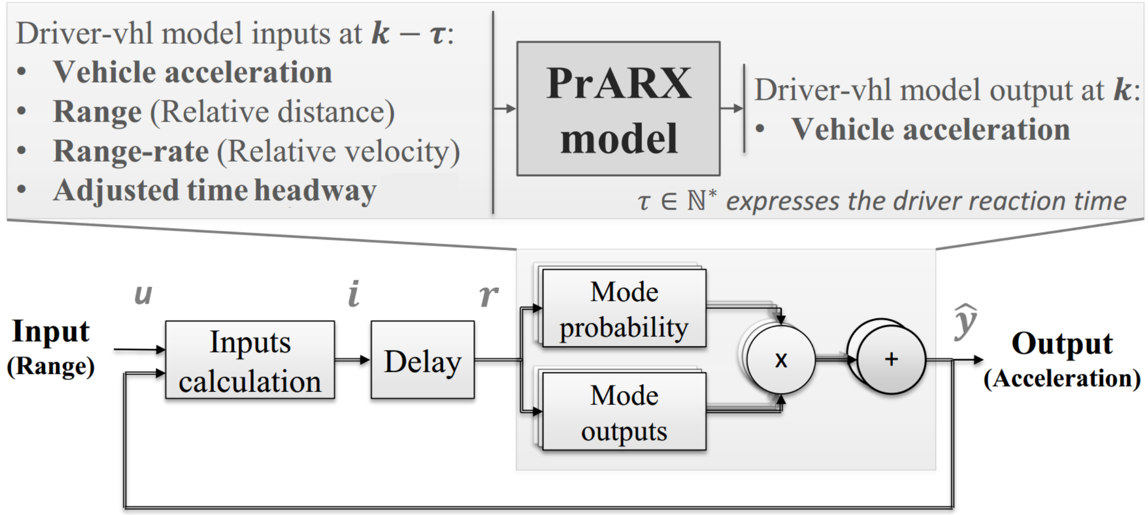

Driver-vhl model inputs at $k-\tau$:
• Vehicle acceleration
• Range (Relative distance)
• Range-rate (Relative velocity)
• Adjusted time headway
PrARX model
Driver-vhl model output at k:
• Vehicle acceleration
$\tau \in \mathbb{N}^*$ expresses the driver reaction time
Input (Range)
u
i
r
Inputs calculation
Delay
Mode probability
Mode outputs
x
+
$\hat{y}$
Output (Acceleration)

Figure 4.2: Driver-vehicle model, expressed as the feed-back implementation of a PrARX model with input-delay. Relation between inputs and outputs.

Definition of the driver-vehicle model is given as follows:

$$
\begin{gathered}
i_k = f_{inputs}(u_k, \hat{y}_k)\text{with} \\
f_{inputs}(u_k, \hat{y}_k) = \begin{cases} a_k \\ R_k \\ RR_k \\ \frac{1}{1+THW_k} \end{cases} = \begin{cases} \hat{y}_k \\ u_k \\ \dot{u}_k \\ \frac{\sum_{i=0}^{k}(\hat{y}_i)+v_0}{s_k+\sum_{i=0}^{k}(\hat{y}_i)+v_0} \end{cases} \\
r_k = i_{k-\tau} \\
\hat{y}_k = f_{PrARX}(r_k) \\
THW_k = \frac{R_K}{v_k} = \frac{u_k}{\sum_{i=0}^{k}(\hat{y}_i)+v_0}
\end{gathered}
\tag{4.2}
$$

where a refers to the vehicle acceleration, R refers to the range, RR refers to the range-rate, THW refers to the time headway, v_k for the vehicle velocity at time k. The choice of variables in 4.2 is discussed in 4.3.2.

4.3.2 Model parameters identification

Original parameter identification of the PrARX model is based on a steepest descent method [59], with a cost function is defined by the Euclid norm of the output error. Although both parameters in the ARX model and the softmax function can be identified simultaneously by a single algorithm, this identification scheme is a non-convex optimization problem. In order to increase the level of reliability and accuracy of the identified of the PrARX model, a two-stage identification process is newly developed in this work. In the first stage, a classification technique is applied to the data, and PrARX hyperplanes parameters are identified. The set of data is separated into subsets depending on the preferable mode separation, and the mode separation parameters η are identified with a multinomial logistic regression method. In the second stage, the parameters of the ARX models θ are identified using a steepest descend method. The advantage of using classification over clustering in the case of applying a multi-mode model to analyze realistic driving data lies in the nature of the observed data. If constancy in the physical understanding of the modes parameter is desired, data clusters must have high margin hyperplanes separation, enabling similar data clusters formation for every identification. Unfortunately, the available data does not show clear separation pattern. Thus, in this work, subjective prior-knowledge about the data classification is assumed, based on the vehicle acceleration, to describe the following driving situations: acceleration, deceleration, and constant speed.

4.3.2.1 Choice of the regression vector

Reproduction of the driving behavior depends not only on the structure of the model, but also on the selection of explanatory variables of the model. The regression vector must have strong relation with the output of the ARX models. In addition, the regression vector must be able to distinguish the driving modes, i.e. to represent the partition between modes. In this work, the output of the model is set to be the longitudinal acceleration of the vehicle because our goal is the evaluation of the energy consumption. Generally, it seems natural to select the range between leading vehicle, and range-rate as elements of regression vector in the case of vehicle following task. In addition, some indexes have also been considered as variable, which express the feeling of the driver (KDB [93], PRE (Perceptual Risk Estimate) [109]). These indexes are commonly used to trigger emergency systems (e.g. emergency braking), however, it is difficult to use them for behavior reproduction. In this work, we tried to find explanatory variables that can be linked as simply as possible to the output of the system (vehicle acceleration). To understand the necessary input variables for ARX models, multivariable linear regression statistical tests were performed in each mode based on real world recorded data. According to the values of the standard error of the coefficient estimate and on the p-values, it was observed that the past acceleration and range-rate were the two most significant variables to estimate the current acceleration value. The range-rate is a fundamental variable to calculate the output acceleration of the vehicle (consistent p-value lower than 10^{-8}). This result is also reported in the Gazis-Herman-Rothery (GHR) model [3]. The past acceleration is obviously linked to the current acceleration due to the low dynamics of the car (lower than 0.5Hz), and the fact that the model is running at 10Hz. Time headway (time to collision) also has shown strong importance. The range was not directly linked to the model output, but is used in the mode determination process. Note that the magnitude of the linkage between some variables and output highly depends on driver and on the driving mode.

Thus, the selected inputs are the acceleration of the driven vehicle [m/s^2], distance between the driver and leading vehicles (range [m]), relative velocity between the vehicles (range-rate [m/s]), and an adjusted inverse of the time headway ((velocity+1)/range) [1/s] [110, 111]. The inverse of the time headway was adjusted to provide information even when the vehicle is stopped. Time headway provides an improved stability to the output model response as stated later. The identified parameters of the corresponding variable can be interpreted to represent the driving characteristics of each driver. For example, aggressive drivers tend to base their judgment on the range and the range-rate, while soft drivers rely mostly on the time headway.

To provide more information about the effectiveness of the selected input parameters, Figure 4.3 shows the verification results of the driver-vehicle model with input delay in the cases of different regression vector choice. The regression vectors definition are in Table 4.1:

Table 4.1: Regressor vectors definition for Figure 4.3

Name				
wo RR	a	R	$adjTHW^{-1}$	
wo R	a	RR	$adjTHW^{-1}$	
wo THW	a	R	RR	
wo RR	a	R	RR	$adjTHW^{-1}$

where wo stands for without, RR for range-rate, R for range, $adjTHW^{-1}$ for adjusted inverse of time headway, and RegV for the final regression vector.

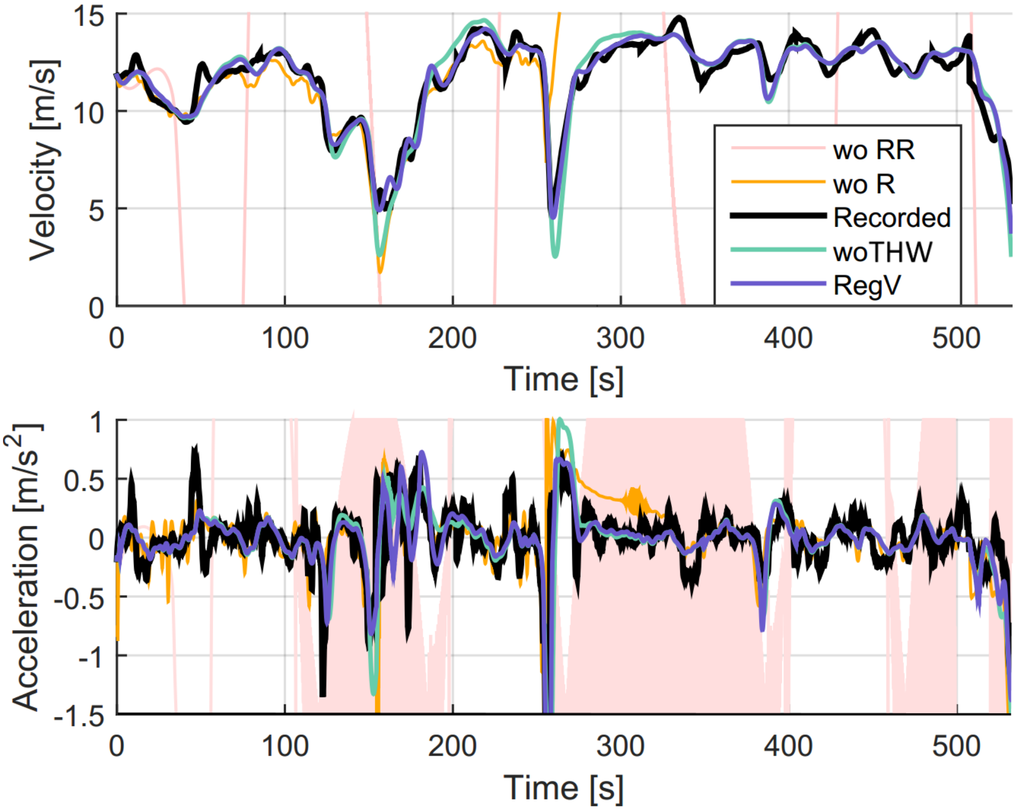


Figure 4.3: 3 modes PrARX input-delay model output depending on the selected learning regression vector. The label Recorded represents the reference recorded vehicle following profile. Definition of the labels is in Table 4.1.

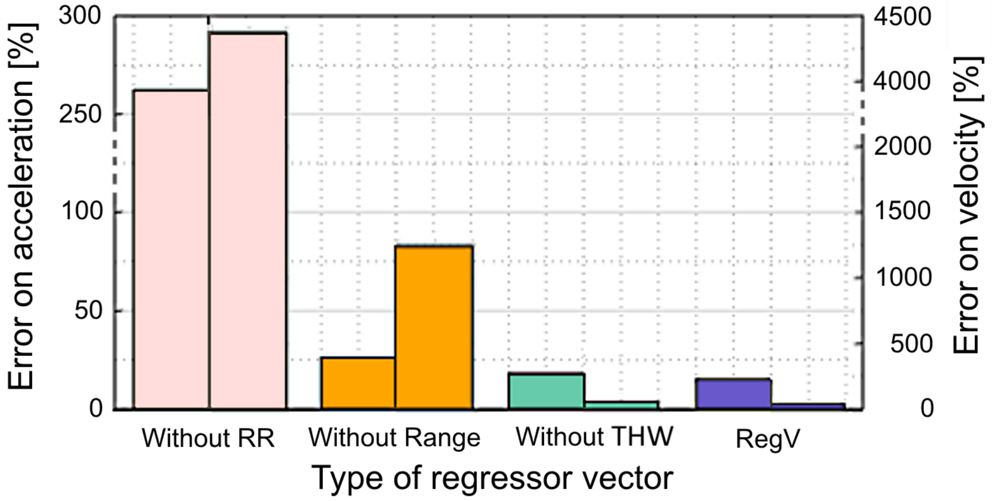


Figure 4.4: Acceleration error of the input-delay model output on the left, and velocity error on the right, depending on the type of regression vector. The error is the Euclid norm of the difference between the reference data and the identified 3 modes PrARX input-delay model output.

4.3.2.2 Data classification for mode definition

Classification is used to determine the modes of the PrARX model. According to the distribution of the recorded driving data and to the energy estimation error of the resulting model, we decided to classify the data into three clusters based on the vehicle acceleration. The defined mode definition is shown in Table 4.2. This segmentation implies to differentiate low-band dynamical driving mode and high-band dynamical driving modes. High-band dynamical driving modes are representative of the acceleration and deceleration phases. To avoid sudden mode changes and take advantage of the smooth mode switching of the PrARX model, overlapping between the simple clusters is considered on a 0.1 m/s^2 range of the acceleration data.

Table 4.2: Learning data clusters definition

Acceleration [m/s^2]	Cluster name	Mode number
$(-\infty, -0.35]$	Deceleration cluster (high-band)	1
$[-0.45, 0.45]$	Low-band dynamics cluster	2
$[0.35, +\infty)$	Acceleration cluster (high-band)	3

Figure 4.9 shows the velocity and acceleration profile of the output of simulation using the 3-modes PrARX model with input-delay, which is identified from real-world data. It can be observed that the behavior reproduction is successfully made, and it is expected to play a key role for precise energy consumption evaluation. The Mode probability graph in Figure 4.9 illustrates the instantaneous mode probability which is used as a weighting factor for the calculation of PrARX model output.

Without range or range-rate, the driver-vehicle behavior becomes unstable particularly in high-band dynamics domain. The time headway helps to stabilize the behavior. To get more information about the precision of the reproduced vehicle dynamics, Figure 4.4 shows the error in terms of reproduced accelerations and reproduced velocities, that is Euclid norm of the difference between the recorded vehicle data and the output of the simulation using driver-vehicle model. According to the error bar graphs, we can see that the selected regression vector (i.e., RegV) provides the best acceleration and velocity reproduction performance among the considered set. Since the selected regression vector does not depend on the absolute velocity of the vehicle, different driver models can be identified depending on various velocity spans, congestion states or road types.

4.3.2.3 Overall flowchart of identification process

Overall flowchart of the identification process is depicted in Figure 4.5. Two types of data sets are used during the identification process. The first one is for the parameter identification, and the other one is for the model verification (simulation). The first data set contains as much information as possible. This data set is manually preprocessed to remove noise and outliers. Then data is decimated to reduce the computation burden of the identification process, while ensuring to have enough data in each cluster. The identification step takes about 3 minutes on a personal computer (CPU i7 870, RAM 8Gb). The second data set (velocity profile of leading vehicle) is used to run the simulation using PrARX model with input-delay. The velocity profile of the leading vehicle used for verification can be any velocity profile, as long as the underlying dynamics are coherent with the first data set.

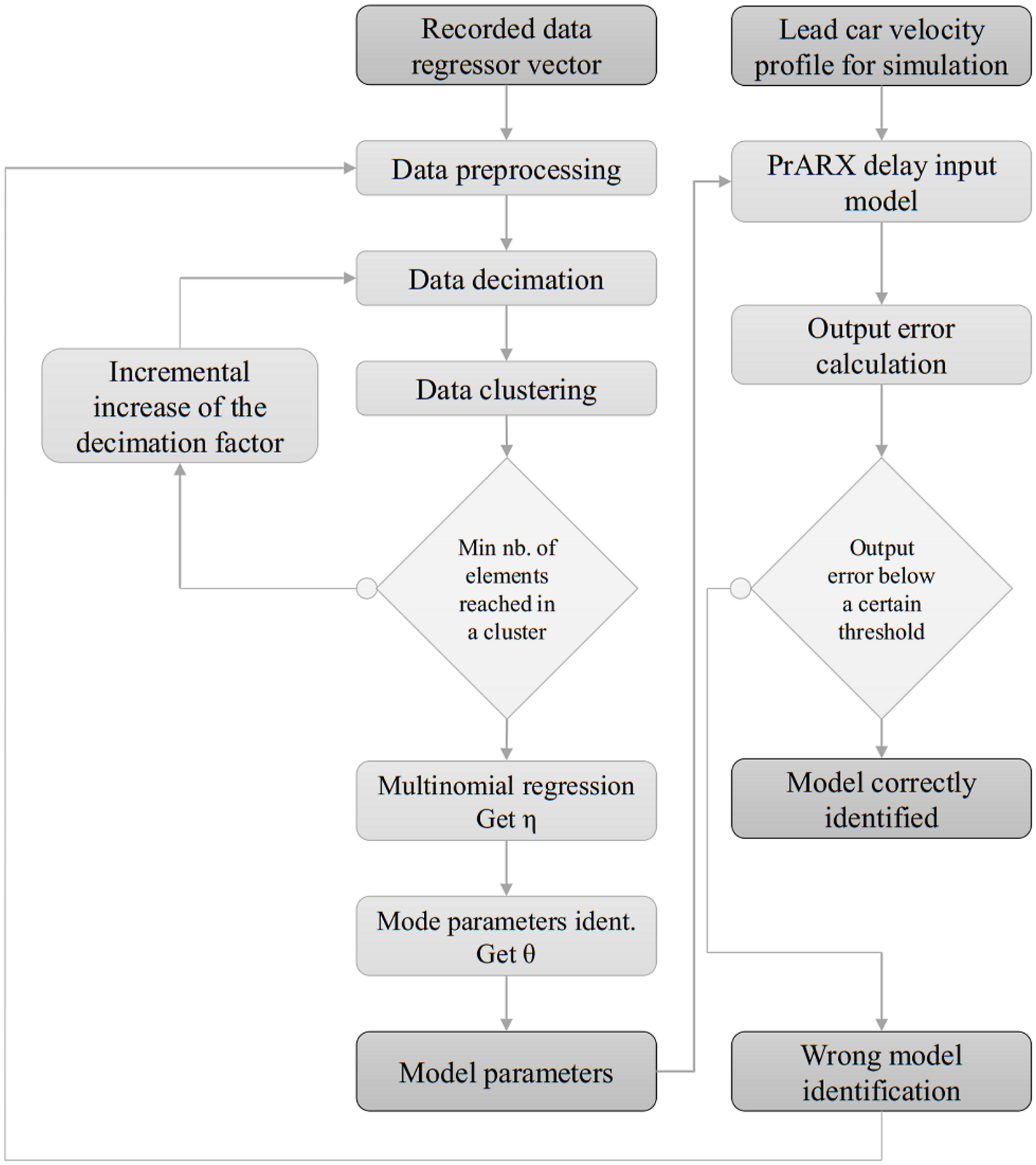

Recorded data regressor vector
Lead car velocity profile for simulation
Data preprocessing
PrARX delay input model
Data decimation
Output error calculation
Incremental increase of the decimation factor
Data clustering
Min nb. of elements reached in a cluster
Output error below a certain threshold
Multinomial regression Get η
Model correctly identified
Mode parameters ident. Get θ
Model parameters
Wrong model identification

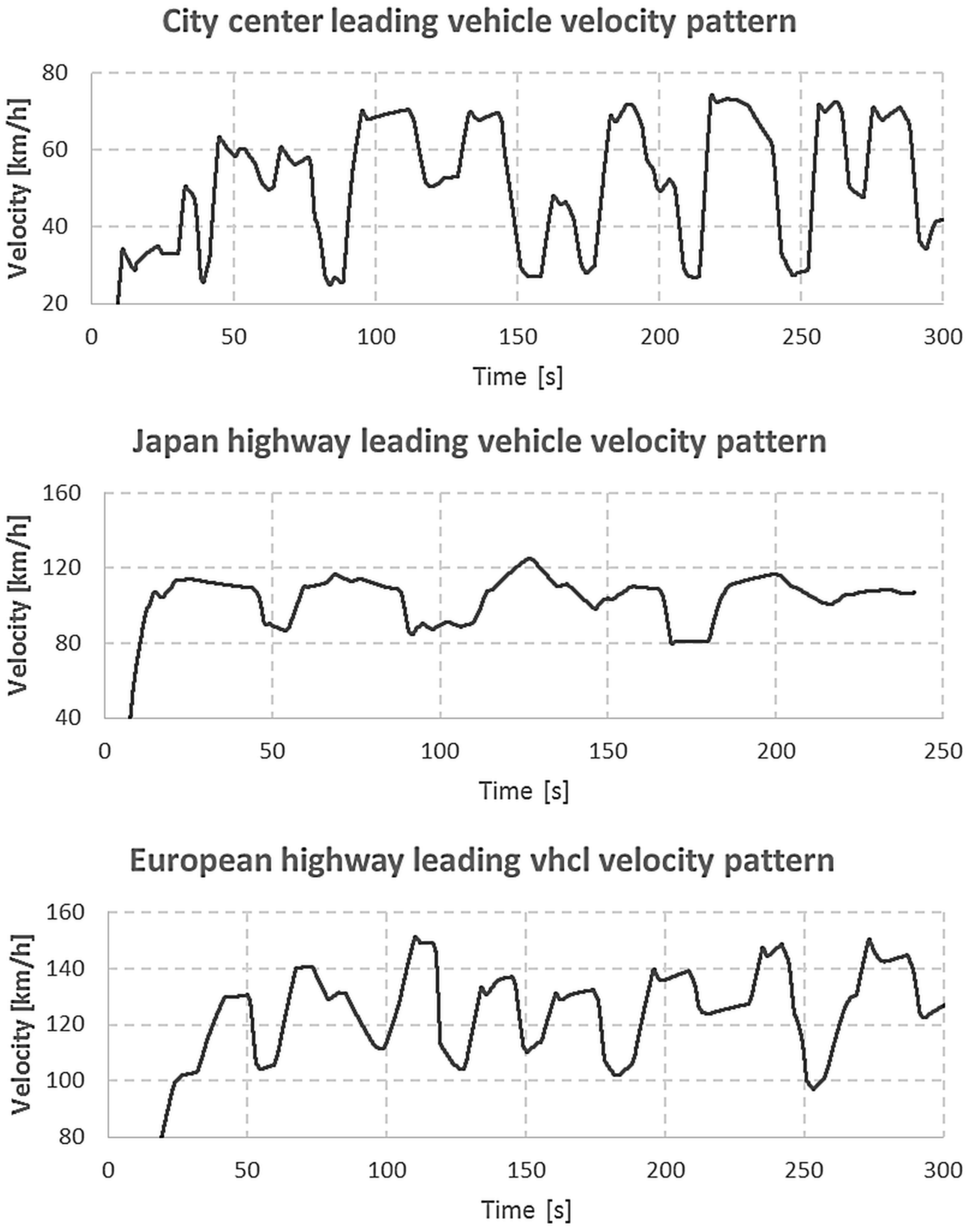
Figure 4.5: Flowchart of the PrARX input-delay model identification process.

4.4 Experimental data

In order to get data from different driving styles in various situations, two types of experimental setups were used for data collection. At first, data was collected by using a driving simulator (see section 2.2), which enabled us to control all the environmental parameters. Then a real world experiment was executed to get realistic driver-vehicle dynamics (see section 2.3).

Four examinees drove in the driving simulator. The leading vehicle ran according to three different velocity patterns designed to represent typical driving scenarios (See Figure 4.6):

- 30 to 70 km/h pattern: representing city use.
- 80 to 110 km/h pattern: representing extra-urban/Japan highways.
- 100 to 150 km/h pattern: representing European highways.



City center leading vehicle velocity pattern

Japan highway leading vehicle velocity pattern

European highway leading vhcl velocity pattern

Figure 4.6: Velocity patterns of the leading vehicles used in the DS experiments.

For the real-world measurement, data of the highway experiment has been used. The experimental setup as has been introduced in section 2.3.3. This experimental measurement setup and situation enabled us to get large variations of driving dynamics with a consistent leading vehicle velocity profile for all examinees.

4.5 Energy consumption evaluation

The energy consumption of the vehicle is estimated by inputting a vehicle velocity pattern, which is calculated by using the driver-vehicle model, to the car dynamics simulation software Carmaker (IPG Automotive), as shown in Figure 4.1. Carmaker is known to be able to calculate the fuel consumption with high accuracy based on the different car losses, including the engine efficiency mapping of the vehicle. This software is industry standard, and it is used by the biggest manufacturers to model car dynamics and powertrains. The fuel consumption volume flow $\dot{vol}_f$ is calculated by

$$\dot{vol}_f = \frac{\dot{m}_F(\omega_{Eng}, Trq_{Eng}) + |P_{Eng}|}{\zeta_F * 3.6 * 10^9} \tag{4.3}$$

where $\dot{m}_F(\omega_{Eng}, Trq_{Eng})$ is the specific mass flow extracted from the engine mapping (see Figure 4.7), ω_{Eng} the engine frequency of rotation, and Trq_{Eng} the torque load at the crank shaft. ζ_F is the petrol density (0.75 kg/L), and P_{Eng} is the engine output power. (4.3) is provided by IPG Carmaker.

The speed profile tracking function of Carmaker can realize very precise reproduction of any velocity pattern. The simulated environment is a flat straight line, and the selected powertrain are 250hp and 130hp petrol engines for the DS and real-world driving, respectively. The average velocity difference between the evaluated pattern and the reproduced pattern is 0.3km/h and the median absolute deviation is 0.1km/h.

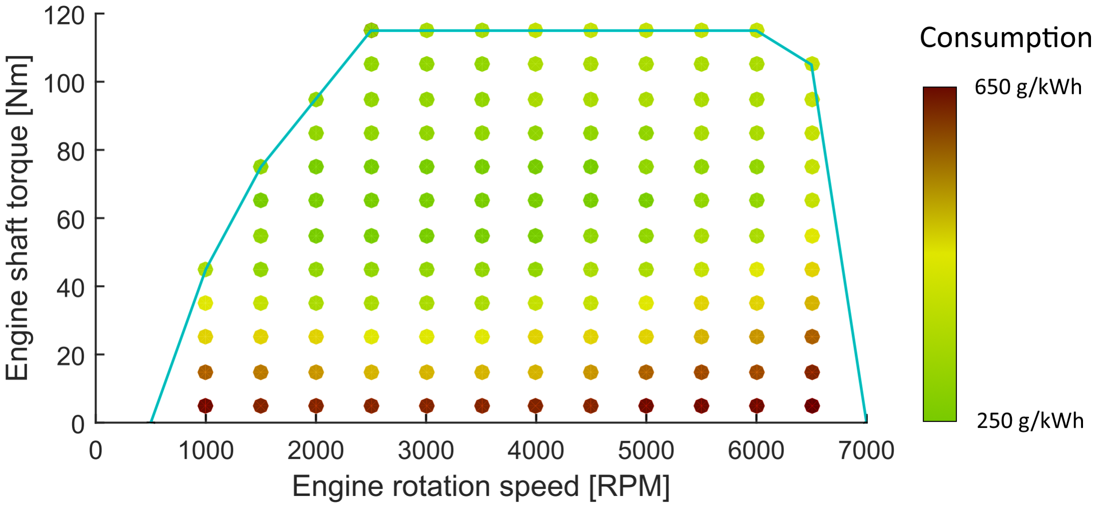


Figure 4.7: Engine mapping of the 130hp petrol powertrain (IPG Carmaker). The blue line represents the torque at full load, and colored dots the specific fuel consumption.

Using the energy consumption evaluation scheme shown in Figure 4.1, the fuel consumption of the different vehicles can be assessed and compared, considering the variety of driving characteristics.

4.6 Results and analysis

In this section, results of fuel consumption analysis are shown and discussed.
Fuel consumption modeling error is calculated by comparing the fuel consumption estimated from the driver-vehicle model, and the fuel consumption estimated from the reference velocity profile used to train the driver-vehicle model. The formula is detailed in equation 4.4.

$$error_{FC}[\%] = \frac{FC_{estimate} - FC_{reference}}{FC_{reference}} * 100 \tag{4.4}$$

where FC stands for fuel consumption.

4.6.1 Results using data from driving simulator

Tables 4.3 and 4.4 show the fuel consumption estimation values and their estimation errors, respectively.

Table 4.3: DS experiment fuel consumption values [L/100km]

	Recoded vehicles	**Simulated vehicles**
City center **Aggressive driver**	10.12	9.81
City center **Soft driver**	8.63	8.11
Extra-urban **Aggressive driver**	9.76	8.71
Extra-urban **Soft driver**	7.45	7.15
European highway **Aggressive driver**	10.75	10.46
European highway **Soft driver**	8.37	8.24

Table 4.4: DS experiment fuel consumption estimation error [%]

	Aggressive driver	**Soft driver**
City center	-3.1	-5.9
Extra-urban	-10.7	-4.1
European highway	-2.7	-1.6

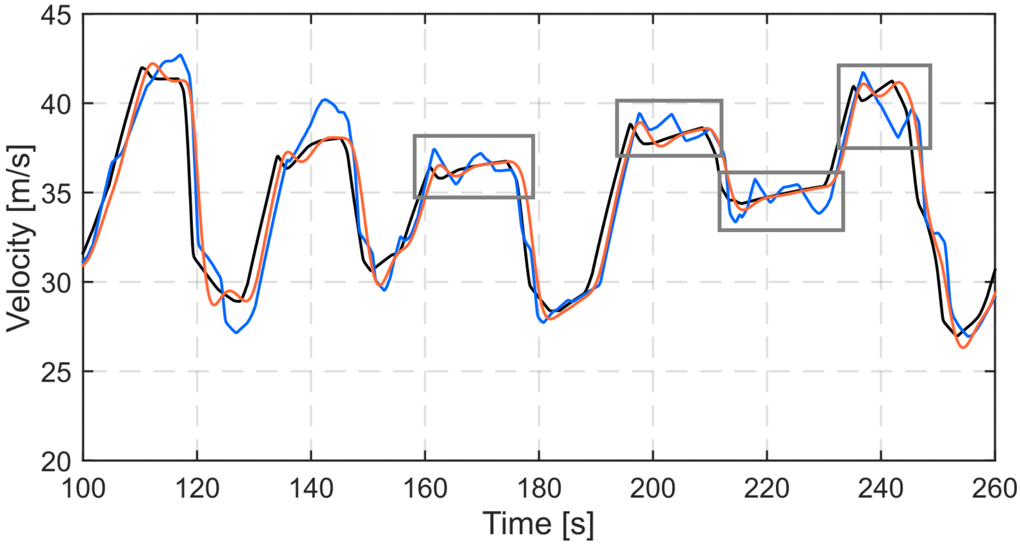


Figure 4.8: Velocity of the driver-vehicle model output. DS European highway profile with aggressive following. In black the leading vehicle, in blue the recorded ego vehicle, in orange the driver-vehicle model simulated ego vehicle. The oscillatory behavior of the aggressive driver during constant velocity phases is squared in grey.

The estimation error of the fuel consumption varies from -10.7% to -1.6%, with an average error of -4.6% and an absolute standard deviation of 3.4%. The energy consumption value is underestimated due to the low-pass characteristics of ARX models. The lack of acceleration feeling of vehicle in the DS makes the driver act quite aggressively, and examinees struggled to follow the leading car with creating acceleration oscillations in usual low dynamical band. Figure 4.8 shows these oscillations squared in black. This behavior cannot be correctly modeled by the PrARX input-delay model, due to the absence of correlation between the inputs and this output.

4.6.2 Results using data from real-world driving

Table 4.5 shows the fuel consumption evaluation values for three drivers on real-world measurement. Lead vehicle represents the fuel-consumption of the leading car. Due to the impossibility to exactly realize the desired leading vehicle velocity profile (shown in Figure 2.19), leading vehicle energy consumption is calculated for each driver-vehicle model using realized velocity profile. Follow recorded represents the fuel consumption of the vehicle used for driver-vehicle model identification, and PrARX input-delay represents the fuel consumption of the simulated driver-vehicle model.

Table 4.5: Read world experiment fuel consumption evaluation [L/100km]

	Driver 1	**Driver 2**	**Driver 3**
Lead vehicle	5.59	5.19	5.87
Follow recorded	6.02	5.59	5.81
PrARX input-delay	6.02	5.48	5.59
Error values	**-0.06**	**-1.8**	**-3.8**

In Table 4.5, we can observe good results in energy consumption estimation. The average estimation error is -1.9%, and the absolute standard deviation 1.5%. The energy estimation error is much lower in real-world environment than in the driving simulator experiment.

Figure 4.9 illustrates the driver-vehicle model dynamics reproduction ability based on real-world recorded data. The low estimation error of energy consumption in real-world experiment is due to the fact that examinees seem to drive the vehicle with lower frequency dynamics in real world, so that the recorded and reproduced signals are no more limited by the low pass behavior of the PrARX input-delay model. Moreover the observed correlation between the leading vehicle and ego-vehicle is much better than in the driving simulator.

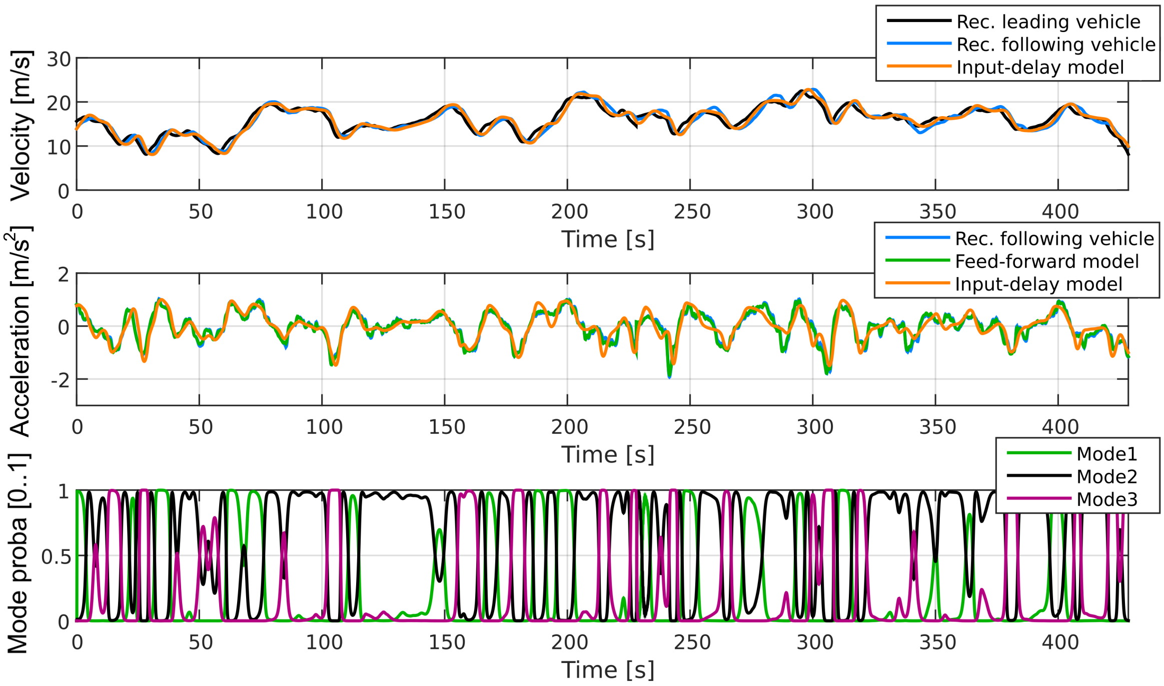


Figure 4.9: Velocity, acceleration, and modes probability weight of an identified 3 modes PrARX input-delay model. ”Rec. leading vehicle” represents the recorded leading vehicle used for simulation, ”Rec. following vehicle” represents a section of the recorded ego vehicle of the learning phase, ”Feed-forward model” is the output of the PrARX model when using pre-calculated learning data regression vector, without feedback loop, ”Input-delay model” is the output of the driver-vehicle model by using ”Rec. leading vehicle” for the lead vehicle.

4.7 Application examples

It could be observed in the previous sections that the PrARX input-delay model is able to provide ego-vehicle dynamics with enough precision to evaluate first order energy consumption of the vehicle. The computation cost of the PrARX input-delay model being very low, online use in a vehicle is possible. Although the parameter identification process requires high computational cost, thanks to the increase of V2X communication in recent years, parameter identification can be done remotely on most cars without any implementation on the vehicle computer system. Based on this information, two in-vehicle and one traffic flow model oriented applications are proposed in this section. The first application can be described as a customer decision assistance system for the choice of an appropriate powertrain in buying new vehicle. The second application aims to help the driver to reduce his fuel or electricity consumption by challenging his behavior with somebody elses. The last application is a method to evaluate energy consumption of vehicles embedded in a traffic flow model.

4.7.1 Customer decision assistance for powertrain choice

The first application of the developed driver-vehicle model with energy consumption estimation is based on the ability of the model to reproduce a users behavior on a variety of lead velocity patterns, as long as driving situation is equivalent. Each different vehicle powertrain has specific high and low efficient zones. Depending on the human driving manner, different types of powertrains will be adapted to different users. The goal of this application is to help customers to select an appropriate vehicle powertrain depending on their individual driving habits.

The typical situation places a customer comparing some possible new vehicles. The parameters of customers vehicle-following behavior model have already been identified during daily driving. These parameters can then be applied to classic regulatory cycles or any usual velocity pattern. Knowing that every manufacturer is able to provide the powertrain performance map of their vehicles, the customer will be able to receive a personalized estimation of the energy consumption of the vehicle depending on his personal behavior.

4.7.2 Social eco-driving challenge

The second proposed application of the behavior personalized energy consumption estimation model is based on the ability of the driver-vehicle model to reproduce different driving behaviors under identical lead velocity pattern. The idea behind social eco-driving challenge is based on the concept of social games [112]. The aim of this application is to get people interested in eco-driving by challenging them to outperform others. Setting goals to reach their best results in the form of eco-indicators is the main focus to obtain good efficiency results [113]. Thus the combination of goal reaching and the interest of social game could assist the development of a platform which proposes advice to help drivers to reduced energy consumption.

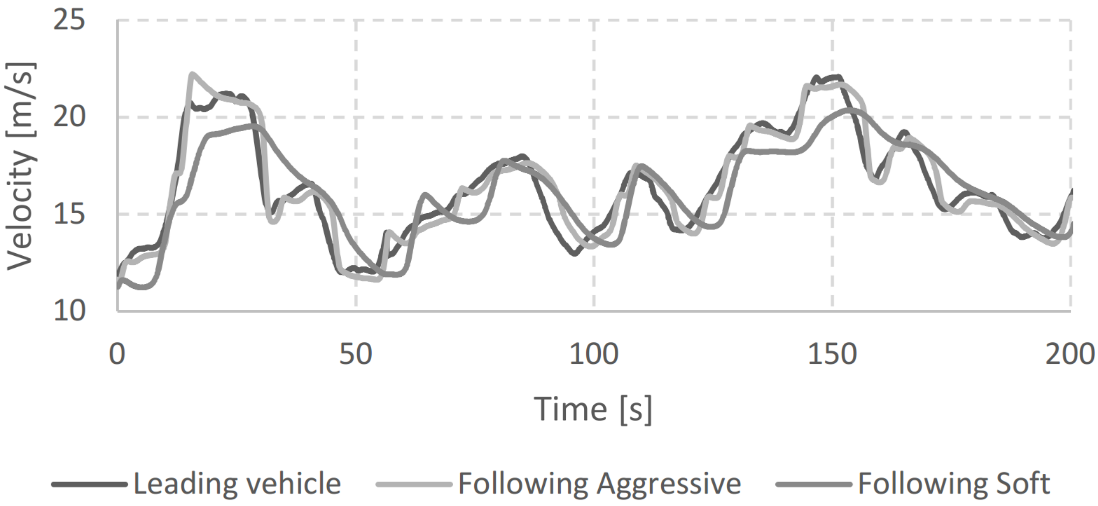


Figure 4.10: A leading vehicle from real-world, followed by two different PrARX input-delay models. The PrARX input-delay models are representative of an aggressive and a soft driver. These models have been identified from distinctive driving measurements.

The PrARX input-delay model can be used to calculate the reaction of different drivers online, and thus compare in real time the energy consumption of virtual drivers. As an example, comparison of behavior in the cases of using aggressive driver model and soft driver model is shown in Figure 4.10. The estimated energy consumption are simulated and compared based on two leading vehicle velocity patterns. The driver-vehicle parameters have been identified from city-center driving situation in section 4.6. The selected leading vehicle velocity patterns are: a DS recorded velocity profile, and a velocity pattern measured in real-world (R-W) experiment.

Table 4.6: Comparison of FC of D-VMs following DS recorded vehicle 1

Evaluated — Reference	**Aggressive**	**Soft**
Aggressive	—	20.9%
Soft	-17.3%	—

Table 4.7: Comparison of FC of D-VMs following DS recorded vehicle 2

Evaluated — Reference	**Aggressive**	**Soft**
Aggressive	—	20.0%
Soft	-16.7%	—

Tables 4.6 and 4.7 show the relative difference of fuel consumption between the two driver-vehicle models (D-VMs) with different leading vehicle velocity patterns.

The fuel consumption comparisons in Tables 4.6 and 4.7 show that the driver models keep their own relative energy consumption behavior independently on the leading vehicle velocity profile. The soft D-VM is consistently 17% more energy efficient than the aggressive D-VM. Therefore, the PrARX input-delay model is a good candidate to compare drivers based on a selected driving situation.

4.7.3 Estimation of vehicle energy consumption in traffic flow model

Traffic flow models enable users to analyze wide road networks dynamical behavior depending on the road topology, the traffic flow density, the basic vehicle characteristics and other macroscopic parameters [72, 107, 114, 115]. Nevertheless, conventional driver-vehicle models used in traffic flow simulation cannot provide realistic microscopic behavior due to the lack of personalized driving characterization.
PrARX input-delay model can be used as an adaptive cruise control model when implemented with the Virtual Leading Vehicle (Vlv-ACC) system [9]. Vlv-ACC system is based on the philosophy of action-point microscopic traffic flow models. Thus, by embedding the developed driver-vehicle model in the Vlv-ACC system, an interesting traffic flow model can be developed. This model can provide more precise and user personalized driver-vehicle dynamics of certain road sections, and as the results, energy consumption of particular vehicles can be assessed in the context of traffic flow.

The following chapter presents a novel parameters identification method for hybrid dynamical system models, where parameters have stochastic and time-varying characteristics. This method has been developed to allow the identification of personalized time-varying driver behavior.

Chapter 5

Driving behavior analysis using a filtered sequential Monte-Carlo approach

5.1 Introduction

Thanks to the recent development of computer technology, data-centric system design is attracting great attention [59–62, 80, 116]. In the field of system identification, although numerous dedicated mathematical models have been proposed to represent target systems [3, 16, 34, 107], discrete-continuous hybrid system modeling has great potential to represent complex dynamical behavior including switching mechanisms [75].

As a result of its high describability and understandability, hybrid system modeling has been applied to various domains, such as communication systems, autopilot systems, automotive engine control, traffic control, and chemical processes [10,59–62,75,80,116,117]. A promising application domain of hybrid system modeling is the human behavior analysis and reproduction, due to the possibility to represent both the decision making and the motion control aspects of human behavior.

From the viewpoint of data-centric modeling, the PieceWise AutoRegressive eXogenous (PWARX) model is extensively used, and various parameters identification have been developed [81]. The clustering approach is based on dynamics clustering and on identification of each clustered data set [86]. The bounded-error approach allows to define the maximal identification error [87]. The mixed-integer programming approach guarantees to converge to a global optimum [88]. The algebraic approach defines an analogy to the identification and decomposition of an algebraic variety [89]. Finally the Bayesian approach uses the Bayesian inference to identify both parameters and mode partitioning of hybrid systems from noisy data [90].

From this perspective, several hybrid system modeling methods have been proposed, especially focusing on applications to the human driving behavior analysis. A Stochastic Switching ARX (SSARX) model has been developed by extending a conventional Hidden Markov Model [60]. Hierarchical PWARX (Hi-PWARX) has been developed with the idea to create a hierarchical structure of the data based on unsupervised clustering technique [61]. Finally the Probability-Weighted ARX (PrARX) model has been developed as a new hybrid system model wherein the mode switching is represented by softmax function, which represents the probability of mode occurrence [59].

These works are focused on the identification of time-invariant hybrid system models. When analysis of the human driving behavior is considered, it has been shown that stochastic and time-varying characteristics should be included in addition to decision taking mechanisms [2]. Each driver shows a different response to a given stimuli, leading a driver's individual statistical dispersion in the reproduction of a given action, and under long-time driving situations, drivers' behavior can vary drastically. The understanding of time-varying characteristics of driving behavior through hybrid system model parameters can inform on driving consistency, expressed as short-term variance, and long-term driving characteristics, expressed as global model parameters variations. This can also be used as a new source of information for the design of better driving assistance and health monitoring systems.

These considerations highly motivated us to develop a new identification technique for time-varying parameters of hybrid system models. The parameter identification process should identify time-varying parameters while complying to the parameters dynamics.

In the case of fitting a driving model on real-world measured data, optimality of the identified parameter is not the final goal. The model can never perfectly represent the real situation, and noise in the measurement and time-variability characteristics of the driving behavior are against the concept of optimal solutions. The major concern in this type of identification process is the ability to avoid local minima, and the ability to get parameters estimations within a known error margin. These perspectives were lacking in the previous works [81, 86–90, 118–121].

To realize stochastic and time-varying hybrid system models parameters identification, parameters are regarded as random variables and identified with a Bayesian approach [90, 118, 119]. The parameters' time dynamics are also bounded using a time-smoothing technique.

The proposed identification method differs from the conventional Bayesian approaches [90, 118, 119] in the sense that as far as we know, former studies did not consider time-varying parameter explicitly. In the main reference article on this topic [90], the prior knowledge of Bayes inference is based on the prior time step. Thus the estimated parameter $\theta(k)$ at time step k depends on the prior time step identified parameter $\theta(k-1)$, on the observation (model output) $z(k)$ and on the probability density function (pdf) p. With this formulation, the identification process directly imposes the parameter time-dynamics through the pdf p. The time-varying dynamics of the parameter are considered implicitly. On the other hand, the method introduced in this chapter has been developed to consider explicitly the time-varying dynamics of the parameters in the identification process. The prior knowledge of Bayes inference is based on the prior identification-step for each time step, such that $\theta^i(k)$ depends on the same time step k but previous identification iteration $\theta^{i-1}(k)$, where i the identification iteration-step, on the observation (model output) $z(k)$ and on the pdf p. A filtering process based on moving average with a pdf g is implemented to explicitly control the identified parameters time-dynamics during the parameter identification. This method enables us to separate the identification process from the time-smoothing process.

In section 5.2 the hybrid dynamical system modeling definitions are introduced. In section 5.3 the existing and novel parameter identification methods are detailed. Then in section 5.4 the selected application models are presented, in section 5.5 examples of parameters identifications are shown, and in section 5.6 driving behavior analysis and modeling are discussed.

5.2 Modeling definitions

The identified models are piecewise hybrid dynamical systems models with time-varying parameters. The usual hybrid dynamical system modeling framework [80] is described as

$$\begin{cases} y(k) = f^1\left(r^1(k)\right) + e^1(k) & \text{if} \quad \mu(k) = 1 \\ & \quad\vdots \\ y(k) = f^M\left(r^M(k)\right) + e^M(k) & \text{if} \quad \mu(k) = M \end{cases} \tag{5.1}$$

where r is the model input, a time-series vector composed of an external input time-series vector u and of the observed output time-series vector y, such that $r(k) = [s(k), u(k-1), ..., u(k-n_a), y(k-1), y(k-2), ..., y(k-n_b)]$ where $n_a \in \mathbb{N}$ represents the exogenous input order, and $n_b \in \mathbb{N}^*$ represents the autoregressive input order. f^m

is a set of functions with $m \in \{1, 2, ..., M\}$ the mode index number, $k \in \{1, 2, ..., K\}$ is the discrete time step, e^m is the modeling error, and $\mu \in \{1, 2, ..., M\}^K$ is the mode index vector.

In signal processing field, filtering is usually applied to state-space models. For using particle filtering in parameter identification, dynamical model parameters are usually considered as model states [118, 119]. Expressed as a state-space simulation model, mode equations of the hybrid dynamical system model (5.1) become

$$\left\{ \begin{array}{l} x^m(k) = h_p^m \left(x^m(k-1), r^m(k) \right) \\ \hat{y}(k) = h_o^m \left(x^m(k), r^m(k) \right) \end{array} \right. \tag{5.2}$$

where x is the state vector (here observable), r the simulation model input, $\hat{y}$ the simulation model output, h_p is a process function, h_o is an output function and $m = \mu(k)$ the mode index number.

The difference between the dynamical systems expressed by the equation (5.1) and the equation (5.2) is the formulation of the system state. In equation (5.1), the dynamical system is expressed as a transfer function, where the state of the system is implicitly expressed in the model output y, whereas in equation (5.2), the dynamical system is expressed as a state-space system, where the states are explicitly expressed by the state variable x. Thus state-space formulation enables the formulation of a larger set of models, including hidden states models. The formulation (5.1) is the most common for simple dynamical models such as the applications models of this article, but this formulation does not allow to clearly express the model parameters. Thus, the state-space formulation is introduced to explicitly represent how the identified parameters are included in the filtering problem. Moreover, this formulation allows notations compliance with the reference articles [118, 119]. In that way, equation (5.2) can be considered as a generalization of equation (5.1).

The parameter vector θ is then assimilated to a state of the state-space model (5.2) to be identified by the particle filter as

$$\left\{ \begin{array}{l} x^m(k) = h_p^m \left(x^m(k-1), \theta^m(k), r^m(k) \right) \\ \hat{y}(k) = h_o^m \left(x^m(k), \theta^m(k), r^m(k) \right) \end{array} \right. \tag{5.3}$$

where θ is the parameter vector extended as a model state.

Finally, expressed as a transfer function hybrid dynamical system model, the state-space simulation model (5.3) becomes

$$\begin{cases} y(k) = f^1\left(u^1(k), \theta_\mu(k)\right) + e^1(k) & \text{if } \mu(k) = 1 \\ & \vdots \\ y(k) = f^M\left(u^M(k), \theta_\mu(k)\right) + e^M(k) & \text{if } \mu(k) = M \end{cases} \tag{5.4}$$

where $\theta^m(k) \in \Theta^m$ is the parameter vector of the mode m at the time step k and Θ^m the parameter space of the mode m.

The concept of mode output occlusion is also introduced in this section. As described in Equation (5.4), the model output y is a composition of the modes m outputs over the time steps k depending on the value of $\mu(k)$. Thus at each time step k, $M-1$ mode outputs are not observable. These non-observable mode outputs are called occluded mode outputs. During mode output occlusion, time-varying parameters of the mode cannot be identified.

5.3 Parameter identification process

In this section, the method and the implementation of the identification process for time-varying parameters are detailed.

In numerous application cases, modeling is a rough approximation of the real measured data, and the reproduced situations are non-deterministic. Moreover the raw data used for model fitting is often known within an error margin. In those cases, as long as the parameter identification process error is lower than the modeling error or than the raw data error margin, an optimal parameters identification scheme will not bring any advantage over an metaheuristic scheme. Bayesians methods explicitely enable to select the density probability of the estimated parameters, and thus to know the error margin of the estimated parameter in case of identification convergence.

To be able to identify a wide range of hybrid system models, including non-differentiable nonlinear heterogeneous hybrid system models, a suboptimal nonparametric Bayesian method is selected. Parameters estimate (posterior) are calculated based on prior parameters estimates, on the parameters estimation pdf and on an observation. Parameters are estimated from the marginal distribution in (5.5) [122]. Nonparametric methods do not rely on a fixed functional form of the posterior, but instead create an approximate of the posterior state by a finite number of particles. Thus nonparametric methods converge uniformly to the correct posterior as the number of particles goes to infinity.

Bayes rule used for parameter identification is expressed as follows:

$$p\left(\theta(k)|z(1:k)\right) = \frac{p\left(z(k)|\theta(k)\right)p(\theta(k)|z(1:k-1))}{p\left(z(k)|z(1:k-1)\right)} \tag{5.5}$$

where $\theta(k) \in \mathbb{R}^{n_\theta}$ is the estimated time-varying parameter vector and $z(k) \in \mathbb{R}^{n_z}$ is the observation. n_θ and n_z are the parameters and observation dimensions in $\mathbb{N}^*$. $k \in \{1, 2, ..., K\}$ represents the current time and identification step.

To be able to identify several parameters per mode and to filter each individual parameter time-variation to its time dynamic, a novel implementation scheme of the Bayesian approach based on particle filtering combined with parameter time-smoothing algorithm is proposed in this article.

5.3.1 Particle filtering for parameter identification

The particle filter (PF), also called Sequential Monte-Carlo (SMC), is a nonparametric Bayesian approach, creating a recursive Bayesian filtering by Monte-Carlo type sampling [119], [79, 122–125]. The key idea in PF is to represent the posterior density by a set of random samples drawn from this posterior, to calculate associated weights considering an observation, and then to compute the new estimates based on these samples and weights. Markov assumption on the parameters is used in system identification to make the calculation tractable. Particle filters have been used to determine real-time nonlinear model parameters [118], hybrid system model (Piecewise ARX) constant parameters and modes estimation [90], and nonlinear non-hybrid (NARX) time-varying parameters from a predefined finite parameter set [120]. This literature uses a conventional implementation of the PF, and as far as we know, does not consider the parameter time-variation explicitly

In this paper the Sample Importance Resample (SIR) scheme is selected over the Sequential Importance Sampling (SIS) scheme, to avoid the degeneracy problem [119, 122]. Moreover no assumption is taken on the ergodicity of the parameter time-variation, thus usage of SIS is not possible. The importance density and the resampling algorithm must be carefully selected to avoid respectively a large variance in the particles weights and sample impoverishment. The main steps of the SIR algorithm are shown in Figure 5.1.

Standard particle filtering method for parameter identification is expressed as follow:

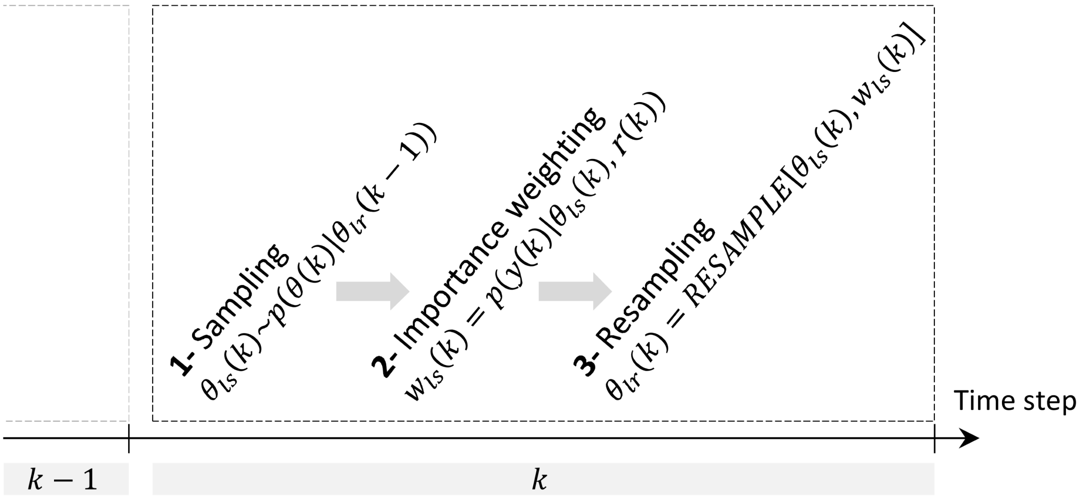

1- Sampling
$\theta_{ls}(k) \sim p(\theta(k)|\theta_{lr}(k-1))$
2- Importance weighting
$w_{ls}(k) = p(y(k)|\theta_{ls}(k), r(k))$
3- Resampling
$\theta_{lr}(k) = RESAMPLE[\theta_{ls}(k), w_{ls}(k)]$
Time step
$k-1$
k

Figure 5.1: Standard SIR particle filter identification process.

Algorithm 1 SIR Particle Filter

```
$\{\theta_{lr}(k)\}_{lr=1}^{Lr} = SIR\left[\{\theta_{lr}(k-1), r(k), \hat{y}(k)\}_{lr=1}^{Lr}\right]$
for $lr = 1 : Lr$ do
    Step 1: Sampling
    - Draw $\theta_{ls}(k) \sim p(\theta(k)|\theta_{lr}(k-1))$
end for
for $ls = 1 : Ls$ do
    Step 2: Importance weighting
    - Calculate $w_{ls}(k) = p(y(k)|\theta_{ls}(k), r(k))$
end for
- Calculate total weight: $w^{tot}(k) = \sum_{ls=1}^{Ls} w_l(k)$
for $ls = 1 : Ls$ do
    - Normalize: $w_{ls}(k) = (w^{tot}(tot))^{-1} * w_{ls}(k)$
end for
Step 3: Resampling, using Algorithm 2 in [122]
$\{\theta_{lr}(k)\}_{lr=1}^{Lr}$ =RESAMPLE$\left[\{\theta_{ls}(k), w_{ls}(k)\}_{ls=1}^{Ls}\right]$
```

In Algorithm 1, $lr \in \{1, 2, ..., Lr\}$ is the resampled particle index, $ls \in \{1, 2, ..., Ls\}$ is the sampled particle index, k is the discrete time step, $\theta \in \Theta$ the estimated parameter, w is the associated weight, r the model input, and y is the observation.

5.3.2 Smoothing algorithm

An algorithm of smoothing over time is implemented in the parameters identification process to avoid the effect of noise on the parameter identification, and to be able to identify parameters time-variation within specified dynamics. Parameters dynamics filtering can be done independently for each parameter, enabling dynamics decoupling of the parameters during the identification process. Smoothing can be done directly by solving the problem $p(\theta(k)|\hat{y}(1:K))_{K\in\mathbb{N}^*}, k \in \{1, 2, ..., K\}$, but time-smoothed particle filtering methods as presented in [125] tend to be complex to implement. Thus a simple moving average method has been adapted in this work.

Parameter estimates profile is generated based on the maximum likelihood estimate (see (5.8)) at each time steps. This profile is time-smoothened by using a moving average method (MA) weighted by the pdf g. The smoothened parameter estimate profile is then used to attribute smoothing weights to all the particles of the identification process based on the pdf p (see Figure 5.2).

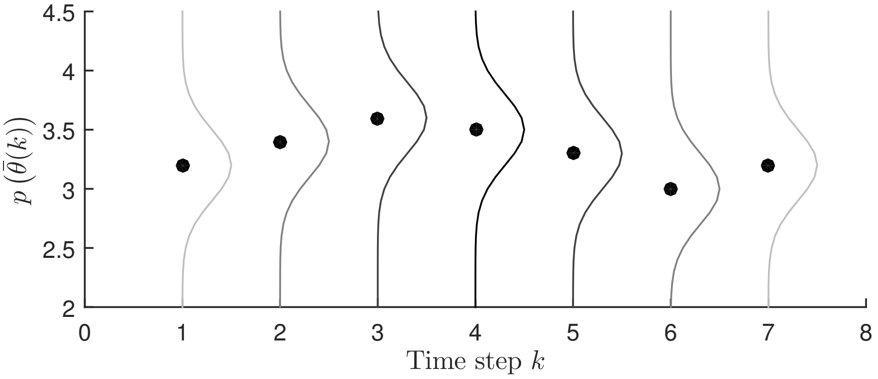


Figure 5.2: Example of Normal weighting distributions $p(\bar{\theta}(k))$ over the time steps k, used to define smoothening particles weights. Black points represent the smoothened parameter estimate $\bar{\theta}(k)$, curves represent the probability distribution $p(\bar{\theta}(k))$ over the possible particles values.

5.3.3 Algorithm initialization

In this section, two methods are proposed to initialize the parameter identification algorithm.

Case A: Data classification is known without a-priori knowledge on parameters.

The minimum a-priori knowledge required to initialize the parameter identification method is the modes partitioning of the hybrid system model. Numerous methods can be used to segment the data [121]. Once the mode separation is obtained, initial particles can be spread uniformly over the candidate parameters space.

Case B: Prior knowledge from Bayesian parameter identification approach.

Initial parameters and modes partition can be determined using a conventional method developed for the identification of hybrid systems parameters. The most suitable solution for nonlinear hybrid system models is the Bayesian online identification method [90]. This method provides good results for single parameter identification, and can also be used to provide a-priori knowledge for multiple parameter cases. Nevertheless, the problem proposed in [90] is not well posed for multiple parameters cases. In this algorithm, the particles weights would have to be determined based on a number of time steps at least equal to the number of identified parameters per mode. This initialization method is only suitable for cases with clear mode separation in the data.

Regarding the applications of this paper, initialization of the identification process based on real driving data is done using Case A.

5.3.4 Parameter identification scheme

In this section, the proposed parameter identification algorithm is described. In order to identify time-varying parameters of hybrid dynamical system models with parameter smoothing over time, the particle filter is implemented as an iterative process instead of a time dependent process (see Figure 5.3). Variables and indexes notations are detailed in Tables 5.1 and 5.2.

The pseudocode of the proposed parameter identification method is detailed in Algorithm 2, and the method is described as follows:

- **Initialization:** Particles representative of each parameter are initialized.
- **Step 1:** Sampling of the particles is done at each time step k using a standard particle sampling scheme. A particle set $\Xi = \{\theta_{ls}, w_{ls}\}_{ls \in \{1,2,\ldots,Ls\}}$ is generated for each parameter of each mode at each time step, based on:

  $$\theta^i_{ls}(k) \sim p\left(\theta^i(k) | \theta^{i-1}_{lr}(k)\right). \tag{5.6}$$

- **Step 2:** If the mode output is not occluded, the particles weights are calculated using

  $$w^i_{ls}(k) = p\left(y(k) | \theta^i_{ls}(k), r(k)\right), \tag{5.7}$$

  otherwise all particles representing a parameter have equal weights.

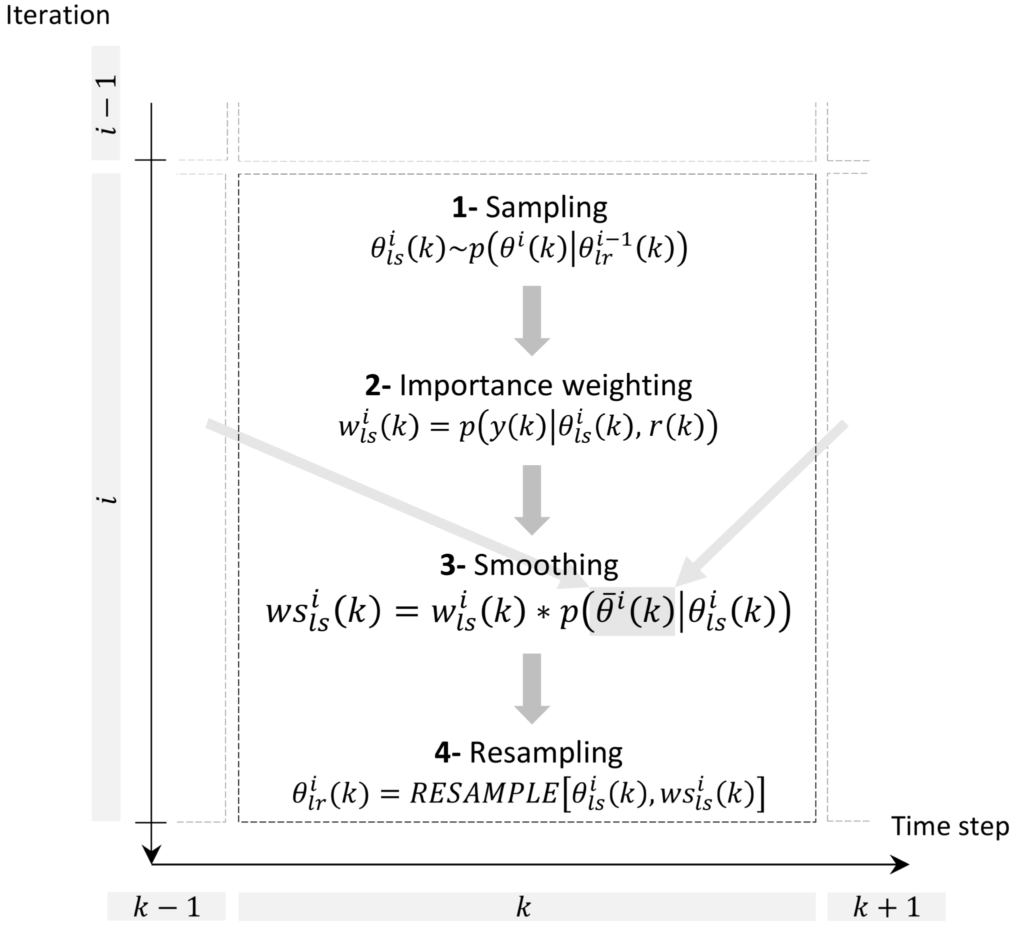

Iteration
$i-1$
i
1- Sampling
$\theta_{ls}^{i}(k)\sim p\left(\theta^{i}(k)\middle|\theta_{lr}^{i-1}(k)\right)$
2- Importance weighting
$w_{ls}^{i}(k)=p\left(y(k)\middle|\theta_{ls}^{i}(k),r(k)\right)$
3- Smoothing
$ws_{ls}^{i}(k)=w_{ls}^{i}(k)*p\left(\bar{\theta}^{i}(k)\middle|\theta_{ls}^{i}(k)\right)$
4- Resampling
$\theta_{lr}^{i}(k)=RESAMPLE\left[\theta_{ls}^{i}(k),ws_{ls}^{i}(k)\right]$
Time step
$k-1$
k
$k+1$

Figure 5.3: Novel parameters identification scheme, composed of an iterative SIR particle filter and a time-smoothing algorithm. $\bar{\theta}^i(k)$ represents the smooth parameter estimate at iteration i and discrete time step k. Definition of $\bar{\theta}^i(k)$ is given in Equation (5.9).

- **Step 3:** If the mode output is not occluded, the point estimate of $\theta^i_{ls}(k)$ is calculated for each parameter:

$$\hat{\theta}^i(k) = \underset{\theta^i_{ls}(k)\in\Xi}{\operatorname{argmax}} \left(w^i_{ls}(k)\right). \tag{5.8}$$

Otherwise $\hat{\theta}^i(k)$ is extrapolated from adjacent point estimates at non-occluded time steps.

Then smooth parameters estimates are calculated from the point estimates and the moving average weight function g_n as follows:

$$\bar{\theta}^i(k) = \sum_{j=1}^{K} \left[\hat{\theta}^i(j)\cdot g_n(j|k)\right]. \tag{5.9}$$

Smoothing weights are associated to each particles, and calculated based on the pdf p, the smooth point estimates and the particles values:

$$s_{ls}^{i}(k) = p\left(\bar{\theta}^{i}(k) | \theta_{ls}^{i}(k)\right). \tag{5.10}$$

Finally, particles weights are recalculated based on the particles weights (5.7) and the smoothing weights (5.10):

$$ws_{ls}^{i}(k) = w_{ls}^{i}(k) * s_{ls}^{i}(k). \tag{5.11}$$

- **Step 4:** Particles are resampled using a standard particle resampling scheme, based on the recalculated weights (5.11):

$$\theta_{lr}^{i}(k) = \text{RESAMPLE}\left[\theta_{ls}^{i}(k), w_{ls}^{i}(k)\right], \tag{5.12}$$

and i is iterated:

$$i = i + 1. \tag{5.13}$$

The algorithm goes to Step 1 if it did not reach the end criteria (stability of the modeling error e, Equation (5.4)).

Table 5.1: Identification scheme parameters and variables

θ	One identified parameter
θ_l	One identified particle
ν	All parameters stacked in a vector
y	Observation
r	Model input
μ	Mode index vector
w	Particle weight
s	Smoothing weight
ws	Total weight

Table 5.2: Identification scheme indexes

k	Discrete time step
i	Identification iteration step
m	Mode index number
n	Parameter number index
lr	Resampled particle number index
ls	Sampled particle number index

Algorithm 2 Time-varying parameters identification

```
$\Theta^i = \text{Ident\_Params}[\Theta^{i-1}, r, y]$
Step 1: Sampling
for $k = 1 : K$ (time) do
    for $m = 1 : M$ (mode) do
        for $n = 1 : N$ (parameters) do
            - Draw $\left\{\theta_{n,ls}^{m,i}(k)\right\}_{ls=1}^{Ls} \sim \left\{p\left(\theta_n^{m,i}(k)|\theta_{n,lr}^{m,i-1}(k)\right)\right\}_{lr=1}^{Lr}$
        end for
    end for
end for
Step 2: Importance weighting
for $k, m, ls$ (time, mode, sampled particles) do
    if $\mu(k) == m$ then
        - Calculate $w_{ls}^{m,i}(k) = p\left(y(k)|\theta_{ls}^{m,i}(k), r(k)\right)$
    else
        - $w_{ls}^{m,i}(k) = Ls^{-1}$
    end if
    - Normalize $w_{ls}^{m,i}(k)$ over $ls$
end for
Step 3: Smoothing
for $k, m$ (time, mode) do
    if $\mu(k) == m$ then
        - Point estimate index: $\hat{l}^m(k) = \underset{ls}{\text{argmax}}\left(w_{ls}^{m,i}(k)\right)$
    else
        - Extrapolate: $\hat{l}^m(k)\alpha\left\{\hat{l}^m(\mu_{k^-} == m), \hat{l}^m(\mu_{k^+} == m)\right\}$
    end if
    - Estimate: $\hat{\theta}^{m,i}(k) = \theta_{\hat{l}^m(k)}^{m,i}(k)$
end for
for $k, m, ls$ (time, mode, sampled particles) do
    - Smooth estimate: $\bar{\theta}_n^{m,i}(k) = \sum_{j=1}^{K}\left[\hat{\theta}_n^{m,i}(j) * g_n(j|k)\right]$
    for $ls = 1 : Ls$ (sampled particles) do
        - Smoothing weights: $s_{ls}^{m,i}(k) = p\left(\bar{\theta}^{m,i}(k)|\theta_{ls}^{m,i}(k)\right)$
    end for
    - Normalize $s_{ls}^{m,i}(k)$ over $l$
end for
for $k, m$ (time, mode) do
    for $ls = 1 : Ls$ (sampled particle) do
        - Final weight: $ws_{ls}^{m,i}(k) = w_{ls}^{m,i}(k) * s_{ls}^{m,i}(k)$
    end for
    - Normalize $ws_{ls}^{m,i}(k)$ over $ls$
end for
Step 4: Resampling, using Algorithm 2 in [122]
for $k, m$ (time, mode) do
    - $\left\{\theta_{lr}^{m,i}(k)\right\}_{lr=1}^{Lr} = \text{RESAMPLE}\left[\left\{\theta_{ls}^{m,i}(k), ws_{ls}^{m,i}(k)\right\}_{ls=1}^{Ls}\right]$
end for
```

With this method, the modes of the hybrid dynamical system model are identified separately. Thus the parameters of each mode are different spaces and can take similar values. In case of mode output occlusion, values of the parameters are extrapolated thanks to the integrated time-smoothing process. Extrapolation of the parameters could be improved by replacing the selected MA time-smoothing method with a more sophisticated method.

The proposed parameter identification scheme has a few limitations. The first limitation is the number of modes of the hybrid model. A high number of modes induces frequent output occlusion, and thus results in low parameter identification accuracy. The second limitation is the number of identified parameters. A high number of identified parameters implies strong time-smoothing as explained in section 5.3.6, and thus it reduces the added value of this method over a constant parameter identification method. Finally, very large problems are to avoid due to the high number of required particles, involving long calculation durations. A typical problem solved with this method would be composed of up to 15 parameters distributed in 3 modes.

5.3.5 Details on parameters tuning

The created model has two main tuning parameters: the particle filtering pdf p and the smoothing pdf g. If the pdf are normal distributions, the tuning parameters can be expressed as standard deviations. Be aware that normal distributions are not the best distributions for weighting due to the quick drop to zeros of the pdf envelope. It can be advised to use a weighting pdf p_{Step2} different from the sampling and resampling pdf p_{Step1} and p_{Step4}. For example, the weights w_{Step2} can be a negative power of the modeling error.

The particle filtering distribution p is used to sample, weight and resample. For filtering purpose, this pdf is usually associated to the signal noise. For parameter identification, this parameter is usually associated to the input signal noise, to the algorithm convergence speed and to the identified parameters precision. In the case of the developed parameter identification scheme, the particle filtering pdf p is not used to provide any knowledge from time-prior conditional probability. Thus the pdf p is only related to the algorithm convergence speed and to the identified parameters precision.

The smoothing distribution g proposed in the the developed parameter identification scheme is used to generate a filtered point estimate for each parameter at each discrete time step k. Thus g is the tuning parameter related to the input noise and to the parameters dynamics. Due to the creation process of the final weight ws, g is also related to the algorithm convergence speed. Strong smoothening will have a negative effect on the algorithm convergence speed.

5.3.6 Uniqueness of the solution

In this section, uniqueness of the identification equation system is verified. For a defined mode m, the identification problem can be formulated as

$$\left\{\begin{array}{rcl} y^m(1) & = & h(r^m(1), \theta^m(1)) + e_{h(1)} \\ \vdots & & \\ y^m(k) & = & h(r^m(k), \theta^m(k)) + e_{h(k)} \\ \vdots & & \\ y^m(K) & = & h(r^m(K), \theta^m(K)) + e_{h(K)} \end{array}\right. \tag{5.14}$$

where $k \in \{1, 2, ..., K\}$ is the time step, y^m is the mode output vector, h is a model, r^m is the mode input vector, θ^m is the mode parameters vector and e is the modeling error.

The identification problem is formulated as

$$\begin{array}{rcl} \theta & = & \underset{\theta}{\operatorname{argmin}} \sum_k \|e_{h(k)}\| \\ & = & \underset{\theta}{\operatorname{argmin}} \sum_k \|y(k) - h\left(r(k), \theta\right)\|. \end{array} \tag{5.15}$$

If the model parameters are considered as constant, $\forall k \in \{1, 2, ..., K\}$, $\theta(k) = \theta$, and theoretically, if $\|e_{h(k)}\| = 0$, as many system equations are the dimension of the parameter vector ν are required to get a unique solution. Thus hybrid systems identification processes usually use large sets of data to get a good guess of the model parameters values.

In the case of time-varying parameters identification, the assumption of constant parameter is not valid. Thus each equation of the system (5.14) is independent. The identification problems have $dim(\theta) - 1$ degree of freedom. A solution to avoid this issue is frequency decoupling. It can be considered that the identified parameters have very different natural frequencies, and thus lower frequencies parameters can be considered as constant. From this point of view, the problem is about solving a system with $dim(\theta) = 1$, and a unique solution exist. This frequency decoupling can be applied in Algorithm 2 through the g_n smoothing pdfs.

5.4 Selected application models

This section details the selected application models. The application goal of this research is to understand drivers variability and behavior modifications according to models parameters' evolution. Thus the selected application models must have been used for driving modeling and must have physically understandable parameters values.

Based on these considerations, PieceWise AutoRegressive eXogenous (PWARX) and Gipps microscopic traffic-flow [41] models have been selected. The PWARX model is a linear hybrid system model presented in section 2.4.1. It has been selected due to the fact that physical meaning could be attributed to model parameters in previous study [59]. The Gipps model has also been selected due to the fact that it provides comprehensive and physical parameters, that can be used for driver behavior identification. Gipps model is a nonlinear non-homogeneous hybrid model.

5.5 Parameters identification examples

In this section, validation of the developed parameters identification scheme is proposed. To validate the identification procedure, PWARX and Gipps model are used. This allows to cover linear and nonlinear cases, and single and multiple parameters cases. The particle filtering pdf $p = p_{Step1} = p_{Step4}$ is Gaussian with a mean equal to zero and a standard deviation σ_p, weighting of the particles is done based on the inverse of the square root of the modeling error, and the MA smoothing pdf g is Gaussian with a mean equal to zero and a standard deviation σ_g.

Parameter identification examples are done with at most two simultaneously identified time-varying parameters per mode, and two modes per model. As expalined in section 5.3.6, the number of parameters should be kept low in order to be able to identify high frequency time-variations. The number of modes can be higher, but a high number of modes implies frequent mode output occlusion, and thus reduces the parameters identification precision.

5.5.1 One identified parameter case, PWARX model

The test input, output and parameters are generated by the following system:

$$
\begin{aligned}
&f(x) = \begin{cases} \theta^1(k)^\top \begin{bmatrix} r(k) \\ 1 \end{bmatrix} & \text{if } \ \mu(k) = 1 \\ \theta^2(k)^\top \begin{bmatrix} r(k) \\ 1 \end{bmatrix} & \text{if } \ \mu(k) = 2 \end{cases} \\
&\begin{cases} \theta^1(k)^\top = \begin{bmatrix} 0.5 + \frac{sin(k*12)}{4} & 0.5 \end{bmatrix} \\ \theta^2(k)^\top = \begin{bmatrix} -1 + \frac{sin(k*6)}{6} & 2 \end{bmatrix} \end{cases} \\
&r(k) = u(k) = [-2.5 \ -2.49 \ \dots \ 2.5]
\end{aligned}
\tag{5.16}
$$

where $k \in \{1, 2, ..., K\}$ is the discrete time step, and μ is the mode index vector.

Only the first parameter of each mode is studied. This parameter identification example uses 10 particles per parameter, with a sampling coefficient of 10, $\sigma_p = 0.05$, and $\sigma_g = 2$. Initialization is done with random particles in the $[-2.5; 2.5]$ range. The results of Figure 5.4 are obtained in 5 identification iterations.

As shown in Figure 5.4, the developed parameter identification method can filter out noise in parameter values while preserve the parameters dynamics. It can be observed that the parameters are correctly identified. The identification time was 62 seconds on an Intel i5@3GHz computer with 8GB or RAM.

5.5.2 Multiple identified parameters case, PWARX model

In this section, results from the PWARX model with multiple simultaneous time-varying identification are shown. The problem definition is the same than in (5.16), with the following time-varying parameters

$$
\begin{cases} \theta^1(k)^\top = \begin{bmatrix} 1 + \frac{sin(k*12)}{4} & 0.5 + \frac{sin(k*1.2)}{8} \end{bmatrix} \\ \theta^2(k)^\top = \begin{bmatrix} -1 + \frac{sin(k*6)}{6} & 2 \end{bmatrix}. \end{cases}
\tag{5.17}
$$

This parameter identification example uses 10 particles per parameter, with a sampling coefficient of 10, $\sigma_p = 0.05$, $\sigma_g^1 = 2$ and $\sigma_g^2 = 20$. Initialization is done with random particles as shown in Figure 5.5. 10 iterations i are done.

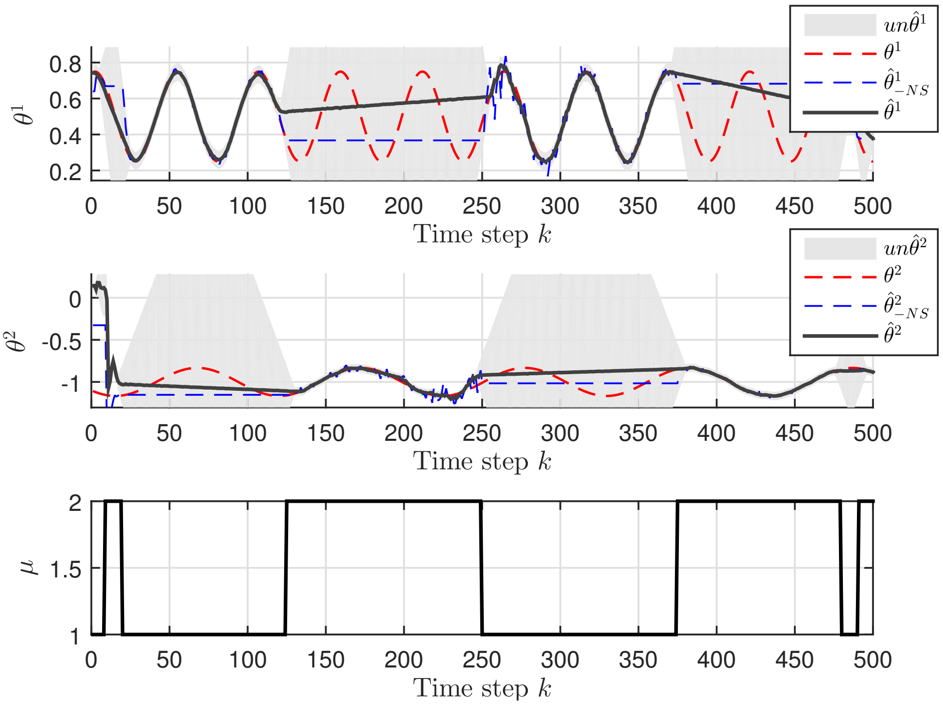


Figure 5.4: Parameter identification of a two modes one time-varying parameter per mode PWARX model.

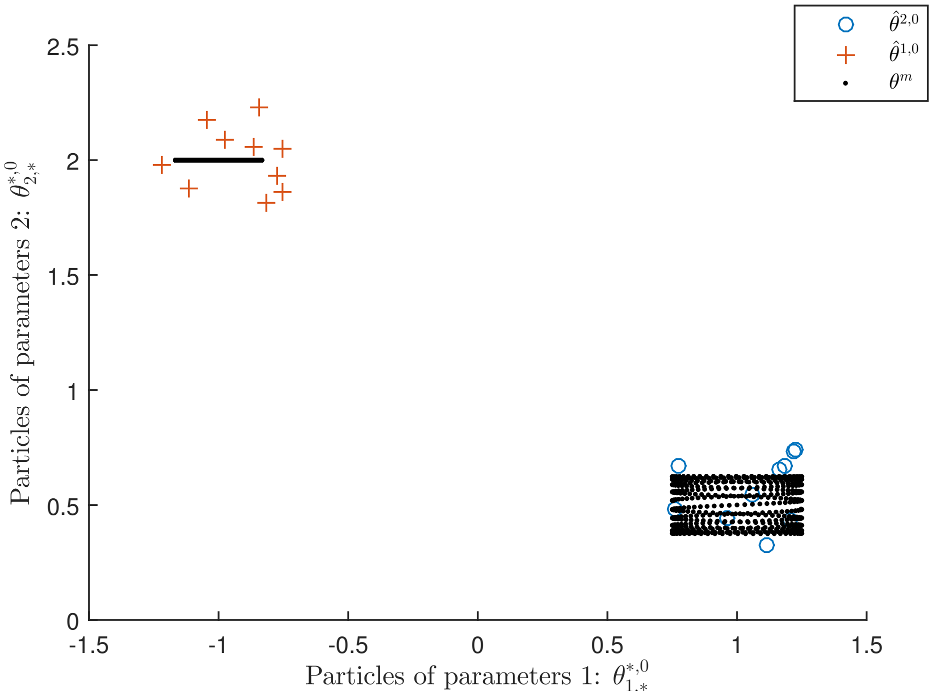


Figure 5.5: Initial particles for two modes two parameters per mode PWARX model. In orange crosses the initial particules for $m = 1$, in blue circles the initial particles for $m = 2$ and in black dots the true parameters values.

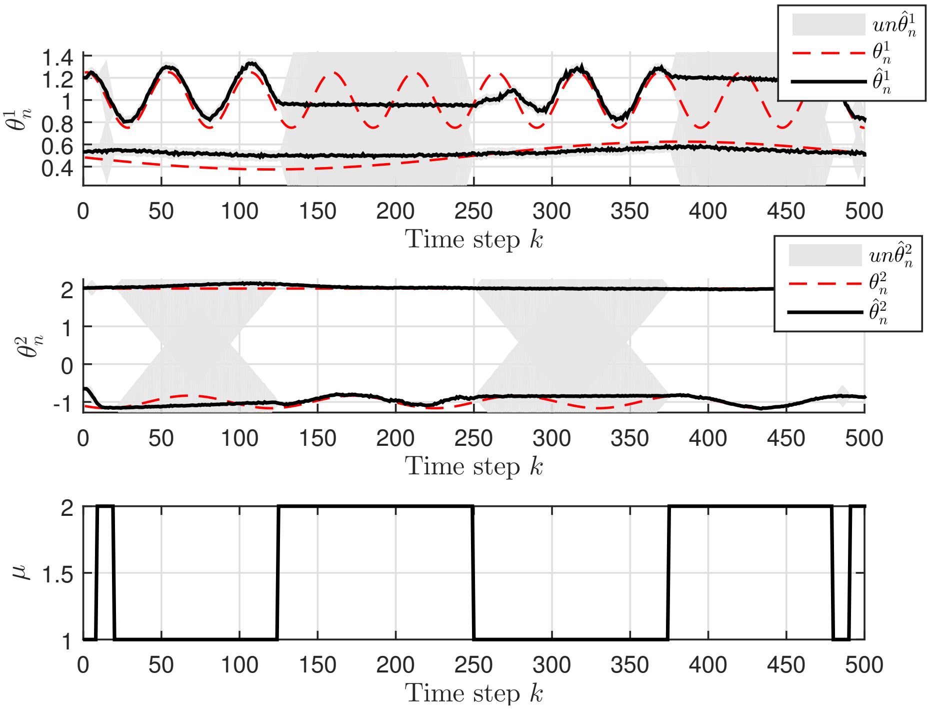


Figure 5.6: Parameter identification of a two modes two parameters per mode PWARX model.

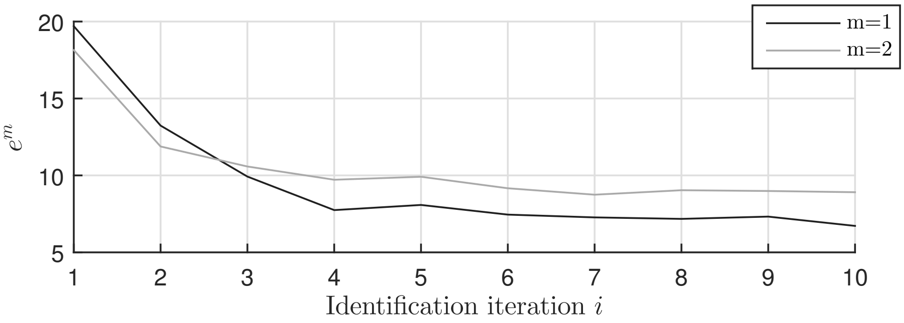


Figure 5.7: Parameter identification error of a two modes two identified parameters per mode PWARX model.

On Figures 5.4 and 5.6, $un\hat{\theta}$ represents the uncertanity of the parameter estimation, based on σ_p and on mode output occlusion. θ^m_n are the true parameter values of parameters number n of mode m, $\hat{\theta}^m_{-NS}$ the non-smoothened parameters estimate values and $\hat{\theta}^m_n$ are the final parameter estimates.

As shown in Figure 5.6, simultaneous identification of multiple time-varying parameters could be realized. When parameters cannot be identified, values of the parameters are extrapolated. Identification error can be observed, as the number of particles is kept low to enable fast identification. Nevertheless, values of time-varying parameters are closer to ideal than a constant parameter. In Figure 5.7, $e^m = \sum_{k[\mu(k)=m]} \|f^m(\theta(k)) - f^m(\hat{\theta}(k))\|_1$ represents the total modeling error per mode m. When identification error stabilizes, the identification process convergence is assumed. The modeling errors do not converge to zero due to the filtering and to the extrapolation processes. The identification time was 118 seconds on an Intel i5@3GHz computer with 8GB or RAM. This computation speed opens the method to online parameters identification.

5.5.3 One parameter case, nonlinear model

In this section, the efficiency of the model is assessed with the heterogeneous nonlinear car-following Gipps traffic flow model shown in section 1.2.2.1.

The identified parameter is b of the braking equation v_b, corresponding to the mode $m = 2$. Gipps model first runs with a known time-varying parameter $b(k)$ to generate a known output. Other Gipps model parameters are known and constant (see Table 5.3). The leading vehicle velocity and relative distance used for this example are extracted from the real-world measurement presented in 2.3.3. The velocity of the leading vehicle is showed in Figure 5.8.

Table 5.3: Gipps model parameters values.

Variable	v_0	s_0	τ_{reac}	a	b	bm
Value	20	6.5	0.3	1.7	$\theta(k)$	-3.2

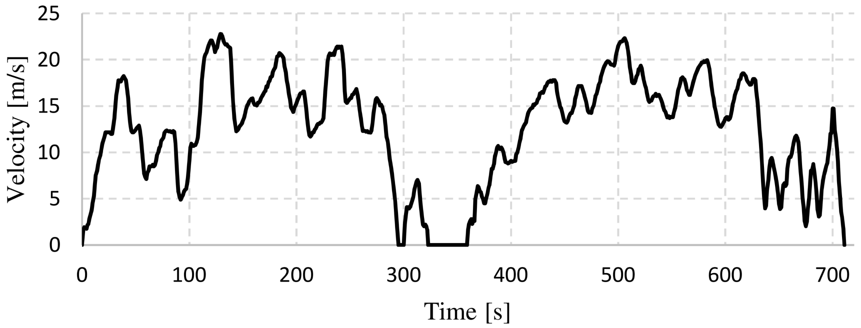


Figure 5.8: Leading vehicle velocity data used in the generation of the example output data and for the time-varying parameter identification.

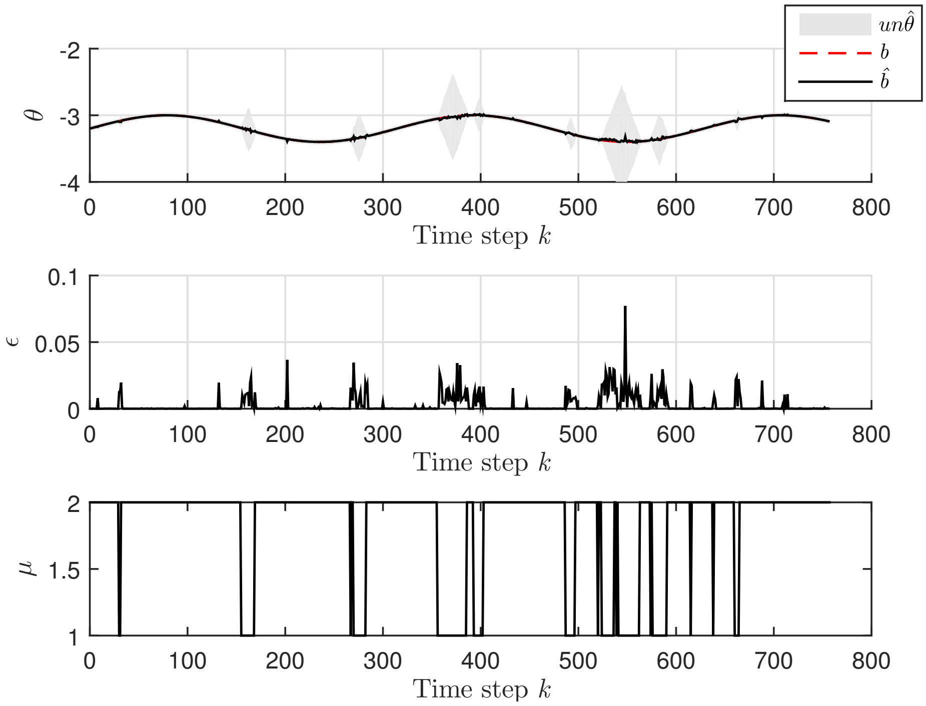


Figure 5.9: Low frequency parameter time-variation case.

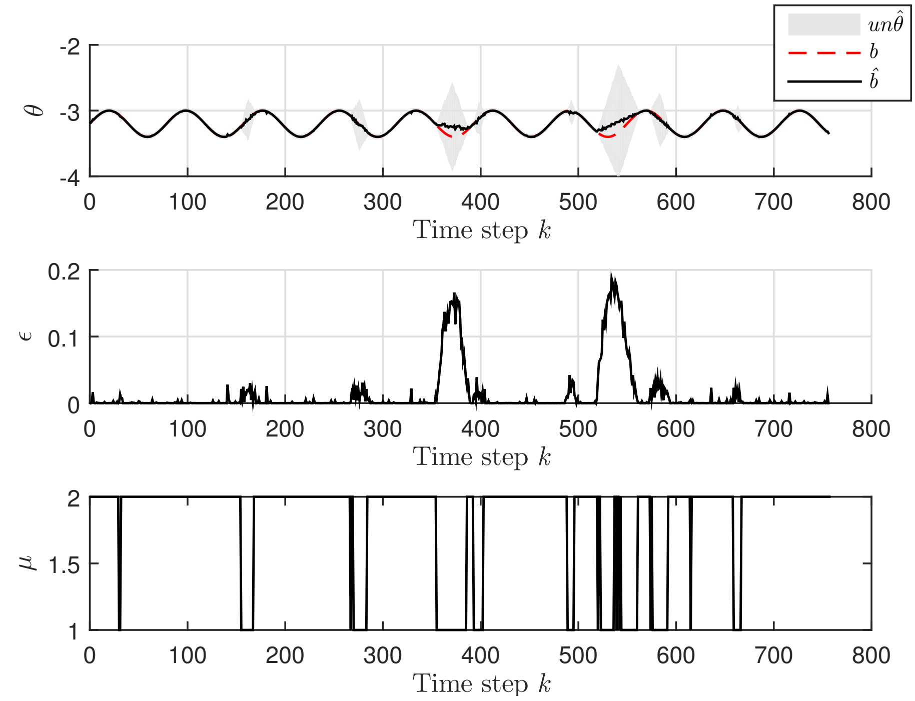


Figure 5.10: High frequency parameter time-variation case.

On Figures 5.9 and 5.10, $un\hat{\theta}$ represents the uncertainty of the parameter estimation, based on σ_p. b is the true parameter and $\hat{b}$ is the final parameter estimate.

Figure 5.9 and 5.10 show the identification of the time-varying parameter $b(k)$, with different time-variation frequencies. As described in table 1.1, the parameter b is in the braking mode, enabling parameter identification when $\mu(k) = 2$.

It can be observed n Figure 5.9 and Figure 5.10 that the parameter b is correctly identified. $\epsilon(k) = \|\theta(k) - \hat{\theta}(k)\|_1$ represents the time-varying parameter estimation error. This parameter identification error is low as long as $\mu(k) = 2$. Thus the proposed parameter identification method can be used to interpret driving behavior through model time-varying parameters. The identification time was 26 seconds on an Intel i5@3GHz computer with 8GB or RAM. This speed of execution opens this method to online parameter identification.

5.6 Driving behavior applications discussion

In this section, a discussion about the application for driver behavior analysis and modeling is proposed.

A large variety of driver modeling methods have been researched in the past years. Initial studies focused on the human psychophysical reactions [16], then an important highlight has been done on the creation of microscopic traffic flow models from data analysis [3, 34, 41, 107], and more recently a focus is done on the usage of machine learning methods [59–62, 80, 116]. While most of these models can have parameters attributed a physical meaning, understanding of the driver behavior from the point of view of the parameters is not common practice. Different aspects of the driving behavior could be investigated from the analysis of time-varying parameters: the driver consistency expressed by high frequency parameters variations or a statistical parameter variance, and the driver behavior modifications expressed by low frequency parameters variations or global parameters changes [11]. Online analysis of the parameters could also be considered for indirect sensing of the human state during the driving operation [59].

Once analysis of drivers is done, we believe that modeling of the human stochasticity could be done. This new class of modeling techniques could be implemented in future driver models for more realistic traffic flow simulations, for naturalistic traffic vehicles behavior prediction in autonomous cars, or for advanced driver-personalized driving assistance systems.

As a first example, identification of the deceleration parameter b of the Gipps model based on real-world measured data is proposed in Figures 5.11, 5.12, 5.13 and 5.14. Data used for the model identification has been extracted from the real-world measurement presented in 2.3.3. Two different drivers have been identified, these two drivers driving respectively with their interpretation of a soft and an aggressive driving style.

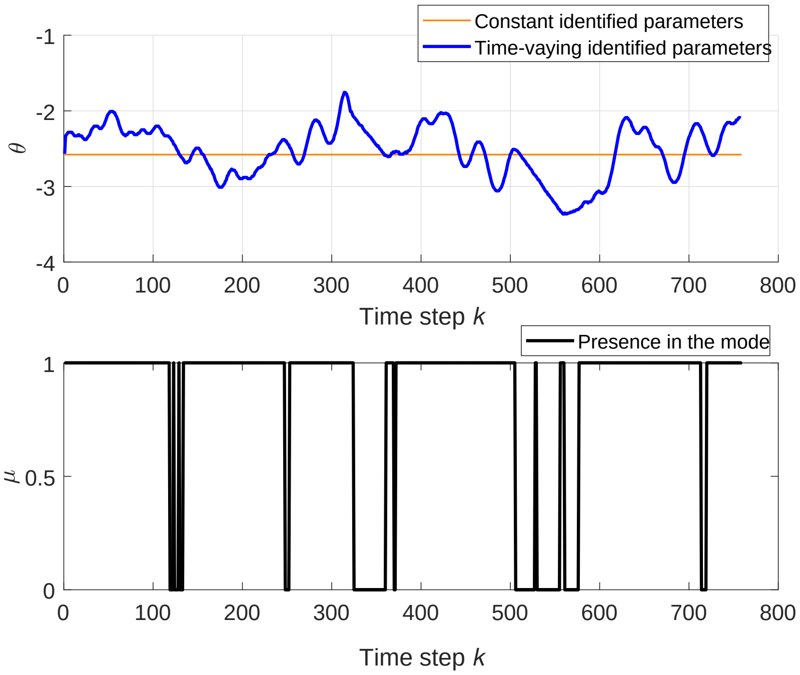


Figure 5.11: Driver A, soft driving style. θ represents estimated parameter and μ the engaged mode.

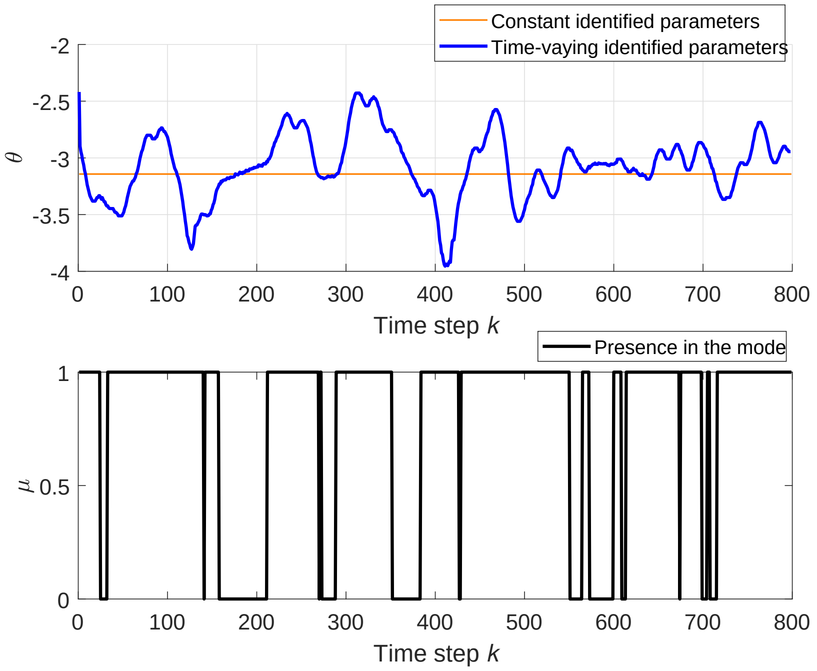


Figure 5.12: Driver A, aggressive driving style. θ represents estimated parameter and μ the engaged mode.

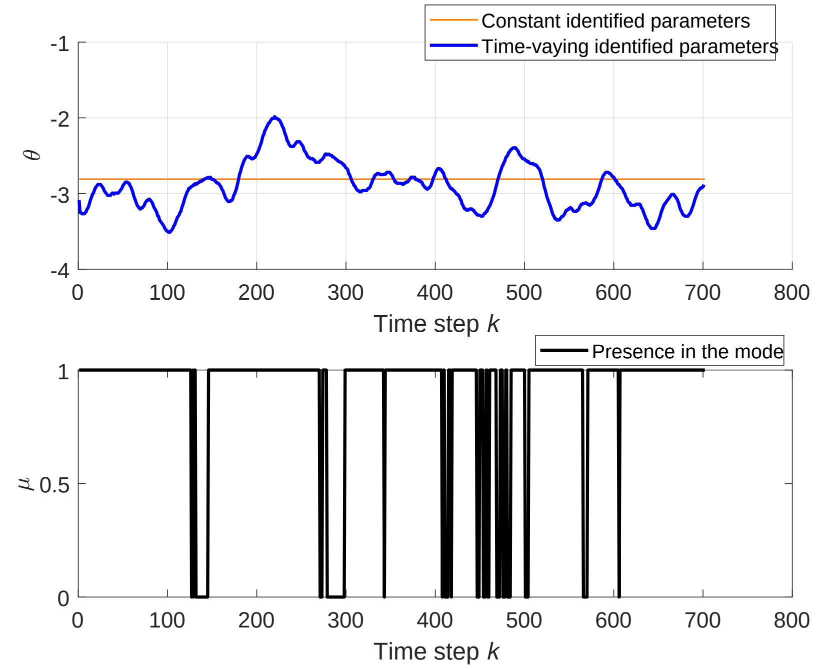


Figure 5.13: Driver B, soft driving style. θ represents estimated parameter and μ the engaged mode.

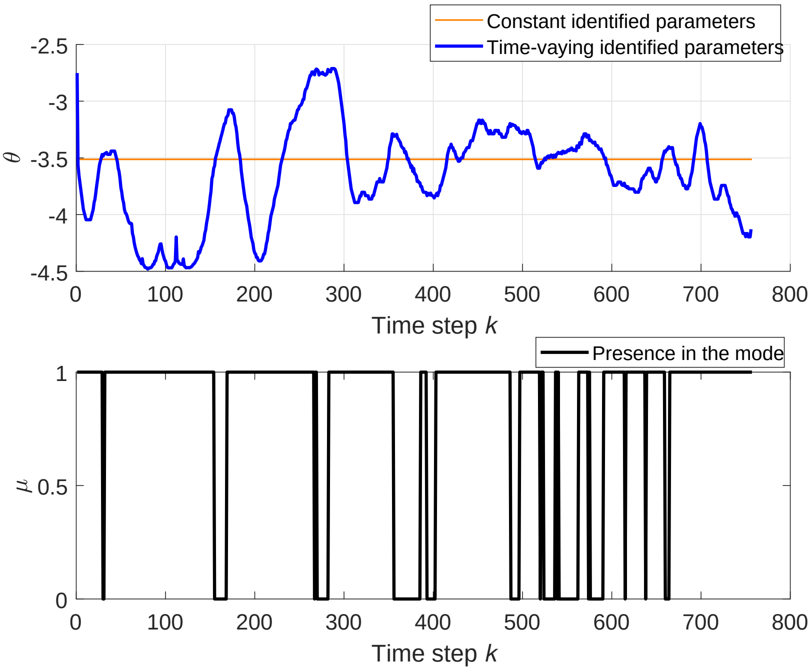


Figure 5.14: Driver B, aggressive driving style. θ represents estimated parameter and μ the engaged mode.

It can be observed in Figures 5.11, 5.12, 5.13 and 5.14 that the average aggressiveness of the driving style is well represented by the braking parameter of Gipps model. Moreover, the aggressive driving styles show stronger braking parameter variations than the soft driving styles. As a further development of such experiment, it would be interesting to limit the parameter identification to the situation of actual effectiveness. In this case the braking parameter b should be identified when negative following vehicle acceleration in the safe distance equation. By doing so, the model parameter is identified during usage, and its value can be used for driver behavior interpretation. This first example proves the applicability of this parameter identification method for driving behavior analysis.

Chapter 6

Conclusion

The analysis and the modeling of human driving behavior is a process involving proper understanding of the driver-vehicle dynamics, in relation to the environment information, the use and the creation of flexible modeling frameworks adapted to all types of drivers and able to reproduce non-linear time-dependent dynamics. All these aspects make human driving behavior modeling a complex and engaging topic. Once correctly understood, it becomes a powerful asset for applications such as traffic flow modeling, ADAS systems creations and all types of driving behaviors prediction applications.

In this research work, the understanding of the human driving behavior could be tackled by in-depth data measurement and analysis. Based on this data analysis, the modeling framework could be selected. Multi-mode dynamical models seemed to be the best compromise between behavior flexibility and dynamics reproduction abilities. In this model class, the PrARX model has been selected, with a specific input-output, model order and parameter identification setup. The PrARX model has been compared to classical car-following models, and it has been used for accurately reproducing behavior personalized energy consumption. The real-world simulation cases showed results with fuel consumption estimation errors from -0.06% to -3.8% depending on the driving style. These results proved the ability of the developed framework to model personalized driving dynamics.

These modeling works were followed by the desire to understand the human more in detail, and to reproduce the human driving behavior by non-averaging methods. To be able to observe driving consistency, also named behavior stochasticity, model parameters have been considered as time-varying and these time-variations were identified. Short-time variations could be interpreted as action-wise driving consistency and longer-time variations could be interpreted as general behavioral consistency. This ability to observe parameters variations over time became possible thanks to the creation of a novel system identification method based on a genetic Bayesian method called Sequential Monte-Carlo, or Particle Filtering.

Future works related to this study can lead to exciting application and publications. Related to the traffic flow research field, the developed car-following PrARX model architecture has been integrated together with a constant-in-time version of the developed model identification scheme. This identification and simulation model enables to reproduce driver personalized traffic flow behavior. Once validated, this model could provide a novel method for road infrastructure planning, city pollution due to vehicle's emissions estimation based on individual driver behavior. The time-varying parameters identification method could also be used on larger measurement data sets for driving behavior patters recognition and prediction. Finally, the combination of time-varying parameters implementation together with personalized driver models open the way for novel behavior prediction, for short-time scales as used in ADAS system, as well as for global traffic flow evolutions analysis. The application of these methods are wide and diverse, and insure a large diversity of possible future evolutions.

Acknowledgements

I would like to thank Professor Tatsuya SUZUKI for welcoming me in his laboratory for three years, for helping me to understand research culture and the art of scientific publication, and for wisely advising my research orientations.

I would like to share my sincere thanks to Assistant Professor Hiroyuki OKUDA, for his support throughout my doctoral research, by allowing me to understand the meaning of scientific research, by sharing his knowledge and passion for engineering, and for the numerous hours spent on experiment preparation and measurements. Thank you for your constant support on these three years in the University of Nagoya.

I wish to express my gratitude to Professor Tetsunori HARAGUCHI for welcoming me in the his research project and for sharing his experience, to Professor Hirofumi AOKI for enabling me to use his equipment, without which most of the data used in this research would not have been available, and to Professor Hiroyuki TAKASHIMA for his support on organisation and vehicle simulation.

I also would like to thank all the laboratory students, assistant researchers and laboratory assistants for their help, and for all the events shared within and outside of the laboratory, unique moments and talks that constitute the memorable experience I had in Nagoya.

Bibliography

[1] Stacy C. Davis ; Susan E. Williams; Robert G. Boundy. Transportation energy: Data book: Edition 35. Technical report, U.S. Department of Energy, 2016.

[2] Christopher J. Nash ; David J. Cole ; Robert S. Bigler. A review of human sensory dynamics for application to models of driver steering and speed control. *Biological Cybernetics*, Vol. 110(No. 2):pp. 91–116, June 2016.

[3] Mark Brackstone ; Mike McDonald. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 2(No. 4):pp. 181–196, December 1999.

[4] H.G. Pottering ; G. Gloser. Regulation (ec) no 715/2007 of the european parliament and of the council of 20 june 2007. Technical report, Official Journal of the European Union, June 2007.

[5] Office of Transportation and Air Quality. Light-duty vehicles and light-duty trucks: Clean fuel fleet exhaust emission standards. Technical report, United States Environmental Protection Agency, 2016.

[6] G. Maggetto ; J. Van Mierlo. Electric and electric hybrid vehicle technology: a survey. In *IEE Seminar on Electric, Hybrid and Fuel Cell Vehicles*, pages pp. 1–12, 2000.

[7] Alternative Fuels Data Center. Light-duty afv, hev, and diesel model offerings, by fuel type. Technical report, US Department of ENERGY, 2016.

[8] Caroline Brugier ; Laure de Servigny. Hybrid air: Dveloppement dun nouveau type de motorisation hybride essence. Technical report, ADEME, PSA Peugeot-Citroen, 2014.

[9] T. Wilhelem ; H. Okuda ; B. Levedahl ; T. Suzuki ; T. Haraguchi. Behavior personalized adaptive cruise control using probability-weighted arx model. In *22nd ITS World Congress*, October 2015.

[10] Thomas Wilhelem ; Hiroyuki Okuda ; Blaine Levedahl ; Tatsuya Suzuki. Energy consumption evaluation based on a personalized driver-vehicle model. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18(No. 6):pp. 1468 – 1477, October 2016.

[11] Thomas Wilhelem ; Hiroyuki Okuda ; Akihiko Kawashima ; Tatsuya Suzuki. Identification of time-varying parameters in gipps model for driving behavior analysis. In *IEEE/SMC Systems, Man, and Cybernetics Society*, October 2016.

[12] Thomas Wilhelem ; Hiroyuki Okuda ; Tatsuya Suzuki. Identification of time-varying parameters of hybrid dynamical system models and its application to driving behavior. *IEICE Trans. on Fundamentals*, Vol. 100(No. 10), October 2017.

[13] JI Elkind. A survey of the development of models for the human controller. *Guidance and Control II*, Vol. 13:pp. 623–643, 1964.

[14] David H. Weir ; A. V. Phatak. Model of human operator response to step transitions in controlled element dynamics. Technical Report Vol. 671, National Aeronautics and Space Administration (NASA), 1967.

[15] Sheldon Baron ; Dana S. Kruser ; Beverly Messick Huey. *Quantitative Modeling of Human Performance in Complex, Dynamic Systems.* National Academy Press, 1990.

[16] Charles C. MacAdam. Understanding and modeling the human driver. *Vehicle System Dynamics*, Vol. 40(No. 1-3):pp. 101–134, 2003.

[17] Wenshuo Wang ; Junqiang Xi ; Huiyan Chen. Modeling and recognizing driver behavior based on driving data: A survey. *Mathematical Problems in Engineering*, Vol. 2014, 2014.

[18] N. Rashevsky. Some remarks on the mathematical aspect of automobile driving. *Bulleting of Mathematical Biophysics*, Vol. 21:pp. 299–308, 1959.

[19] W.W. Wierwille ; G.A. Gagne ; J.R. Knight. An experimental study of human operator models and closed-loop analysis methods for high-speed automobile driving. *IEEE Transactions on Human Factors in Electronics*, Vol. 8(No. 3):pp. 187–201, 1967.

[20] D. T. Mc Ruer ; D. H Weir. Models for steering control of motor vehicles. *NASA Technical Documents*, 1969.

[21] N. Rashevsky. Automobile driving as psychophysical discrimination. *Bulleting of Mathematical Biophysics*, 1962.

[22] N. Rashevsky. Neglected factors in highway safety. *University of Michigan Mental Health Research Institute, Grant GM-12032-01*, 1966.

[23] Richard W. Pew ; Scheldon Baron. Perspectives on human performance modelling. *Automatica*, Vol. 19(No. 6):pp. 663–676, 1983.

[24] Kenneth R. Boff ; Lloyd Kaufman ; James P. Thomas, editor. *Handbook of Perception and Human Performance Sensory Process and Perception*, volume Vol. 1. Wiley-Interscience Publication, 1986.

[25] Kenneth R. Boff ; Lloyd Kaufman ; James P. Thomas, editor. *Handbook of Perception and Human Performance Cognitive Processes and Performance*, volume Vol. 2. Wiley-Interscience Publication, 1986.

[26] Malcolm L. Ritchie ; William K. McCoy ; William L. Wlede. A study of the relation between forward velocity and lateral acceleration in curves during normal driving. *Human Factors*, Vol. 10(No. 3):pp. 255–258, 1968.

[27] David N. Lee. A theory of visual control of braking based on information about time-to-collision. *Perception*, Vol. 5:pp. 437–459, 1976.

[28] M. C. Good. Sensitivity of driver-vehicle performance to vehicle characteristics revealed in open-loop tests. *International Journal of Vehicle Mechanics and Mobility*, Vol. 6(No. 4):pp. 245–277, 1977.

[29] T. Newcomb. Driver behaviour during braking. In *West Coast International Meeting and Exposition*, August 1981.

[30] Win Van Winsum. Speed choice and steering behavior in curve driving. *Human Factors*, Vol. 38(No. 3):pp. 434–441, 1996.

[31] William Levison. Interactive highway safety design model: Issues related to driver modeling. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1631:pp. 20–27, 1998.

[32] Yuji Hisaoka ; Masaki Yamamoto ; Akio Okada. Closed-loop analysis of vehicle behavior during braking in a turn. *JSAE Review*, Vol. 20(No. 4):pp. 537–542, October 1999.

[33] Daniel Fambro ; Rodger Koppa ; Dale Picha; Kay Fitzpatrick. Driver braking performance in stopping sight distance situations. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1701, 2000.

[34] Daiheng Ni. *Traffic flow theory.* Butterworth-Heinemann, 2016.

[35] Louis A. Pipes. An operational analysis of traffic dynamics. *Journal of Applied Physics*, Vol. 24:pp. 274–281, 1953.

[36] Johan Bengtsson. *Adaptive Cruise Control and Driver Modeling.* phdthesis, Lund Institute of Technology, November 2001.

[37] Dirk Helbing. Traffic and related self-driven many-particle systems. *REVIEWS OF MODERN PHYSICS*, Vol. 73(No. 4), October 2001.

[38] Mark Brackstone ; Ben Waterson ; Mike McDonald. Determinants of following headway in congested traffic. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 12(No. 2):pp. 131–142, March 2009.

[39] Robert E. Chandler ; Robert Herman ; Elliott W. Montroll. Traffic dynamics: Studies in car following. *Operations Research*, Vol. 6(No. 2.):pp. 165–184, April 1958.

[40] W. Helly. Simulation of bottlenecks in single lane trac flow. *Proceedings of the Symposium on Theory of Traffc Flow*, pages pp. 207–238, 1961.

[41] P.G. Gipps. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, Vol. 1(No. 2):pp. 105–111, April 1981.

[42] R. Wiedemann. Simulation des strassenverkehrsflusses. *Schriftenreihe des Instituts fr Verkehrswesen der Universitt Karlsruhe*, Vol. 8, 1974.

[43] Erwin R. Boer. Car following from the drivers perspective. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 2(No. 4):pp. 201–206, 1999.

[44] Wim Van Winsum. The human element in car following models. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 2(No. 4):pp. 207–211, 1999.

[45] Thomas A. Ranney. Psychological factors that influence car-following and car-following model development. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 2(No. 4), December 1999.

[46] Gnther Prokop. Modeling human vehicle driving by model predictive online optimization. *International Journal of Vehicle Mechanics and Mobility*, Vol.35(No. 1):pp. 19–53, August 2001.

[47] Charles C. MacAdam. Application of an optimal preview control for simulation of closed-loop automobile driving. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 11(No. 6):pp. 393–399, June 1981.

[48] Huei Peng. Evaluation of driver assistance systems - a human centeredapproach. In *Proceedings of 6th symposium on advanced vehicle control*, 2002.

[49] D. J. Cole ; A. J. Pick ; A. M. C. Odhams. Predictive and linear quadratic methods for potential application to modelling driver steering control. *International Journal of Vehicle Mechanics and Mobility*, Vol. 44(No. 3):pp. 259–284, February 2007.

[50] A J Pick ; D J Cole. Dynamic properties of a drivers arms holding a steering wheel. *SAGE Journal*, December 2007.

[51] D. M. Pool; M. Mulder ; M. M. Van Paassen ; J C. Van Der Vaart. Effects of peripheral visual and physical motion cues in roll-axis tracking tasks. *Journal of Guidance, Control, and Dynamics*, Vol. 31(No. 6):pp. 1608–1622, November 2008.

[52] Cristian Musardo ; Giorgio Rizzoni ; Yann Guezennec ; Benedetto Staccia. A-ecms: An adaptive algorithm for hybrid electric vehicle energy management. *European Journal of Control*, Vol. 11(No. 4-5):pp. 509–524, June 2005.

[53] Antonio Sciarretta ; Lino Guzzella. Control of hybrid electric vehicles. *IEEE Control Systems*, Vol. 27(No. 2):pp. 60–70, April 2007.

[54] Stephanie Stockar ; Vincenzo Marano ; Marcello Canova ; Giorgio Rizzoni ; Lino Guzzella. Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. *IEEE Transactions on Vehicular Technology*, Vol. 60(No. 7):pp. 2949–2962, September 2011.

[55] Andreas A. Malikopoulos. Supervisory power management control algorithms for hybrid electric vehicles: A survey. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15(No. 5):pp. 1869–1885, October 2014.

[56] Zhi Liang Tan ; Thomas Wilhelem ; Hiroyuki Okuda ; Blaine Levedahl ; Tatsuya Suzuki. Computation of energy-optimal velocity profile for electric vehicle considering slope of route. In *IEEE/SICE International Symposium on System Integration (SII)*, number pp. 472-478, December 2015.

[57] M. Bichi ; G. Ripaccioli ; S. Di Cairano ; D. Bernardini ; A. Bemporad ; I.V. Kolmanovsky. Stochastic model predictive control with driver behavior learning for improved powertrain control. In *49th IEEE Conference on Decision and Control (CDC)*, December 2010.

[58] Stefano Di Cairano ; Daniele Bernardini ; Alberto Bemporad ; Ilya V. Kolmanovsky. Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management. *IEEE Transactions on Control Systems Technology*, Vol. 22(No. 3):pp. 1018–1031, 2014.

[59] Hiroyuki Okuda ; Norimitsu Ikami ; Tatsuya Suzuki ; Yuichi Tazaki ; Kazuya Takeda. Modeling and analysis of driving behavior based on a probability-weighted arx model. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14(No. 1):pp. 98–112, March 2013.

[60] Shogo Sekizawa ; Shinkichi Inagaki ; Tatsuya Suzuki ; Soichiro Hayakawa ; Nuio Tsuchida ; Taishi Tsuda ; Hiroaki Fujinami. Modeling and recognition of driving behavior based on stochastic switched arx model. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8(No. 4):pp. 593–606, December 2007.

[61] Ryota Terada ; Hiroyuki Okuda ; Tatsuya Suzuki ; Kazuyoshi Isaji ; Naohiko Tsuru. Multi-scale driving behavior modeling using hierarchical pwarx model. In *IEEE International Conference on Intelligent Transportation Systems*, November 2010.

[62] Pongtep Angkititrakul ; Chiyomi Miyajima ; Kazuya Takeda. Modeling and adaptation of stochastic driver-behavior model with application to car following. In *IEEE Intelligent Vehicles Symposium*, pages pp. 814–819, June 2011.

[63] Jonas Sjberg ; Qinghua Zhang ; Lennart Ljung ; Albert Benveniste ; Bernard Delyon ; Pierre-Yves Glorennec ; Hkan Hjalmarsson ; Anatoli Juditsky. Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, Vol. 31(No. 12):pp. 1691–1724, December 1995.

[64] Kumpati S. Narendra ; Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, Vol. 1(No. 1):pp. 4–27, March 1990.

[65] Lin Zhao ; Witold Pawlus ; Hamid Reza Karimi ; Kjell G. Robbersmyr. Data-based modeling of vehicle crash using adaptive neural-fuzzy inference system. *IEEE/ASME Transactions on Mechatronics*, Vol. 19(No. 2):pp. 684–696, 2014.

[66] Biagio Ciuffo ; Vincenzo Punzo ; Marcello Montanino. Thirty years of gipps' car-following model. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2315:pp. 89–99, 2012.

[67] R. Eddie Wilson. An analysis of gipps' car-following model of highway traffic. *IMA Journal on Applied Mathematics*, March 2001.

[68] J. A. Ward R. E. Wilson. Car-following models: fifty years of linear stability analysis a mathematical perspective. *Transportation Planning and Technology*, Vol. 34(No. 1):pp. 3–18, December 2010.

[69] Ioanna Spyropoulou. Simulation using gipps' car-following model - an in-depth analysis. *Transportmetrica*, Vol. 3(No. 3):pp. 231–245, June 2007.

[70] K. V. R. Ravishankar ; Tom V. Mathew. Vehicle-type dependent car-following model for heterogeneous traffic conditions. *Journal of Transportation Engineering*, Vol. 137(No. 11):pp. 775–781, November 2011.

[71] Venkatesan Kanagaraj ; Gowri Asaithambi ; C. H. Naveen Kumar ; Karthik K. Srinivasan ; R. Sivanandan. Evaluation od different vehicles following models under mixed traffic conditions. In *2nd Conference of Transportation Research Group of India*, volume Vol. 104 of *Procedia - Social and Behavioral Sciences*, pages pp. 390–401, 2013.

[72] Johan Janson Olstam ; Andreas Tapani. Comparison of car-following models. *VTI meddelande 960A*, 2004.

[73] Bryan Higgs ; Montasir Abbas ; Alejandra Medina. Analysis of the wiedemann car following model over different speeds using naturalistic data. In *International Conference on Road Safety and Simulation*, number No. 3, September 2011.

[74] Rafal Goebel ; Ricardo G. Sanfelice ; Andrew R. Teel. Hybrid dynamical systems. *IEEE Control Systems*, Vol. 29(No. 2):pp. 28–93, March 2009.

[75] Rafal Goebel ; Ricardo G. Sanfelice ; Andrew R. Teel. *Hybrid Dynamical Systems.* Princeton University Press, 2012.

[76] Hui Ye ; A.N. Michel ; Ling Hou. Stability theory for hybrid dynamical systems. *IEEE Transactions on Automatic Control*, Vol. 43(No. 4):pp. 461–474, August 1998.

[77] T.A. Henzinger. The theory of hybrid automata. In *Eleventh Annual IEEE Symposium on Logic in Computer Science*, August 1996.

[78] Daniel Liberzon. *Switching in Systems and Control.* Springer, 2003.

[79] Sebastian Thrun ; Wolfram Burgard ; Dieter Fox. *Probability Robotics.* The MIT Press, 2006.

[80] Lennart Ljung. *System Identification: Theory for the user.* Prentice Hall, 1999.

[81] Aleksandar Lj. Juloski ; W. P. M. H. Heemels ; Giancarlo Ferrari-Trecate ; Ren Vidal ; Simone Paoletti ; J. H. G. Niessen. Comparison of four procedures for the identification of hybrid systems. In *Hybrid Systems: Computation and Control*, pages pp. 354–369, 2005.

[82] Hiroyuki Okuda ; Tatsuya Suzuki ; Ato Nakano ; Shinkichi Inagaki ; Soichiro Hayakawa. Multi-hierarchical modeling of driving behavior using dynamics-based mode segmentation. *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E92-A(No. 11):pp.2763–2771, November 2009.

[83] Toshikazu Akita ; Shinkichi Inagaki ; Tatsuya Suzuki ; Soichiro Hayakawa ; Nuio Tsuchida. Analysis of vehicle following behavior of human driver based on hybrid dynamical system model. In *IEEE International Conference on Control Applications*, November 2007.

[84] Koji Mikami ; Hiroyuki Okuda ; Shun Taguchi ; Yuichi Tazaki ; Tatsuya Suzuki. Model predictive assisting control of vehicle following task based on driver model. In *IEEE International Conference on Control Applications*, September 2010.

[85] Toshikazu Akita ; Tatsuya Suzuki ; Soichiro Hayakawa ; Shinkichi Inagaki. Analysis and synthesis of driving behavior based on mode segmentation. In *International Conference on Control, Automation and Systems*, October 2008.

[86] Giancarlo Ferrari-Trecate ; Marco Muselli ; Diego Liberati ; Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, Vol. 39(No. 2):pp. 205217, February 2003.

[87] A. Bemporad ; A. Garulli ; S. Paoletti ; A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, Vol. 50(No. 10):pp. 1567–1580, October 2005.

[88] J. Roll ; A. Bemporad ; L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, Vol. 40(No. 1):pp. 37–50, 2004.

[89] Y. Ma ; R. Vidal. Identification of deterministic switched arx systems via identification of algebraic varieties. *Hybrid Systems: Computation and Control*, Vol. 3431:pp. 449–465, 2005.

[90] A.Lj. Juloski ; S. Weiland ; W.P.M.H. Heemels. A bayesian approach to identification of hybrid systems. *IEEE Trans. On Automatic Control*, Vol. 50(No. 10):pp. 1520–1533, October 2005.

[91] Avi Rosenfeld ; Zevi Bareket ; Claudia V. Goldman ; Sarit Kraus ; David J. LeBlanc ; Omer Tsimhoni. Learning drivers behavior to improve the acceptance of adaptive cruise control. In *Proceedings of the Twenty-Fourth Innovative Appications of Artificial Intelligence Conference*, 2012.

[92] Sbastien Glaser ; Benoit Vanholme ; Sad Mammar ; Dominique Gruyer ; Lydie Nouvelire. Maneuver-based trajectory planning for highly autonomous vehicles

on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11(No. 3):pp. 589–606, May 2010.

[93] K. Imai ; N. Tsuru T. Wada ; S. Doi. Analysis of drivers behaviors in car following based on a performance index for approach and alienation. In *SAE Technical Paper*, January 2007.

[94] Martin Treiber ; Ansgar Hennecke ; Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, Vol. 62(No. 2):pp. 1805–1824, August 2008.

[95] Arne Kesting ; Martin Treiber ; Martin Schnhof ; Dirk Helbing. Extending adaptive cruise control to adaptive driving strategies. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2000:pp. 16–24, 2007.

[96] Norimitsu Ikami ; Hiroyuki Okuda ; Yuichi Tazaki ; Tatsuya Suzuki ; Kazuya Takeda. Online parameter estimation of driving behavior using probability-weighted arx models. In *IEEE International Conference on Intelligent Transportation Systems*, number No. 14, pages pp. 1874–1879, 2011.

[97] Marco Bottero ; Bruno Dalla Chiara ; Francesco Deflorio ; Gino Franco ; Ezio Spessa. Model-based approach for estimating energy used by traffic flows on motorways with its. *IET Intelligent Transport Systems*, Vol. 8(No. 7):pp. 598–607, November 2014.

[98] P.W.G. Newman ; J.R. Kenworthy. The transport energy trade-off: Fuel-efficient traffic versus fuel-efficient cities. *Transportation Research Part A: General*, Vol. 22(No. 3):pp. 163–174, May 1988.

[99] Aleksandar Kostikj ; Milan Kjosevski ; Ljupcho Kocarev. Impact of mixed traffic in urban environment with different percentage rates of adaptive stop&go cruise control equipped vehicles on the traffic flow, travel time, energy demand and emission. In *IEEE International Conference on Intelligent Transportation Systems*, number No. 18, September 2015.

[100] Markus Schori ; Thomas J. Boehme ; Benjamin Frank ; Bernhard P. Lampe. Optimal calibration of map-based energy management for plug-in parallel hybrid configurations: A hybrid optimal control approach. *IEEE Transactions on Vehicular Technology*, Vol. 64(No. 9):pp. 3897–3907, September 2015.

[101] H. Khayyam. Stochastic models of road geometry and wind condition for vehicle energy management and control. *IEEE Transactions on Vehicular Technology*, Vol. 62(No. 1):pp. 61–68, January 2013.

[102] Ravi Shankar ; James Marco. Method for estimating the energy consumption of electric vehicles and plug-in hybrid electric vehicles under real-world driving conditions. *IET Intelligent Transport Systems*, Vol. 7(No. 1):pp. 138–150, May 2013.

[103] Peter J. Pudney Philip G. Howlett. *Energy-Efficient Train Control*. Springer, 1995.

[104] Philip G. Howlett ; Peter J. Pudney. Local energy, minization in optimal tain control. *Automatica*, Vol. 45(No. 11), November 2009.

[105] Yeran Huang ; Lixing Yang ; Tao Tang ; Fang Cao ; Ziyou Gao. Saving energy and improving service quality: Bicriteria train scheduling in urban rail transit systems. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17(No. 12):pp. 3364–3379, May 2016.

[106] Jiateng Yin ; Tao Tanga ; Lixing Yang ; Ziyou Gao ; Bin Ran. Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. *Transportation Research Part B: Methodological*, Vol. 91:pp. 178–210, September 2016.

[107] S. Panwai ; H. Dia. Comparative evaluation of microscopic car-following behavior. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6(No. 3):pp. 314–325, September 2005.

[108] J. Ackermann ; J. Guldner ; W. Sienel ; R. Steinhauser ; V.I. Utkin. Linear and nonlinear controller design for robust automatic steering. *IEEE Transactions on Control Systems Technology*, Vol. 3(No. 1):pp. 132–143, August 1995.

[109] H. Aoki ; H. Yasuda ; Van Q. H. Nguyen. Perceptual risk estimate (pre): Development of an index of the longitudinal risk estimate. In *Proceedings of the 22nd International Technical Conference on the Enhanced Safety of Vehicles Conference*, 2011.

[110] Liu Yan ; Wang Dianhai. Minimum time headway model by using safety space headway. In *IEEE World Automation Congress*, July 2012.

[111] T.J. Ayres ; L. Li ; D. Schleuning ; D. Young. Preferred time-headway of highway drivers. In *IEEE Intelligent Transportation Systems*, August 2001.

[112] Donghee Yvette Wohn ; Cliff Lampe ; Rick Wash ; Nicole Ellison ; Jessica Vitak. The s in social network games: Initiating, maintaining, and enhancing relationships. In *Proceedings of the 44th Hawaii International Conference on System Sciences*, January 2011.

[113] Tai Stillwater ; Kenneth S. Kurani. Drivers discuss ecodriving feedback: Goal setting, framing, and anchoring motivate new behaviors. *Transportation Research Part F: Traffic Psychology and Behaviour*, Vol. 19:pp. 85–96, July 2013.

[114] Lily Elefteriadou. *An introduction to traffic flow theory.* Springer, 2014.

[115] A. Schadschneider. The nagel-schreckenberg model revisited. *The European Physical Journal B*, pages pp.573–582, 1999.

[116] G.S. Aoude ; V.R. Desaraju ; L.H. Stephens ; J.P. How. Driver behavior classification at intersections and validation on large naturalistic data set. *IEEE Trans. On Intel. Trans. Systems*, Vol. 13(No. 2):pp. 724–76, June 2012.

[117] P.J. Antsaklis. A brief introduction to the theory and applications of hybrid systems. *Proc. of the IEEE Special Issue on Hybrid Systems: Theory and Applications*, Vol. 88(No. 7), August 2002.

[118] G. Chowdhary ; R. Jategaonkar. Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. *Aerospace Science and Technology*, Vol. 14:pp. 106–117, 2010.

[119] C. Andrieu ; A. Doucet ; S. S. Singh ; V. B. Tadic. Particle method for change detection, system identification, and control. *Proceedings of the IEEE*, Vol. 92(No. 3):pp. 423–438, 2004.

[120] A. Hartmann ; S. Vinga ; J. M. Lemos. Hybrid identification of time-varying parameter with particle filtering and expectation maximization. In *IEEE 21st Mediterranean Conf. on Control & Automation*, pages pp. 884–889, 2013.

[121] S. Paoletti ; A. Lj. Juloski ; G. Ferrari-Trecate ; R. Vidal. Identification of hybrid systems: A tutorial. *European Journal of Control*, Vol. 13(No. 2-3):pp. 242–260, 2007.

[122] M.S. Arulampalam ; S. Maskell ; N. Gordon ; T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing*, Vol. 50(No. 2):pp. 174–188, February 2002.

[123] P.M. Djuric ; J. H. Kotecha ; J. Zhang ; Y. Huang ; T. Ghirmai ; M.F. Bugallo ; J. Miguez. Particle filtering: A review of the theory and how it can be used for solving problems in wireless communication. *IEEE Signal Process Magazine*, Vol. 20(No. 5):pp. 19–38, October 2003.

[124] C. Andrieu ; N. de Freitas ; A. Doucet; M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, Vol. 50(No. 1):pp. 5–43, January 2003.

[125] A.M. Johansen A. Doucet. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, Vol. 12(No. 3):pp. 656–704, December 2008.

Appendix: Time-varying parameters identification Matlab code

```matlab
function particles_ident = fct_variating_ident(model, data, particles_ident_ini, particles_new)
%fct_variating_ident Time-varying parameter identification with multi-mode handling.
%
% Input:
% - model: information about the model and the identification algorithm
% - data: data used for parameters identification
% - particles_ident_ini: initial particules population, if existing
% - particles_new : parameters related to the generated particle population
%
% it: iteration number
% k: time step number
% i: mode number
% p: parameter number
% l: particle number
%
% particles_ident(k,it) is the output structure including the identified parameters
%
% Thomas WILHELEM 2016

%Parameters
sigma_e = particles_new.sigma_e;
sigma_filt = particles_new.sigma_filter;
nb_modes = model.s;
nb_param = particles_ident_ini(1).nb_parameters;
```

```
nb_resample = particles_new.nb_resample; %Number of resampled particles
nb_sample = nb_resample * particles_new(1).sample_mult; % Number of sampled particles
k_max = size(data.y(1,:),2); %Number of time steps
it_max = model.varying_max_it; %Number of identification iterations

%% Initial particles based on particles_ident_ini
if model.initialization_J == 0
disp('**Warning** Juloski initialization not used')
end

for k=1:k_max %on time
for i = 1:nb_modes %on modes
for p = 1:nb_param %on parameters (line not required)
%Copy particles of each parameter
if model.initialization_J == 1 % If use initialization result
particles_ident(k,1).mode(i).resample(p,:) = particles_ident_ini(k).mode(i).resample(p,:);
else %If not use initialization results
% Put the same particles over all the time steps
particles_ident(k,1).mode(i).resample(p,:) = particles_ident_ini(1).mode(i).init_theoEx(p,:);
end
end
end
end

%Matrix of presence in the mode
in_mode = zeros(nb_modes, k_max); %initialize
in_mode(:,1) = ones(nb_modes, 1); %low border
in_mode(:,k_max) = ones(nb_modes, 1); %high border

for k=1:k_max %on time
for i = 1:nb_modes %on modes
if model.initialization_J == 1 % If use initialization result
if particles_ident_ini(k).mu == i
in_mode(i,k) = 1;
end
else
if data.mode(k) == i
in_mode(i,k) = 1;
```

```
end
end
end
end

%Initial values
it = 1; %Iteration number
multiWaitbar( 'CloseAll' );
multiWaitbar( 'Identification in progress: Time varying', 0 );
tic

%% Main loop
while it <= it_max % Iteration number based end condition
it = it+1; %Iteration number increment

%Progress bar
multiWaitbar( 'Identification in progress: Time varying', 'Value', (it-2) / it_max );
multiWaitbar( '1- Sampling new particles', 'Value', 0 ,'Color', 'b' );
multiWaitbar( '2- Importance weighting', 'Value', 0 ,'Color', 'b' );
multiWaitbar( '3- Smoothing', 'Value', 0 ,'Color', 'b' );
multiWaitbar( '4- Resampling the particles', 'Value', 0 ,'Color', 'b' );

% 1- Sample particles for all modes
for k = 1:k_max %on time step

multiWaitbar( '1- Sampling new particles', 'Value', k/k_max );

for i = 1:nb_modes %on mode
for p = 1:nb_param %on parameter
ls = 1; %Sampled particle index
for l = 1:nb_resample %on resampled particle

particles_ident(k,it).mode(i).sample(p,ls:(ls-1+particles_new(1).sample_mult)) =...
random('Normal', particles_ident(k,it-1).mode(i).resample(p,l), sigma_e, ...
1, particles_new(1).sample_mult);

ls = ls + particles_new(1).sample_mult;
```

```

end
end
end
end

multiWaitbar( 'Identification in progress: Time varying', 'Value', (it-2 +0.25 ) / it_max);
multiWaitbar( '2- Importance weighting', 'Value', 0 );

% 2- Importance weighting
for k = 1:k_max %on time step
multiWaitbar( '2- Importance weighting', 'Value', k/k_max );

for i = 1:nb_modes %on mode

% If in the mode
if particles_ident_ini(k).mu == i
esr = model.esr; %Use more data to regidify the system
for k_eq = max(1,k-esr*ceil(nb_param/2)):min(k_max, k+esr*ceil(nb_param/2))
%particles_ident(k,it).mode(i).w is a row vector
particles_ident(k,it).mode(i).w_k_eq(k_eq).w = ...
fct_weight(data, data.x(:,k_eq), data.y(1,k_eq), ...
particles_ident(k,it).mode(i).sample, sigma_e);

%Normalize
particles_ident(k,it).mode(i).w_k_eq(k_eq).w_nf = ...
sum(particles_ident(k,it).mode(i).w_k_eq(k_eq).w); %normalization factor
particles_ident(k,it).mode(i).w_k_eq(k_eq).w_norm = ...
particles_ident(k,it).mode(i).w_k_eq(k_eq).w ...
/ particles_ident(k,it).mode(i).w_k_eq(k_eq).w_nf;
end

%Sum the weights
particles_ident(k,it).mode(i).w = ...
particles_ident(k,it).mode(i).w_k_eq(max(1,k-esr*ceil(nb_param/2))).w;
for k_eq = max(1,k-esr*ceil(nb_param/2))+1 : min(k_max, k+esr*ceil(nb_param/2))
particles_ident(k,it).mode(i).w = particles_ident(k,it).mode(i).w + ...
particles_ident(k,it).mode(i).w_k_eq(k_eq).w;
```

```
end

%Normalize
particles_ident(k,it).mode(i).w_nf = ...
sum(particles_ident(k,it).mode(i).w); %normalization factor
particles_ident(k,it).mode(i).w_norm = ...
particles_ident(k,it).mode(i).w ...
/ particles_ident(k,it).mode(i).w_nf;


else % if not in mode
%Same weight for all the particles
particles_ident(k,it).mode(i).w_norm = ones(1,nb_sample)/nb_sample;

end
end
end

multiWaitbar( 'Identification in progress: Time varying', 'Value', (it-2 +0.5 )/it_max);


% 3- Smoothing -- Support update based on system dynamics

iw2_max_est = zeros(nb_param, k_max, nb_modes);

%Distribution max value
for k = 1:k_max %on time step
for i = 1:nb_modes %on mode

% If in the mode
if in_mode(i,k) == 1
%Pick up the max estimate
particles_ident(k,it).mode(i).iw2_max_est = ...
particles_ident(k,it).mode(i).sample(:,...
argmax(particles_ident(k,it).mode(i).w_norm));

else % if not in mode
%Linearly extrapolate on both temporal sides
% in_mode(mode,time_step)

km = argmax(in_mode(i,1:k-1).*[1:k-1]); %Last value with mode availability
```

```matlab
kp = k + argmax(in_mode(i,k+1:k_max)./[k+1:k_max]); %Next value with mode availability

particles_ident(k,it).mode(i).iw2_max_est = ...
( particles_ident(km,it).mode(i).sample(:,argmax(particles_ident(km,it).mode(i).w_norm))*(kp-k) ...
+ particles_ident(kp,it).mode(i).sample(:,argmax(particles_ident(kp,it).mode(i).w_norm))*(k-km) ...
) / (kp-km);

end

%Copy the structure to a matrix for ease of use
% iw2_max_est(nb_param, k_max, nb_modes)
iw2_max_est(:,k,i) = particles_ident(k,it).mode(i).iw2_max_est;

end
end


multiWaitbar( '3- Smoothing', 'Value', 0 );

for k = 2:k_max-1

multiWaitbar( '3- Smoothing', 'Value', k/k_max*3/4 );

% Weighting distribution over time steps values, for each parameter
for p=1:nb_param
theo_dist_filter_k(p).dist = makedist('Normal', 'mu', k, 'sigma', sigma_filt(p));
values_dist_filter_k(p,:) = pdf(theo_dist_filter_k(p).dist, [1:k-1 k+1:k_max]');
values_dist_filter_k_norm(p,:) = values_dist_filter_k(p,:)/sum(values_dist_filter_k(p,:));
end

for i = 1:nb_modes

% Average max estimate of each parameter around step time k
%avg_max_est_filter_k is a column vector
avg_max_est_filter_k = sum( values_dist_filter_k_norm .* ...
squeeze(iw2_max_est(:,[1:k-1 k+1:k_max],i)) , 2) ./ ...
sum( values_dist_filter_k_norm , 2 );

for p=1:nb_param

% Distribution over this average max estimate around step k
theo_dist_update_k = makedist('Normal', 'mu', avg_max_est_filter_k(p), 'sigma', sigma_e);

values_dist_update_k(:,k,i,p) = ...
pdf(theo_dist_update_k, [particles_ident(k,it).mode(i).sample(p,:)]');
values_dist_update_k_norm(:,k,i,p) = values_dist_update_k(:,k,i,p) / ...
sum(values_dist_update_k(:,k,i,p));

end

end
clear theo_dist_filter_k values_dist_filter_k values_dist_filter_k_norm
end

clear iw2_max_est

%Sum over the parameters to have one weight by particle set
for k = 2:k_max-1
multiWaitbar( '3- Smoothing', 'Value', k/k_max/4 + 3/4 );
for i = 1:nb_modes

for l = 1:nb_sample
particles_ident(k,it).mode(i).w2(l) = sum( values_dist_update_k_norm(l,k,i,:) );
end

particles_ident(k,it).mode(i).w2_norm = ...
particles_ident(k,it).mode(i).w2 / ...
sum( particles_ident(k,it).mode(i).w2 );

end
end

% Adjust the weights
```

```
for k = 2:k_max-1
for i = 1:nb_modes

particles_ident(k,it).mode(i).w_adj = ...
particles_ident(k,it).mode(i).w_norm .* ...
particles_ident(k,it).mode(i).w2_norm;

particles_ident(k,it).mode(i).w_adj_norm = ...
particles_ident(k,it).mode(i).w_adj / ...
sum( particles_ident(k,it).mode(i).w_adj );
end
end

% For the extreme k values, no filtering

k = 1;
for i = 1:nb_modes
particles_ident(k,it).mode(i).w_adj_norm = ...
particles_ident(k,it).mode(i).w_norm;
end

k = k_max;
for i = 1:nb_modes
particles_ident(k,it).mode(i).w_adj_norm = ...
particles_ident(k,it).mode(i).w_norm;
end


multiWaitbar( '3- Smoothing', 'Value', 1 );
multiWaitbar( 'Identification in progress: Time varying', 'Value',(it-2 +0.75 ) / it_max);
multiWaitbar( '4- Resampling the particles', 'Value', 0 );


% 4- Resampling of the particles
for k = 1:k_max
multiWaitbar( '4- Resampling the particles', 'Value', k/k_max );
for i = 1:nb_modes
[~,rI] = sort( particles_ident(k,it).mode(i).w_adj_norm ); %Get the order of the elements
rI_flip = flip(rI); %To have the higher weights first
particles_ident(k,it).mode(i).resample = ...
particles_ident(k,it).mode(i).sample(:,rI_flip(1:nb_resample));
```

```

clear rI
clear rI_flip

%Max estimate
particles_ident(k,it).mode(i).estimate = ...
particles_ident(k,it).mode(i).resample(:,1);

%Error on the max estimate
if in_mode(i,k) == 1 % particles_ident_ini(k).mu == i
[ ~, particles_ident(k,it).mode(i).error_k , ~ ] = ...
fct_weight(data, data.x(:,k), data.y(1,k), particles_ident(k,it).mode(i).estimate, sigma_e);
else

particles_ident(k,it).mode(i).error_k = 0;

end
end
end

particles_ident(1,it).error = 0;

for i = 1:nb_modes %On the modes

particles_ident(1,it).mode(i).error_i = 0; %Error in the modes for each iteration

for k = 1:k_max %on the time step

particles_ident(1,it).mode(i).error_i = ...
particles_ident(1,it).mode(i).error_i + ...
abs(particles_ident(k,it).mode(i).error_k);

end

particles_ident(1,it).error = particles_ident(1,it).error + ...
particles_ident(1,it).mode(i).error_i;
end
end

toc

multiWaitbar( '4- Resampling the particles', 'Close' );
multiWaitbar( '3- Smoothing', 'Close' );
multiWaitbar( '2- Importance weighting', 'Close' );
multiWaitbar( '1- Sampling new particles', 'Close' );
multiWaitbar( 'Identification in progress: Time varying', 'Close' );

% Generation of the error bars based on the particles_new.sigma_e
for i = 1:nb_modes %On the modes
for k = 2:k_max-1 %on the time step
if in_mode(i,k) == 1
% If in the mode

particles_ident(k,end).e_bar(i) = sigma_e;

else %If not in the mode

km = k - argmax(in_mode(i,1:k-1).*[1:k-1]); %Last value with mode availability
kp = argmax(in_mode(i,k+1:k_max)./[k+1:k_max]); %Next value with mode availability

particles_ident(k,end).e_bar(i) = min(km, kp) * sigma_e;

end
end
particles_ident(1,end).e_bar(i) = particles_new.sigma_e;
particles_ident(k_max,end).e_bar(i) = particles_new.sigma_e;
end
```

```
function [output, error, y_calc] = fct_weight(data, x_data, y_data, theta, sigma_e)
%fct_weight Calculate the weight of the function based on a normal distrib of sigma sigma_el
%theta(:,mode_nb)
%output(1,nb_sample)

size_theta = size(theta,2); %Number of particles

% Calculate model output
for l = 1:size_theta
y_calc(l) = fct_PWARX_modeNb(x_data, theta(:,l), 1);
end

```

```
% Difference data to calculation
error = y_data - y_calc;

c=0.5;
%% Output
output = (1./(abs(error)+1)).^c;
```

```
function y = fct_PWARX_modeNb(x, theta, mode_nb)
%fct_PWARX Calculates the output of a PWARX model based on
    the mode number
%Several inputs, one output
% x is row vectors
% theta is a parameter matrix, with as many rows as modes

phi = [x ; 1];

y = sum(theta(:,mode_nb)' * phi);
```