Control Engineering Practice 21 (2013) 807–817

Contents lists available at SciVerse ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A model identification scheme for driver-following dynamics in road traffic

Xiaoliang Ma [a,*,1], Magnus Jansson [b,2]

[a] Traffic and Logistics (ToL)/Centre for Traffic Research (CTR), School of Architecture and the Built Environment, Royal Institute of Technology (KTH), Teknikringen 72, SE-100 44 Stockholm, Sweden

[b] Signal Processing, School of Electrical Engineering, Royal Institute of Technology (KTH), Osquldas väg 10, SE-100 44 Stockholm, Sweden

ARTICLE INFO

Article history:
Received 4 April 2012
Accepted 8 February 2013
Available online 22 March 2013

Keywords:
Driver-following behavior
Model identification and learning
Extended Kalman filter
Recursive least square method
Traffic simulation
Advanced driving assistance systems

ABSTRACT

The driver-following, or car-following, model is one of the most fundamental driver behavior models that are applied in intelligent transport applications. Its fidelity determines the applicability of microscopic traffic simulators, where the model is often implemented to mimic real traffic. Meanwhile, the behavioral model is fundamental to the development of advanced driving assistance systems (ADAS). This paper develops a dynamic model identification approach based on iterative usage of the extended Kalman Filtering (EKF) algorithm. Among other things, this allows to carry out model identification using a rather general optimization objective on the whole physical states of the following vehicle. In particular, the method is established on the basis of the equivalence between the Kalman filter and the recursive least squares (RLS) method in a specific context of parameter identification. To illustrate the method, two car-following models are studied in numerical experiments using real car-following data. The method has shown advantages in replication and prediction of vehicle dynamics in car-following over the conventional approaches. It has also the potential to be further extended for building tactical driving controllers in intelligent transportation applications.

1. Introduction

With the continuous development of worldwide economy, the demands on road transportation have experienced a skyrocketing increase. To satisfy the requirements from social economy, safety and environment aspects, efficient traffic control and management become of vital importance. In particular, modern information technology (IT) has promoted the fast development of Intelligent Transportation Systems (ITS) as a multidisciplinary area with focuses on incorporating advanced computing and IT technologies into vehicle-based driving support and infrastructure-based control systems.

Traffic simulation models have been evolving as major tools for solving various application problems in road transport engineering. Among them, microscopic traffic simulator describe objects, or agents, in traffic systems (e.g., driver vehicle units, pedestrians, traffic signals etc.) and their interactions in very much details. Since its booming time in the early 1990s, micro-simulation has become more and more sophisticated and gained rocketing popularity not only in the traditional transport planning sector but also in traffic safety research, environmental studies, and in the development of advanced driver warning and assistant systems (ADWAS).

1.1. Driver behavioral models

Human behavior plays a key role in most applications of modern transportation engineering. The driver model is one of the most fundamental parts of a microscopic traffic simulator and directly determines the authenticity of the simulation. In addition, the technical development of vehicle-based ITS systems (e.g., Minoiu, Netto, Mammar, & Lusetti, 2009; Moon, Moon, & Yi, 2009; Wilmink, Kunder, & van Arem, 2007) needs understanding of actions of real drivers on a tactical level, which makes analysis and modeling of behavioral response of drivers even more needed than ever.

In driver behavior, car following is one of the fundamental phenomena, which reflects the longitudinal decisions of drivers when they follow a vehicle and try to adapt to the speed of or a driver-specific distance headway from the leading vehicle. Research of car-following models was launched in 1950s when basic traffic flow theory was developed by a group of mathematicians. There were several types of models which had later been extensively studied. The linear model developed by Helly (1959) takes the mathematical form as follows:

$$a_n(t+T_n) = C_1\Delta v_n(t) + C_2(\Delta x_n(t) - D_n(t+T_n))$$
$$D_n(t+T_n) = \alpha + \beta v_n(t) + \gamma a_n(t) \quad (1)$$

* Corresponding author. Tel.: +46 87908426.
E-mail address: liang@kth.se (X. Ma).

[1] X. Ma is a senior researcher in traffic control and ITS.

[2] M. Jansson is a senior researcher in signal processing and control techniques.

0967-0661/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.conengprac.2013.02.007

where $x(t)$, $v(t)$ and $a(t)$ are the position (of the front), speed, and acceleration of the vehicles. The indices n and $n-1$ refer to the following and the leading vehicle, respectively. Further, $\Delta v_n(t) = v_{n-1}(t)-v_n(t)$ and $\Delta x_n(t) = x_{n-1}(t)-x_n(t)-L_{n-1}$ are speed difference and space between the two cars at time t. $D_n(t)$ is the desired distance that the driver of the following vehicle wants to keep. T is the driver reaction time and L_{n-1} is the length of the leading vehicle. This model reflects the underlying behavioral relations and is essentially a linear regression between the acceleration of the following vehicle and those independent variables. The GM-type models (Chandler, Herman, & Montroll, 1958; Gazis, Herman, & Rothery, 1961), also named GHR models by later modelers, were originally developed in a research laboratory of General Motors (GM) Corporation during the 1950s, and it is probably the most well-known and extensively studied car-following model form. An early general form of the GM model is expressed as a nonlinear equation with a delay as follows:

$$a_n(t+T_n) = \alpha \frac{v_n(t+T_n)^l}{[x_{n-1}(t)-x_n(t)-L_{n-1}]^m}[v_{n-1}(t)-v_n(t)] \tag{2}$$

where all the variables are defined with the same physical meaning as those in the Helly's model. These two models belong to a common category where the relation between driver actions and physical variables is described using a mathematical equation. Later researchers further developed those model forms based on this idea. For example, Yang (1997) and Ahmed (1999) extended GM models to more general forms where acceleration and deceleration are no longer symmetric and parameters in different behavior regimes are estimated separately. Sultan, Brackstone, and McDonald (2004) further extended the model with incorporation of acceleration information of both leading and following vehicles, i.e.,

$$a_n(t+T_n) = \alpha \frac{v_n(t+T_n)^l}{\Delta x_n(t)^m}[v_{n-1}(t)-v_n(t)] + \beta_1 a_{n-1}(t) + \beta_2 a_n(t) \tag{3}$$

In a study on the dynamics of the GM-type model, Addison and Low (1998) proposed to add a nonlinear term on the distance headway to the Eq. (2), i.e.,

$$a_n(t+T_n) = \alpha \frac{v_n(t+T_n)^l}{\Delta x_n(t)^m}[v_{n-1}(t)-v_n(t)] + \beta(\Delta x_n(t)-D_n)^3 \tag{4}$$

where D_n represents the desired distance headway that the follower attempts to achieve. Instead of only achieving velocity matchup in the traditional GM form, the model considers the realization of a safe distance headway between consequent cars.

Besides the type of mathematical models, there are several other approaches worth of mentioning. For example, Chakroborty and Kikuchi (1999) applied a fuzzy rule-based model to represent the knowledge base of drivers; and Leutzbach and Wiedemann (1986) classified behavioral regimes using psycho-physical thresholds and modeled behavior in each regime with a simple function. A comprehensive review of car-following models can be found in Brackstone and McDonald (1999).

1.2. Model identification problem

Identification of driver behavior models such as car-following based on real behavioral measurement is essential for high fidelity traffic models as well as ITS applications. A conventional approach for estimating car-following models is to minimize the sum of the squared deviations between the model outputs and real acceleration values. Mathematically, the estimation approach can be represented by

$$\min_{\boldsymbol{\theta}} \sum_i \sum_t \epsilon_n^i(t)^2 = \min_{\boldsymbol{\theta}} \sum_i \sum_t (\tilde{a}_n^i(t) - f(\hat{\mathbf{x}}_n^i(t), \hat{\mathbf{x}}_{n-1}^i(t), \boldsymbol{\theta})^2 \tag{5}$$

where $\tilde{a}_n^i$ is the real acceleration of the following vehicle n for the data sequence i and $\hat{\mathbf{x}}_n(t)$ is the physical state of vehicle n at time t; $\boldsymbol{\theta}$ is the parameter set and $f(\cdot)$ is the car-following model.

This optimization problem is usually solved by numerical methods, such as gradient search or conjugate gradient methods since Eq. (5) formulates a nonlinear objective. Several previous studies (e.g., Ma & Andréasson, 2006) applied the calibration strategy and implemented the Newton-type search methods in finding the best-fitting parameters for the GM-type car-following models. A variation of the calibration approach above is to use a scaled absolute deviation of performance measure; i.e.,

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{\sum_i \sum_t |\tilde{p}_n^i(t) - g(\hat{\mathbf{x}}_n^i(t), \hat{\mathbf{x}}_{n-1}^i(t), \boldsymbol{\theta})|}{\sum_i \sum_t |\tilde{p}_n^i(t)|} \tag{6}$$

where $\tilde{p}(t)$ is the observed performance measure whereas $g(\cdot)$ is the estimated performance measure based on model outputs. This objective function was adopted in a previous calibration study on GPS based car-following data collected on experiment tracks (Ranjitkar, Nakatsuji, & Kawamura, 2005), and a Genetic Algorithm (GA) based optimization tool has been used to overcome the difficulty of trapping into local minima. However, simple GA algorithm is not as omnipotent as expected in searching a global optimal solution (Hwang & He, 2006). One question posed by many previous model identification studies is that the observed inputs have been used in the model identification or parameter search procedure. The question is whether the parameters estimated can reflect the dynamics in the training data or not. If not, is there an alternative way to realize that?

Punzo and Simonelli (2005) reformulated the estimation problem as a general least squares (GLS) problem based on different physical variables including acceleration, speed, trajectory etc. This formulation includes the vehicle dynamics but they did not explain how the problem could be generally solved. Another question is which performance measure shall be adopted in the general optimization function. For example, will calibration based on acceleration output errors lead to good results in speed and trajectory fitness. Punzo and Simonelli (2005) have explicitly shown their hypothesis in their study that inter-vehicle space (trajectory) is the most reliable performance measure. Although reasonable, their model calibration and analysis were conducted using GPS data, in which trajectory is the only direct measurement. When, for example, additional speed data can be directly measured, it might be not enough to simply use trajectory data.

1.3. Research objective

Adaptive signal processing approaches are known to be useful in the driver-following research (e.g., behavioral models and adaptive cruise control (ACC)). In the early work (Ma & Andréasson, 2005), a linear Kalman smoother was applied to estimate car-following states observed in real traffic. Bifulco, Pariota, Simonelli, and Pace (2011) recently developed, also based on Kalman smoother, a data fusion technique to estimate or track real-time car-following information collected from different sensors. Besides data estimation, Hoogendoorn, Ossen, and Schreuder (2007) applied an Unscented Particle Filter (UPF) to understand the hidden driver adaptivity in the driving tasks. The study focused on identification of temporal adaptivity of car-following parameters, including driver reaction time, by applying the filter once on each car-following time-series. The proposed approach is however limited in giving reliable estimation of driver following parameters by one-time filtering. Meanwhile, the analytical principle is not fully explored (e.g., optimization objective function).

In ACC development, Simonelli, Bifulco, Martinis, and Punzo (2009) proposed to train human-like controller using artificial

neural networks (ANNs) and real car-following data. The conventional learning method is used to train the ANN controller. Adaptive filtering approaches, indeed, have some relations with recursive least square (RLS) method that has been frequently used to identify parameters for nonlinear models or train controllers (e.g., Bouchard, 2001). This study therefore explores the possibility to recursively apply adaptive filtering approaches, especially Kalman filter (KF), and estimate parameters of car-following dynamics that can be applied in microscopic traffic simulation and in-vehicle ITS applications. The method may indicate machine learning techniques that can be extended to train ACC controller designed by not only mathematical models but also neural networks or fuzzy rule-based systems.

To study general driver behavior and identify parameters of different car-following models, an advanced instrumented vehicle was used to collect behavioral data on Swedish roads. The collected data was smoothed and classified in driver characteristics analysis in the former study (Ma & Andréasson, 2007). The remainder of this article will first explain, step by step, the KF-based method developed for model parameter estimation under the driver-following context. Numerical experiments are then conducted in the case studies of different car-following models; the last section summarizes the contribution of the paper and points out future research perspective in this topic.

2. Methodology

2.1. Overview of Kalman filters

This section briefly reviews the Kalman filtering theory. In different fields of science and engineering, there is a common problem to obtain an optimal state estimator for a linear state-space model as follows:

$$\mathbf{x}(t+1)=\mathbf{F}(t)\cdot\mathbf{x}(t)+\mathbf{D}(t)\cdot\mathbf{u}(t)+\mathbf{G}(t)\cdot\mathbf{e}(t)$$
$$\mathbf{y}(t)=\mathbf{H}(t)\cdot\mathbf{x}(t)+\mathbf{v}(t)+\mathbf{m}(t), \tag{7}$$

where $\mathbf{x}(t)$ is the real state at time t; $\mathbf{y}(t)$ is the measurement; $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are the control and measurement inputs; $\mathbf{e}(t)$ and $\mathbf{m}(t)$ are white noise for state and observation equations, respectively. The conventional Kalman filter (Kalman, 1960) described in Table 1 has been a standard approach since its invention. The algorithm aims at minimizing the following least squares loss function (Grewal & Andrews, 1993)

$$\mathbb{E}[(\mathbf{x}(t)-\hat{\mathbf{x}}(t|t))^T\mathbf{M}(\mathbf{x}(t)-\hat{\mathbf{x}}(t|t))] \tag{8}$$

where $\mathbf{M}$ is any symmetric nonnegative definite matrix and $\hat{\mathbf{x}}(t|t)=\mathbb{E}[\mathbf{x}(t)|\mathbf{y}(0)\ldots\mathbf{y}(t)]$ is the state estimate at time t. In the Kalman filter algorithm, $\mathbf{P}(t|t-1)$ and $\mathbf{P}(t|t)$ are the covariance matrices of a prior and posterior error on $\mathbf{x}(t)$ respectively, and $\mathbf{R}_e(t)$ and $\mathbf{R}_m(t)$ are the covariance matrices of the white noise processes in the plant and measurement model respectively (the system is assumed nonstationary in the formulation above).

Table 1
The conventional discrete-time Kalman filter algorithm.

State update (prediction):
$\hat{\mathbf{x}}(t|t-1)=\mathbf{F}(t)\hat{\mathbf{x}}(t-1|t-1)+\mathbf{D}(t)\mathbf{u}(t-1)$
$\mathbf{P}(t|t-1)=\mathbf{F}(t)\mathbf{P}(t-1|t-1)\mathbf{F}(t)^T+\mathbf{G}(t)\mathbf{R}_e(t)\mathbf{G}(t)^T$

Measurement update (correction):
$\Gamma_t=\mathbf{P}(t|t-1)\mathbf{H}(t)^T[\mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}(t)^T+\mathbf{R}_m(t)]^{-1}$
$\hat{\mathbf{x}}(t|t)=\hat{\mathbf{x}}(t|t-1)+\Gamma_t(\mathbf{y}(t)-\mathbf{H}(t)\hat{\mathbf{x}}(t|t-1)-\mathbf{v}(t))$
$\mathbf{P}(t|t)=(I-\Gamma_t\mathbf{H}(t))\mathbf{P}(t|t-1)$

However, nonlinear dynamics is often involved in real-life state-space systems in the form as follows:

$$\mathbf{x}(t+1)=\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t)+\mathbf{e}(t)$$
$$\mathbf{y}(t)=\mathbf{h}(\mathbf{x}(t),\mathbf{v}(t),t)+\mathbf{m}(t) \tag{9}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are nonlinear state (or plant) and observation functions. To develop an algorithm for this case, Taylor expansion can be used to approximate the nonlinear system Eq. (9). First let us introduce the following Jacobian matrices for the nonlinear state and measurement equations respectively, i.e.,

$$\mathbf{F}(t)=\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u}(t),t)\big|_{\mathbf{x}=\hat{\mathbf{x}}(t|t)}$$
$$\mathbf{H}(t)=\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},\mathbf{v}(t),t)\big|_{\mathbf{x}=\hat{\mathbf{x}}(t|t-1)}. \tag{10}$$

Based on the Jacobian matrices above, following equations can be derived from first order Taylor expansion, i.e.,

$$\begin{aligned}\mathbf{x}(t+1)&=\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),t)+\mathbf{e}(t)\\&\approx\mathbf{F}(t)\mathbf{x}(t)+\mathbf{f}(\hat{\mathbf{x}}(t|t),\mathbf{u}(t),t)-\mathbf{F}(t)\hat{\mathbf{x}}(t|t)+\mathbf{e}(t)\\&=\mathbf{F}(t)\mathbf{x}(t)+\boldsymbol{\Psi}(\hat{\mathbf{x}}(t|t),\mathbf{u}(t),t)+\mathbf{e}(t)\end{aligned} \tag{11}$$

$$\begin{aligned}\mathbf{y}(t)&=\mathbf{h}(\mathbf{x}(t),\mathbf{v}(t),t)+\mathbf{m}(t)\\&\approx\mathbf{H}(t)\mathbf{x}(t)+\mathbf{h}(\hat{\mathbf{x}}(t|t-1),\mathbf{v}(t),t)-\mathbf{H}(t)\hat{\mathbf{x}}(t|t-1)+\mathbf{m}(t)\\&=\mathbf{H}(t)\mathbf{x}(t)+\boldsymbol{\Upsilon}(\hat{\mathbf{x}}(t|t-1),t)+\mathbf{m}(t)\end{aligned} \tag{12}$$

where both $\boldsymbol{\Psi}(t)$ and $\boldsymbol{\Upsilon}(t)$ are already determined from previous prediction or estimation step; $\mathbf{e}(t)$ and $\mathbf{m}(t)$ are assumed to be white noise. Therefore, we can apply the previous linear Kalman filtering procedures (in particular for estimation of the covariance matrices), leading to the extended Kalman filter (EKF) algorithm in Table 2. In practice, EKF often performs well in solving state estimation problems with nonlinear state-space models. In particular, it fits for systems with relatively smooth nonlinearities or high measurement frequency.

2.2. Model identification for driver following behavior

The dynamics of vehicle movement in the driver-following stage can be described by the following equations:

$$\frac{d^2s_n(t,\boldsymbol{\theta})}{dt^2}=a_n(t,\boldsymbol{\theta})$$
$$a_n(t+\tau,\boldsymbol{\theta})=f(\mathbf{x}_{n-1}(t),\mathbf{x}_n(t),\boldsymbol{\theta}) \tag{13}$$

where $\mathbf{x}_n(t)=[s_n(t)\ v_n(t)\ a_n(t)]^T$ is the physical state (position, speed, and acceleration) of vehicle n at time t; τ is the reaction delay and $\boldsymbol{\theta}$ is the model parameter vector. Since distance headway, speed difference and following speed have been identified as the most common influence factors in car-following models, we can simply reformulate a corresponding discrete time system as follows:

$$s_n(t+1,\boldsymbol{\theta})=s_n(t,\boldsymbol{\theta})+v_n(t,\boldsymbol{\theta})\Delta t+\tfrac{1}{2}a_n(t,\boldsymbol{\theta})\Delta t^2$$
$$v_n(t+1,\boldsymbol{\theta})=v_n(t,\boldsymbol{\theta})+a_n(t,\boldsymbol{\theta})\Delta t$$
$$a_n(t+1,\boldsymbol{\theta})=f(t,v_{n-1}-v_n,s_{n-1}-s_n,v_n,\boldsymbol{\theta}) \tag{14}$$

Table 2
The discrete-time extended Kalman filter algorithm.

State update (prediction):
$\hat{\mathbf{x}}(t|t-1)=\mathbf{f}(\hat{\mathbf{x}}(t-1|t-1),\mathbf{u}(t-1),t-1)$
$\mathbf{F}(t-1)=\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x},\mathbf{u}(t-1),t-1)\big|_{\mathbf{x}=\hat{\mathbf{x}}(t-1|t-1)}$
$\mathbf{P}(t|t-1)=\mathbf{F}(t-1)\mathbf{P}(t-1|t-1)\mathbf{F}(t-1)^T+\mathbf{R}_e(t)$

Measurement update (correction):
$\mathbf{H}(t)=\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x},\mathbf{v}(t),t)\big|_{\mathbf{x}=\hat{\mathbf{x}}(t|t-1)}$
$\Gamma(t)=\mathbf{P}(t|t-1)\mathbf{H}(t)^T[\mathbf{H}(t)\mathbf{P}(t|t-1)\mathbf{H}(t)^T+\mathbf{R}_m(t)]^{-1}$
$\hat{\mathbf{x}}(t|t)=\hat{\mathbf{x}}(t|t-1)+\Gamma(t)(\mathbf{y}(t)-\mathbf{h}(\hat{\mathbf{x}}(t|t-1),\mathbf{v}(t),t))$
$\mathbf{P}(t|t)=(I-\Gamma(t)\mathbf{H}(t))\mathbf{P}(t|t-1)$

where Δt is the sampling time. Above, we have omitted the driver reaction delay, resulting in a simpler state space representation of the car-following dynamics. A fixed driver reaction time is however often assumed during a car-following process. Therefore the delayed acceleration can be manipulated in the offline data when constant reaction time is used.

Suppose that the true car-following dynamics can be described in terms of the parameter vector $\boldsymbol{\theta}$. Based on the observations of the random multivariate vehicle states correlated with the parameter vector, a Bayesian framework can be formulated to identify the parameter values (Ljung, 1983), that is

$$\boldsymbol{\theta} = \mathbb{E}(\boldsymbol{\theta}(t)|\mathbf{y}(t),\mathbf{u}(t),\mathbf{v}(t)) \tag{15}$$

where $\mathbf{y}(t)$ is the measurement while $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are the plant and measurement inputs. Therefore, a recursive adaptive filtering approach can be applied (Haykin, 1986). The parameter vector can be treated as time invariant and the measurement of the state $\mathbf{x}_n$ can be utilized to estimate the parameter vector. A new state-space system model can be formulated accordingly

$$\begin{aligned}
&\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t)\\
&\hat{s}_n(t+1) = s_n(t,\boldsymbol{\theta}) + v_n(t,\boldsymbol{\theta})\Delta t + \tfrac{1}{2}a_n(t,\boldsymbol{\theta})\Delta t^2 + m_1(t)\\
&\hat{v}_n(t+1) = v_n(t,\boldsymbol{\theta}) + a_n(t,\boldsymbol{\theta})\Delta t + m_2(t)\\
&\hat{a}_n(t+1) = f(t, v_{n-1}-v_n, s_{n-1}-s_n, v_n, \boldsymbol{\theta}) + m_3(t)
\end{aligned} \tag{16}$$

or in a compact form as

$$\begin{aligned}
&\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t)\\
&\mathbf{y}_n(t) = \mathbf{h}(\boldsymbol{\theta}(t),\mathbf{x}_n(t),\mathbf{x}_{n-1}(t)) + \mathbf{m}(t)
\end{aligned} \tag{17}$$

where $\mathbf{y}_n(t) = [\hat{s}_n(t)\ \hat{v}_n(t)\ \hat{a}_n(t)]^T$ is the full observation of the physical state at time t (if the usage of all state variables is necessary in the car-following model calibration problem); $\mathbf{m}(t) = [m_1(t)\ m_2(t)\ m_3(t)]^T$ describes the deviation of the estimated states from the measurement.

In driver-following model identification, different objective functions based on individual physical variables, e.g., acceleration or speed or position, have been used for parameter estimation. In this study, we are interested in treating the problem with a general LS optimization objective represented as follows:

$$\min_{\boldsymbol{\theta}} \sum_t (\mathbf{y}_n(t) - \mathbf{h}(\boldsymbol{\theta},\mathbf{x}_n(t),\mathbf{x}_{n-1}(t)))^T \mathbf{W} (\mathbf{y}_n(t) - \mathbf{h}(\boldsymbol{\theta},\mathbf{x}_n(t),\mathbf{x}_{n-1}(t))). \tag{18}$$

Hence, instead of only considering the deviation between real acceleration and acceleration outputs from a car-following model, a more general calibration policy is proposed to minimize the expected deviation square of vehicle states using a positive definite weighting matrix $\mathbf{W} = \mathbf{B}^T \cdot \mathbf{B}$. To estimate the parameters in Eq. (17), it is natural to apply the EKF algorithm. The objective of the Kalman filter is however to minimize Eq. (8). Hence, the next section shows how to adapt the EKF algorithm to fulfill the general objective in Eq. (18).

2.3. Kalman filter and recursive least squares method

To adapt the Kalman filtering algorithm with an objective of Eq. (18) in model estimation, the detailed Kalman filtering procedure should be derived when Eq. (17) is linear. Following the two-stage procedures in Table 1, the algorithm can be written in a compact form as follows:

$$\begin{aligned}
&\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \Gamma(t)(\mathbf{y}(t) - \mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1))\\
&\Gamma(t) = \mathbf{P}(t-1)\mathbf{H}(t)^T[\mathbf{H}(t)\mathbf{P}(t-1)\mathbf{H}(t)^T + \mathbf{R}_m(t)]^{-1}\\
&\mathbf{P}(t) = \mathbf{P}(t-1) - \Gamma(t)\mathbf{H}(t)\mathbf{P}(t-1)
\end{aligned} \tag{19}$$

where $\hat{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}(t|t)$ is the state estimate corrected by measurement and $\mathbf{P}(t) = \mathbf{P}(t|t)$ is the posterior covariance matrix. In the appendix, a detailed proof shows that the iterative procedure above is equivalent to the recursive least squares (RLS) algorithm derived for the linear model

$$\mathbf{y}(t) = \mathbf{H}(t)\boldsymbol{\theta} + \mathbf{m}(t) \tag{20}$$

with optimization objective

$$\min_{\boldsymbol{\theta}} \sum_t (\mathbf{y}(t) - \mathbf{H}(t)\boldsymbol{\theta})^T \mathbf{W}_t (\mathbf{y}(t) - \mathbf{H}(t)\boldsymbol{\theta}) \tag{21}$$

where $\mathbf{W}_t = [\mathbf{R}_m(t)]^{-1}$. Therefore, realization of the general LS objective above can be implemented through manipulation of the measurement noise covariance in the Kalman filter. In fact, the EKF procedure with appropriate $\mathbf{R}_m$ in Table 2 applied for the nonlinear system of Eq. (17) should at least approximately approach the optimal solution of the objective of Eq. (18), though the method may suffer the same defect of local convergence as other derivative-based algorithms.

2.4. Static approach

To apply the Kalman filtering algorithms above, the formulation of $\mathbf{H}(t)$ for linear systems or Jacobian matrices for nonlinear systems is one of the most important procedures. In the car-following Eq. (16), the derivation of the Jacobian for the state-space system is not direct since the measurement equation includes a feed-back loop. One traditional and easy way in model estimation is to use the training data as inputs to the right hand side (RHS) of the acceleration Eq. (16). So for the linear model, the derivative of the acceleration at each time instant can be computed by

$$\mathbf{H}(t) = \frac{\partial a_n(t+1,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial f(t,\tilde{\mathbf{x}}_n,\tilde{\mathbf{x}}_{n-1},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{22}$$

where the observed state $\tilde{\mathbf{x}}(t) = [\hat{s}(t),\ \hat{v}(t),\ \hat{a}(t)]^T$ is used. From the system equation, the following iteration can be then obtained for the derivatives of speed and trajectory processes (the second order terms are neglected):

$$\begin{aligned}
&\frac{\partial v(t+1,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial v(t,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial a(t,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Delta t\\
&\frac{\partial s(t+1,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \approx \frac{\partial s(t,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial v(t,\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\Delta t
\end{aligned} \tag{23}$$

To test this approach numerically, let us consider the estimation of Helly's model of Eq. (1). By replacement of D_n of the first equation with the second equation, a simpler form of the Helly's model is obtained as follows:

$$\begin{aligned}
a_n(t+1) &= f(\mathbf{x}_n(t),\mathbf{x}_{n-1}(t),\boldsymbol{\theta})\\
&= \theta_1(s_{n-1}(t) - s_n(t)) + \theta_2(v_{n-1}(t) - v_n(t)) + \theta_3 a_n(t) + \theta_4 v_n(t) + \theta_5
\end{aligned} \tag{24}$$

Therefore, five model parameters $\theta_i\ i = 1 \ldots 5$ need to be identified. To estimate those parameters, a state space model can be formulated as

$$\begin{aligned}
&\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t)\\
&\mathbf{y}(t) = \mathbf{H}(t) \cdot \boldsymbol{\theta}(t) + \mathbf{m}(t)
\end{aligned} \tag{25}$$

where $\boldsymbol{\theta}$ is the parameter vector and $\mathbf{y}(t)$ is the measurement vector as before. $\mathbf{H}(t)$ is a 3×5 matrix, which can be computed by the update Eq. (23) and the following equation:

$$\partial a(t+1,\boldsymbol{\theta})/\partial \boldsymbol{\theta} = \begin{bmatrix} \hat{s}_{n-1}(t) - \hat{s}_n(t) \\ \hat{v}_{n-1}(t) - \hat{v}_n(t) \\ \hat{a}_n(t) \\ \hat{v}_n(t) \\ 1 \end{bmatrix} \tag{26}$$

This is a linear but nonstationary state-space model, and the parameters (the state) can be estimated by the linear Kalman filtering algorithm in Table 1. Based on the estimated parameters

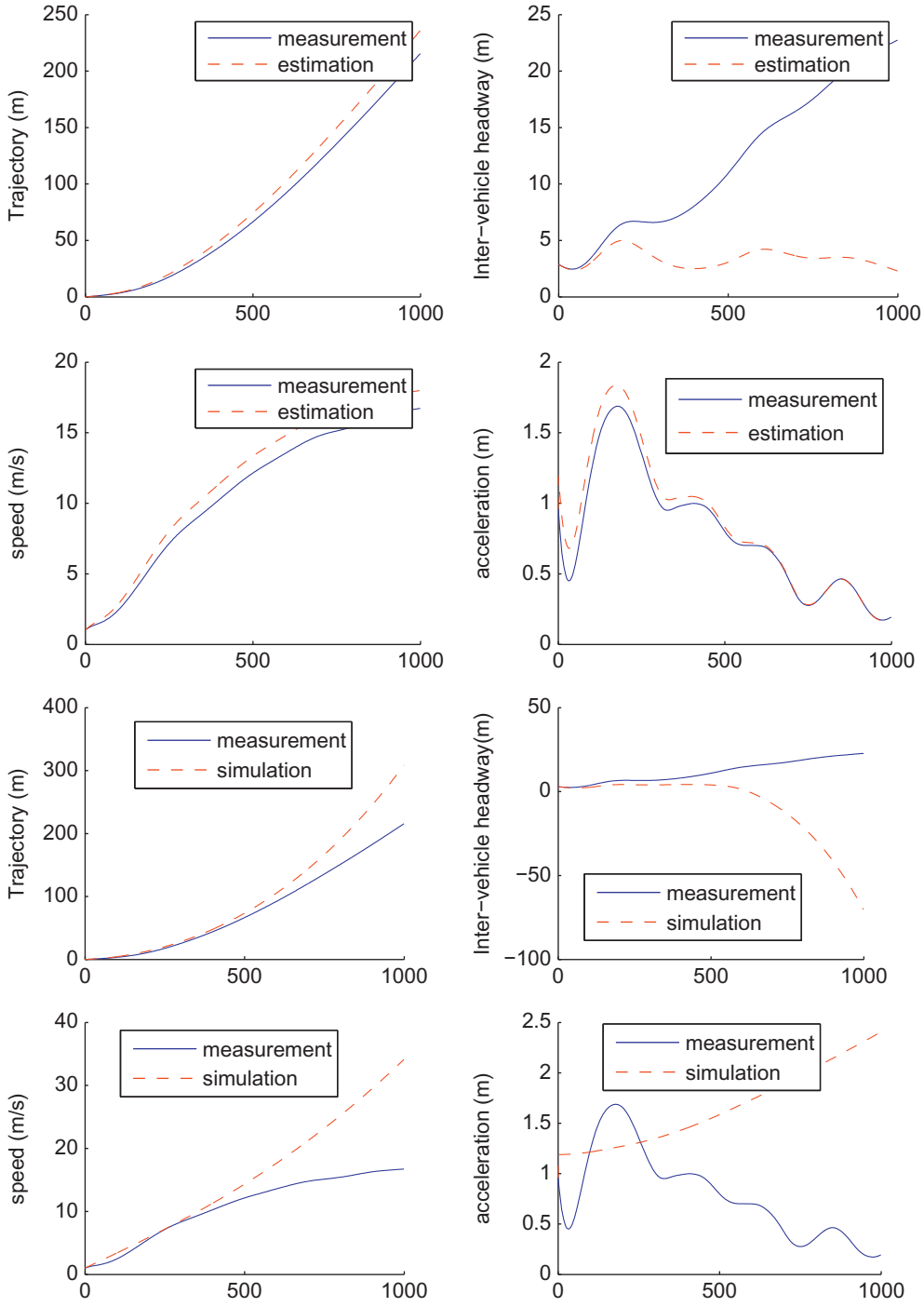

Fig. 1. Estimation of Helly's model based on a static approach: state estimation based on calibrated parameters and measurement inputs (upper) and state simulation based on calibrated parameters and initial condition (lower).

from the approach above, two state simulations are conducted. First, the observations at each time step are used as model inputs to run a one-step prediction using the calibrated parameters. Second, the dynamic states of the following vehicle are simulated in a closed loop given only the initial boundary condition. Fig. 1 shows the numerical results of the experiment. The one-step state prediction gives good results, which indicates that the calibrated model gives good prediction of the next state using measured current state as model inputs. However, the model estimated statically can not replicate the process in a closed loop simulation where only the initial state is given. This indicates that a static model estimation approach such as in Eq. (5) may be not good enough for a model to describe the driver-following dynamics in real data. A modified approach has to be derived.

2.5. Dynamic approach

The main reason that makes the performance of the static model estimation approach poor is probably the use of measurement data $\tilde{\mathbf{x}}_n(t)$ as model inputs in Eq. (22). The previous prediction errors in the process are thus not included in the estimation scheme. To derive an approach that can capture the dynamics, the posterior estimator $\hat{\mathbf{x}}(t|t)$ may be used to represent the current state instead. Therefore, a variation of the state space

Table 3
General car-following model estimation scheme based on the iterative extended Kalman filter algorithm (IEKF).

Initialize θ_0 with random values
while $\sum_i \|\theta_k(i)-\theta_{k-1}(i)\| > \epsilon$
  for each dataset S_i
    initialize parameters with values obtained
    after filtering dataset S_{i-1}
      $\theta_k(i,t=0)=\theta_k(i-1,t=T_{i-1})$
    applying EKF filter to the training dataset S_i
      $\theta_k(i)=\text{EKFprocedure}(\theta_k(i-1),S_i)$
  end for
end while

system of Eq. (16) is formulated as follows:

$$\begin{aligned}
&s_n(t+1)=s_n(t)+v_n(t)\Delta t+\tfrac{1}{2}a_n(t)\Delta t^2+e_1(t)\\
&v_n(t+1)=v_n(t)+a_n(t)\Delta t+e_2(t)\\
&a_n(t+1)=f(v_{n-1}(t)-v_n(t),D_n(t),v_n(t),\boldsymbol{\theta}(t))+e_3(t)\\
&\boldsymbol{\theta}(t+1)=\boldsymbol{\theta}(t)\\
&\hat{s}_n(t)=s_n(t)+m_1(t)\\
&\hat{v}_n(t)=v_n(t)+m_2(t)\\
&\hat{a}_n(t)=a_n(t)+m_3(t)
\end{aligned} \tag{27}$$

or in a compact form as follows:

$$\begin{aligned}
&\mathbf{X}_n(t+1)=\mathbf{F}(\mathbf{X}_n(t),\mathbf{x}_{n-1}(t))+\mathbf{e}(t)\\
&\mathbf{y}_n(t)=\mathbf{H}\cdot\mathbf{X}_n(t)+\mathbf{m}(t)
\end{aligned} \tag{28}$$

where $\mathbf{X}_n(t)=[s_n(t)\ v_n(t)\ a_n(t)\ \boldsymbol{\theta}(t)]^T$ is an extended state vector of the following vehicle and $\mathbf{x}_{n-1}(t)$ is the physical state vector of the leading vehicle, which can be treated as a control input vector; $\mathbf{H}=[\mathbf{I}_3\ \mathbf{0}]$ is a $3\times M$ matrix; $\mathbf{y}_n(t)=[\hat{s}_n(t)\ \hat{v}_n(t)\ \hat{a}_n(t)]^T$ is the measurement vector; $\mathbf{m}(t)=[m_1(t)\ m_2(t)\ m_3(t)]^T$ is a vector of deviation at time t. In the formulation above, the state of the following vehicle becomes explicitly modeled in the state transition equation. To solve the estimation problem, the extended Kalman filter in Table 2 can be directly applied. However, given multiple time series in the driver following dataset, it is necessary to apply EKF recursively until the estimated parameters converge (i.e., the model can thus capture all driver-following patterns). Table 3 summarizes an algorithm for estimating multiple driver-following time series patterns by iterative usage of the EKF algorithm.

Various model identification objectives can be implemented based on the choice of weighting matrix of Eq. (18). Through adjustment of the covariance matrix of the measurement noise, different parameter estimation results can be obtained. In principle, the inverse of the covariance matrix computed from time series data should be used as the weighting matrix to estimate optimal model parameters (Strang, 1986). In practical car-following experiments, only certain variables in the vehicle state vectors are directly measured. For example, trajectory is directly observed in both GPS data and our laser data collection. Therefore, trajectory data is the most reliable for us. A simple way to reflect this fact is to set the noise covariance matrix as e.g.,

$$\mathbf{R}_m=\begin{pmatrix}10^{-6}&0&0\\0&1&0\\0&0&1\end{pmatrix} \tag{29}$$

A more direct approach in the implementation is to keep the measurement equation directly related to measurement (trajectory in our case). If more than one data source, e.g., laser data (trajectory) and radar data (speed), are available, an appropriate noise covariance matrix reflecting the relative reliability of the data sources shall be used. In general, the method developed in this paper provides a potential tool for modelers to evaluate different identification objectives in the driver-following context, especially more than one data sources are available.

3. Numerical experiments

The dynamic approach formulated in the last section has been implemented using an iterative EKF algorithm in the MATLAB environment. Two types of model structures, the linear Helly model and GM models, are estimated using datasets collected. Similar as before, the derivation of the Jacobian matrix for the state transition Eq. (28) is the essential procedure. For the linear Helly model, the following Jacobian matrix can be obtained:

$$\mathbf{F}(t)=\begin{pmatrix}
1&\Delta t&\frac{1}{2}\Delta t^2&0&0&0&0&0\\
0&1&\Delta t&0&0&0&0&0\\
-\theta_1&\theta_4-\theta_2&\theta_3&s_{n-1}-s_n&v_{n-1}-v_n&a_n&v_n&1\\
0&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&1
\end{pmatrix}_{\mathbf{X}=\hat{\mathbf{X}}(t|t)} \tag{30}$$

Figs. 2 and 3 illustrate an example of estimating Helly's model using the dataset as that was used in the previous static approach. Fig. 2 shows the parameter evolution along time in the last estimation epoch. In Fig. 3, the linear Helly model estimated by the dynamic IEKF is applied to simulate the car-following process given only the initial condition. Such state replication reflects the reliability of model identification results. Meanwhile, the result is related to the model fidelity itself. Fig. 3 indicates that the dynamic approach can, though inaccurately, capture the car-following dynamics that the model learned. The method has therefore advantages over the static algorithm in Section 2.4. The deviation of the simulated states from reality is probably due to the limitation of the Helly model in representing the nonlinear process. Meanwhile, even though the Helly model is linear, the estimation problem is indeed nonlinear (Grewal & Andrews, 1993). Hence, it is not guaranteed that a global optimal solution is found by the filtering process.

The study next considers evaluation of the approach using a nonlinear model, the general GM model (GGM) represented by Eq. (3). The Jacobian of state transition equation is quite similar to Eq. (30) but the acceleration gradient vector becomes much more complex, i.e.,

$$\partial a(t,\boldsymbol{\theta})/\partial\boldsymbol{\theta}=\begin{pmatrix}
\frac{\theta_1(t)\cdot\theta_3(t)\cdot v_n(t)^{\theta_2(t)}\cdot\Delta v(t)}{\Delta s(t)^{(\theta_3(t)+1)}}\\
\frac{\theta_1(t)\cdot\theta_2(t)\cdot v_n(t)^{(\theta_2(t)-1)}}{\Delta s(t)^{(\theta_2(t)+1)}}-\frac{\theta_1(t)\cdot(\theta_2(t)+1)\cdot v_n(t)^{\theta_2(t)}}{\Delta s(t)^{\theta_3(t)}}\\
\theta_5(t)\\
\frac{v_n(t)^{\theta_2(t)}\cdot\Delta v(t)}{\Delta s(t)^{\theta_3(t)}}\\
\frac{\theta_1(t)\cdot v_n(t)^{\theta_2(t)}\cdot\log v_n(t)\cdot\Delta v(t)}{\Delta s(t)^{\theta_3(t)}}\\
-\frac{\theta_1(t)\cdot v_n(t)^{\theta_2(t)}\cdot\log\Delta s(t)\Delta v(t)}{\Delta s(t)^{\theta_3(t)}}\\
a_{n-1}(t)\\
a_n(t)
\end{pmatrix}_{\mathbf{X}=\hat{\mathbf{X}}(t|t-1)} \tag{31}$$

where $\Delta\hat{s}(t)=s_{n-1}(t)-s_n(t)$ and $\Delta v(t)=v_{n-1}(t)-v_n(t)$. Of course, the derivative vector can be approximated by a numerical scheme but it involves the risk of undermining the general estimation performance. By running the EKF algorithm in Table 3 repeatedly, the GGM model has been tested on measurement datasets. Fig. 4 shows an example of the simulation results of state replication,

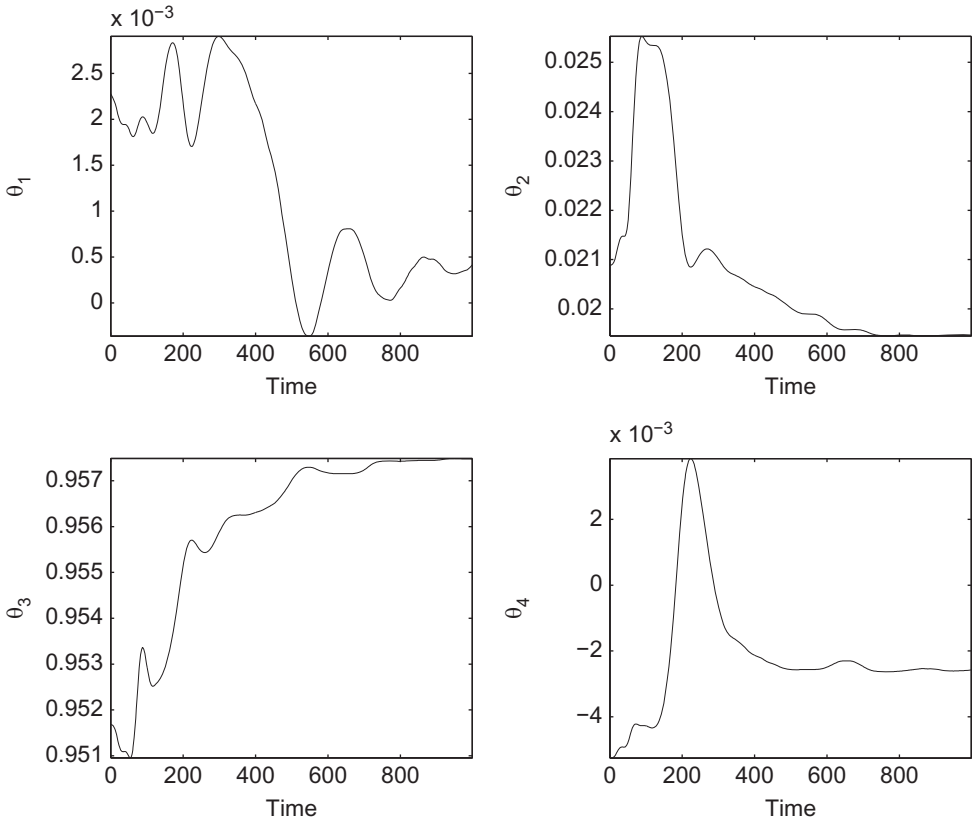

Fig. 2. Calibration of the Helly's model represented by Eq. (24): the evolution of the parameters in the last EKF run.

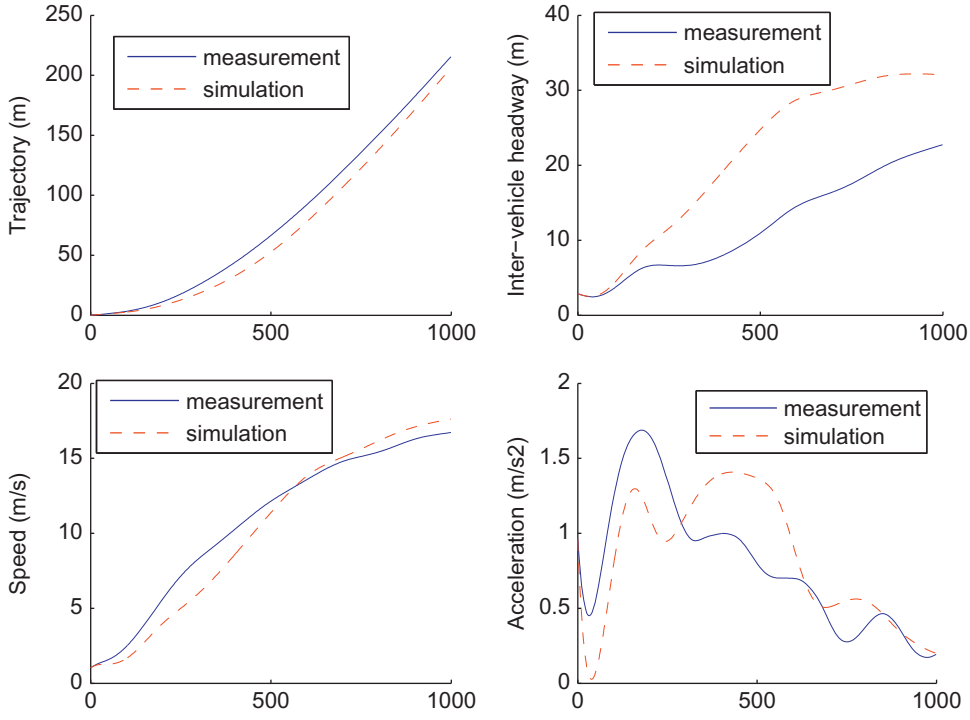

Fig. 3. Numerical experiment of the Helly's model: a closed-loop simulation with parameters estimated from the same data set (lower).

where the same dataset used in the previous Helly's case is applied for model identification. The small difference between the simulated and measured car-following processes indicates that the GGM model can better capture the car-following dynamics. This proves the hypothesis that the model structure affects the model's potential ability: the nonlinear GGM model can describe the behavioral dataset better than the linear Helly model.

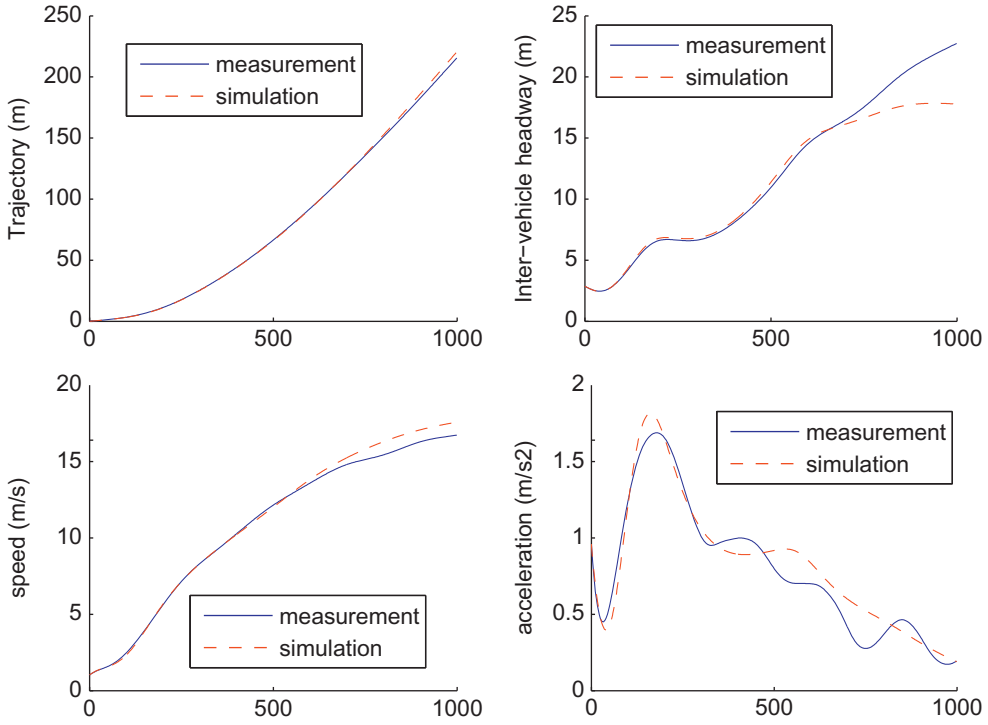

Fig. 4. State replication in the GGM model: closed-loop simulation using parameters estimated dynamically.

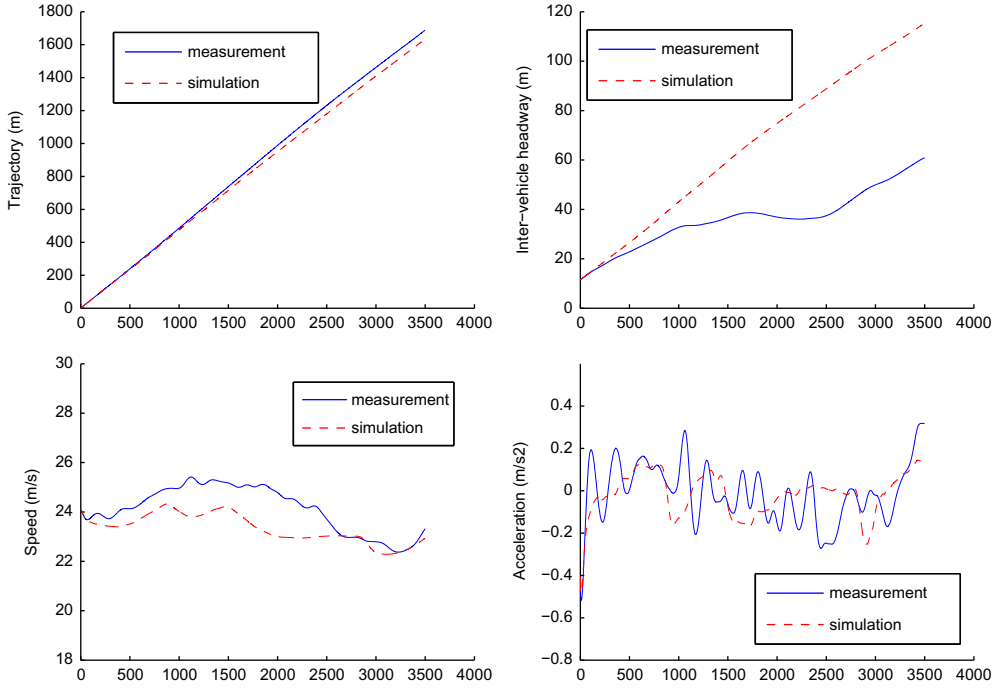

Fig. 5. Cross-validation in the GGM model: closed-loop simulation using model estimated from a single car-following dataset.

In Fig. 5, the model estimated before is applied in a closed-loop simulation to predict another car-following process (3500 data points) given the initial condition of the following vehicle. This is in fact a cross-validation procedure. While the model estimated from the independent dataset cannot capture the detailed trends in the acceleration curve the speed profiles match in general. The latter part of inter-vehicle headway curve is not replicated in a fully satisfying manner. This may be because the GGM model is based on speed matching and an extended GM model with distance headway matching could improve the space headway prediction. Meanwhile, the difference between drivers may also explain such deviation in cross-validation.

In Fig. 6, a cross-validation test is conducted based on the modified GGM model, in which the autoregressive term on acceleration in Eq. (3) is removed. The model is estimated using several datasets of different drivers so that "average" behavior of several drivers is approximated by the model. It is observed that the acceleration curve of the dataset (2500 data points) is well

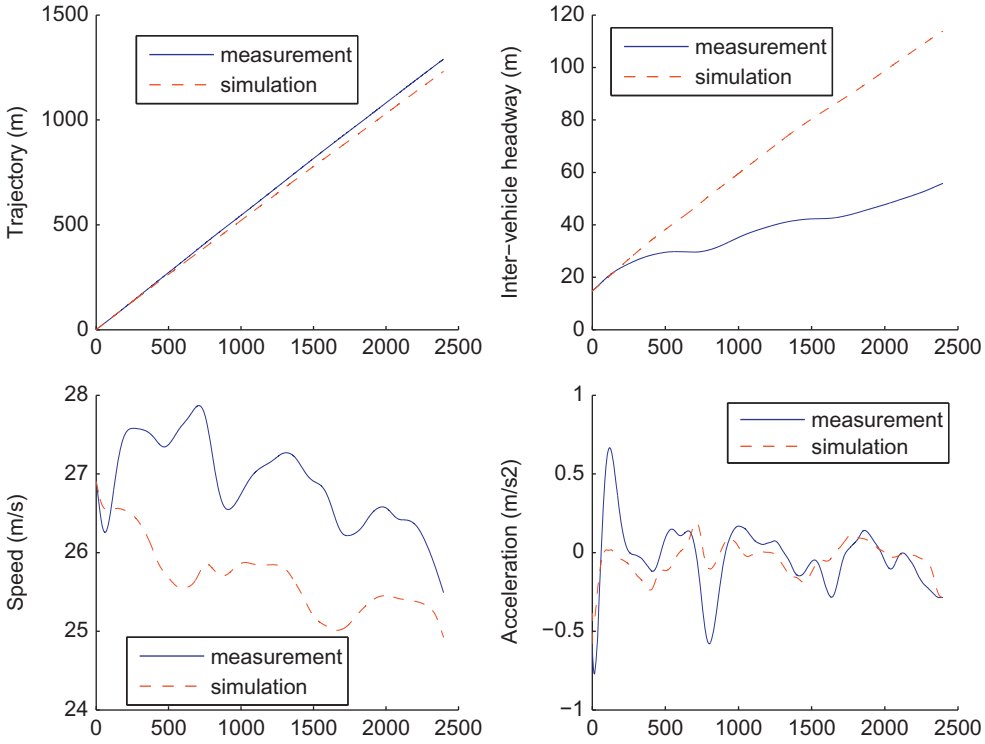

Fig. 6. Cross-validation using the GGM model: closed-loop simulation using model estimated from multiple datasets.

predicted except in the beginning. The initial speed difference hence leads to the deviation of the simulated inter-vehicle headway from reality.

One observation in our experiments is that the parameters estimated by the iterative EKF algorithm are not guaranteed to converge. This meets an expected defect of Kalman filter or RLS: performance corruption by rounding errors (Grewal & Andrews, 1993). However, several other factors in the application may also lead to divergence: improper model structure and design parameters in IEKF. In particular, the first order approximation of the nonlinear system in EKF may undermine the filtering performance significantly. To improve the numerical stability, noise terms are in practice added in the state Eq. (28), especially to the acceleration model (i.e., allowing model inaccuracy in acceleration). Meanwhile, parameters adjustment may improve the performance of the filtering algorithm. In addition to convergence, it is also worth mentioning that the EKF (or RLS) based algorithm is not guaranteed to find the global optimal solution. While multiple initial random guesses may increase the chance to find the global optimum, it also increases the computational time significantly.

4. Summary and conclusion

This paper presents a model identification scheme based on iterative usage of the extended Kalman filtering algorithm. The method can not only be applied for estimation of model parameters but also be extended to train complex controllers for driver-following dynamics in ITS applications. The filtering algorithm is developed to fulfill a general car-following model identification objective based on the full vehicle state variables. That is

$$\min \mathbb{E}_t[(\mathbf{x}_n^{sim}(t)-\mathbf{x}_n^{obs}(t))^T \mathbf{W}(\mathbf{x}_n^{sim}(t)-\mathbf{x}_n^{obs}(t))] \tag{32}$$

where $\mathbf{x}_n(t)=[s_n(t)\ v_n(t)\ a_n(t)]^T$ represents the whole state of vehicle n at time t and $\mathbf{W}$ is a nonnegative definite matrix varying according to applications. Compared to the conventional model identification approach, the IEKF based algorithm can better capture the car-following dynamics. This has been illustrated in the numerical experiments of the study.

The new identification objective function is more general than the conventional setup, which makes it able of incorporating multiple measurement data sources and testing different model identification strategies. Further evaluation of the identification objective functions is still needed in the future study. According to the numerical experiments, the computer implementation of the algorithm is efficient for model identification. The method could be applied to estimate driver-following models or train ACC controller automatically using a large number of car-following patterns.

There is still a large potential to extend the work in algorithm development and applications. For example, the current EKF algorithm approximates the state transition model to the first order with its Jacobian function. This may lead to problems of inaccuracy and instability in the estimation of highly nonlinear models or controllers. An obvious extension is to apply the higher-order filters such as unscented Kalman filter (UKF) algorithm, in which sample points are used to obtain first and second order statistical information by going through the nonlinear systems repeatedly (Julier & Uhlmann, 2004). Such extension can be particularly evaluated when more complex models or controllers such as ANN are estimated for driver-following.

Appendix: Weighted recursive least squares and the Kalman filter

This section shows how the least squares solution of a quite general quadratic criterion function can be updated in time. The derivations parallel the standard derivations in the literature but the solution is usually only found in the scalar measurement

case. Here this is extended to the vector case and also include the possibility to weight the measurements by positive definite weighting matrices. This also allows us to show some further connections between the obtained recursive least squares (RLS) solution and the Kalman filter.

Consider the weighted least squares criterion at discrete time t

$$V_t(\boldsymbol{\theta}) = \sum_{k=1}^{t} \lambda^{t-k}(\mathbf{y}(k)-\mathbf{H}(k)\boldsymbol{\theta})^T\mathbf{W}_k(\mathbf{y}(k)-\mathbf{H}(k)\boldsymbol{\theta})+\lambda^t(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^T\mathbf{Q}_0(\boldsymbol{\theta}-\boldsymbol{\theta}_0) \tag{33}$$

where $\mathbf{W}_k$ are given positive definite weighting matrices, and λ is the so-called forgetting factor ($0<\lambda\leq 1$). Setting λ slightly less than one makes old data less influential on the criterion. In the last term of the criterion, $\boldsymbol{\theta}_0$ is a given vector and $\mathbf{Q}_0$ a given positive definite matrix. This term is included to take potential "initial information" about $\boldsymbol{\theta}$ into account. Note that the influence of this term will decay as more data are collected (t increases).

The minimizer of the quadratic criterion (33) with respect to $\boldsymbol{\theta}$ is obtained by equating the gradient to zero. This leads to the normal equations

$$\boldsymbol{\Phi}_t\boldsymbol{\theta} = \boldsymbol{\Psi}_t \tag{34}$$

where

$$\boldsymbol{\Phi}_t = \sum_{k=1}^{t} \lambda^{t-k}\mathbf{H}(k)^T\mathbf{W}_k\mathbf{H}(k)+\lambda^t\mathbf{Q}_0 \tag{35}$$

$$\boldsymbol{\Psi}_t = \sum_{k=1}^{t} \lambda^{t-k}\mathbf{H}(k)^T\mathbf{W}_k\mathbf{y}(k)+\lambda^t\mathbf{Q}_0\boldsymbol{\theta}_0 \tag{36}$$

Clearly, the solution to the normal equations in (34) $\hat{\boldsymbol{\theta}}(t)=\boldsymbol{\Phi}_t^{-1}\boldsymbol{\Psi}_t$ is the optimal least squares estimate at time t. Next we note that $\boldsymbol{\Phi}_t$ and $\boldsymbol{\Psi}_t$ can be easily updated recursively in time as

$$\boldsymbol{\Phi}_t = \lambda\boldsymbol{\Phi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{H}(t) \tag{37}$$

$$\boldsymbol{\Psi}_t = \lambda\boldsymbol{\Psi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{y}(t) \tag{38}$$

The least squares solution at time t can be rewritten as

$$\hat{\boldsymbol{\theta}}(t) = \boldsymbol{\Phi}_t^{-1}\boldsymbol{\Psi}_t = \hat{\boldsymbol{\theta}}(t-1)+\boldsymbol{\Phi}_t^{-1}[\boldsymbol{\Psi}_t-\boldsymbol{\Phi}_t\hat{\boldsymbol{\theta}}(t-1)] \tag{39}$$

By using (37), we have

$$\begin{aligned}\boldsymbol{\Phi}_t\hat{\boldsymbol{\theta}}(t-1) &= \lambda\boldsymbol{\Phi}_{t-1}\hat{\boldsymbol{\theta}}(t-1)+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)\\ &= \lambda\boldsymbol{\Psi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)\end{aligned} \tag{40}$$

where we used the fact that $\boldsymbol{\Phi}_{t-1}\hat{\boldsymbol{\theta}}(t-1)=\boldsymbol{\Psi}_{t-1}$ in the last equality. Combining (40) and (38) we get

$$\begin{aligned}\boldsymbol{\Psi}_t-\boldsymbol{\Phi}_t\hat{\boldsymbol{\theta}}(t-1) &= \lambda\boldsymbol{\Psi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{y}(t)-\left(\lambda\boldsymbol{\Psi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)\right)\\ &= \mathbf{H}(t)^T\mathbf{W}_t\left(\mathbf{y}(t)-\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)\right)\end{aligned} \tag{41}$$

The parameter update Eq. (39) can now be written as

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1)+\boldsymbol{\Phi}_t^{-1}\mathbf{H}(t)^T\mathbf{W}_t\left(\mathbf{y}(t)-\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1)\right) \tag{42}$$

It remains to show how $\boldsymbol{\Phi}_t^{-1}\mathbf{H}(t)^T\mathbf{W}_t$ can be updated more efficiently. For this purpose, define $\mathbf{P}_t=\boldsymbol{\Phi}_t^{-1}$, and recall the matrix inversion lemma

$$(\mathbf{A}+\mathbf{BCD})^{-1} = \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{DA}^{-1}\mathbf{B}+\mathbf{C}^{-1})^{-1}\mathbf{DA}^{-1} \tag{43}$$

for any matrices $\{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}\}$ of compatible dimensions and provided the inverses exist. Applying (43) to (37) with $\mathbf{A}=\lambda\boldsymbol{\Phi}_{t-1}$, $\mathbf{B}=\mathbf{H}(t)^T$, $\mathbf{C}=\mathbf{W}_t$, $\mathbf{D}=\mathbf{H}(t)$ yields

$$\begin{aligned}\mathbf{P}_t = \boldsymbol{\Phi}_t^{-1} &= [\lambda\boldsymbol{\Phi}_{t-1}+\mathbf{H}(t)^T\mathbf{W}_t\mathbf{H}(t)]^{-1}\\ &= \lambda^{-1}\mathbf{P}_{t-1}-\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T[\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1}]^{-1}\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\end{aligned} \tag{44}$$

Table 4
The weighted recursive least squares method which minimizes the criterion in (33).

Initialize: $\hat{\boldsymbol{\theta}}(0)=\boldsymbol{\theta}_0$ and $\mathbf{P}_0=\mathbf{Q}_0^{-1}$
For $t=1,2,\ldots$
$\mathbf{K}(t)=\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T(\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1})^{-1}$
$\hat{\boldsymbol{\theta}}(t)=\hat{\boldsymbol{\theta}}(t-1)+\mathbf{K}(t)(\mathbf{y}(t)-\mathbf{H}(t)\hat{\boldsymbol{\theta}}(t-1))$
$\mathbf{P}_t=\lambda^{-1}\mathbf{P}_{t-1}-\mathbf{K}(t)\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}$

Finally,

$$\begin{aligned}\boldsymbol{\Phi}_t^{-1}\mathbf{H}(t)^T\mathbf{W}_t &= \lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T\mathbf{W}_t\\ &\quad-\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T[\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1}]^{-1}\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T\mathbf{W}_t\\ &= \lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T[\mathbf{I}-[\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1}]^{-1}\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T]\mathbf{W}_t\\ &= \lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T(\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1})^{-1}\\ &\quad\times[\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1}-\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T]\mathbf{W}_t\\ &= \lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T(\mathbf{H}(t)\lambda^{-1}\mathbf{P}_{t-1}\mathbf{H}(t)^T+\mathbf{W}_t^{-1})^{-1}\end{aligned} \tag{45}$$

By noting that for $t=0$, $\hat{\boldsymbol{\theta}(0)}=\boldsymbol{\theta}_0$ and $\mathbf{P}_0=\mathbf{Q}_0^{-1}$, we can summarize the weighted RLS algorithm as in Table 4.

The estimates provided by the RLS formulation in Table 4 can be shown to be equivalent to the estimates obtained by applying the Kalman filter to the following stochastic state-space model:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t)+\mathbf{e}(t) \tag{46}$$

$$\mathbf{y}(t) = \mathbf{H}(t)\boldsymbol{\theta}(t)+\mathbf{m}(t) \tag{47}$$

where $\mathbf{e}(t)$ and $\mathbf{m}(t)$ are independent zero-mean white stochastic processes with covariance matrices $\mathbf{R}_e(t)=1-\lambda/\lambda\mathbf{P}(t|t)$ and $\mathbf{R}_m(t)=\mathbf{W}_t^{-1}$, respectively. In particular, note that with this state equation we have $\mathbf{P}(t|t-1)=\lambda^{-1}\mathbf{P}(t-1|t-1)$. The initial conditions are $\hat{\boldsymbol{\theta}}(0|0)=\boldsymbol{\theta}_0$ and $\mathbf{P}(0|0)=\mathbf{Q}_0^{-1}$. This observation of the equivalent formulations of the Kalman filter and RLS, respectively, aids in the understanding of the problem and gives intuition how to tune the estimation methods. It is also worth mentioning that when no forgetting factor exists i.e., $\lambda=1$ the mathematical equivalence between RLS and Kalman filter in the parameter identification context leads to the fact that no noise is involved in the state equation i.e., $\mathbf{e}(t)=\mathbf{0}$.

References

Addison, P., & Low, D. (1998). A novel nonlinear car-following model. *Chaos*, *8*(4 (December)), 791–799.

Ahmed, K. (1999). *Modeling drivers' acceleration and lane changing behavior*. Ph.D. Thesis, Massachusetts Institute of Technology.

Bifulco, G., Pariota, L., Simonelli, F., & Pace, R. D. (2011). Real-time smoothing of car-following data through sensor-fusion techniques. In: *Procedia*: *Social and behavioral sciences* (Vol. 20, pp. 524–535). Elsevier Ltd.

Bouchard, M. (2001). New recursive-least-squares algorithms for nonlinear active control of sound and vibration using neural networks. *IEEE Transactions on Neural Networks*, *12*(1), 135–147.

Brackstone, M., & McDonald, M. (1999). Car following: A historical review. *Transportation Research Part F*, *2*, 181–196.

Chakroborty, P., & Kikuchi, S. (1999). Evaluation of general motors based car following models and a proposed fuzzy inference model. *Transportation Research Part C*, *7*, 209–235.

Chandler, R., Herman, R., & Montroll, E. (1958). Traffic dynamics: Studies in car following. *Operational Research*, *6*, 165–184.

Gazis, D., Herman, R., & Rothery, R. (1961). Nonlinear follow-the-leader models of traffic flow. *Operational Research*, *9*, 545–567.

Grewal, M., & Andrews, A. (1993). *Kalman filtering: Theory and practice*. New Jersey: Prentice Hall.

Haykin, S. (1986). *Adaptive filter theory*. Englewood Cliffs, New Jersey: Prentice-Hall.

Helly, W. (1959). Simulation of bottlenecks in single lane traffic flow. In *Proceedings of the symposium on theory of traffic flow. Research laboratories, General Motors* (pp. 207–238). New York: Elsevier.

Hoogendoorn, S. P., Ossen, S., & Schreuder, M. (2007). Adaptive car-following behavior identification by unscented particle filtering. In *Proceedings of 86th transportation research board annual meeting*.

Hwang, S., & He, R. (2006). A hybrid real-parameter genetic algorithm for function optimization. *Advanced Engineering Informatics*, *20*, 7–21.

Julier, S., & Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of IEEE*, *92*(3), 401–422.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *ASME Transaction: Journal of Basic Engineering*, *D*, 35–45.

Leutzbach, W., & Wiedemann, R. (1986). Development and applications of traffic simulation models at the karlsruhe institut fur verkehrwesen. *Traffic Engineering and Control*. 270–278.

Ljung, L. (1983). *Theory and practice of recursive identification*. Cambridge, MA: The MIT Press.

Ma, X., & Andréasson, I. (2005). Dynamic car following data collection and noise cancellation based on the Kalman smoothing. In *Proceedings of IEEE international conference on vehicular electronics and safety* (ICVES'05).

Ma, X., & Andréasson, I. (2006). Driver reaction time estimation from the real car following data and its application in GM-type model evaluation. Transportation Research Record No. 1965, 130–141.

Ma, X., & Andréasson, I. (2007). Behavior measurement, analysis and regime classification in car-following. *IEEE Transactions on Intelligent Transportation Systems*, *8*(1), 144–156.

Minoiu, E., Netto, M., Mammar, S., & Lusetti, B. (2009). Driver steering assistance for lane departure avoidance. *Control Engineering Practice*, *17*, 642–651.

Moon, S., Moon, I., & Yi, K. (2009). Design, tuning and evaluation of a full-range adaptive cruise control system with collision avoidance. *Control Engineering Practice*, *17*, 442–455.

Punzo, V., & Simonelli, F. (2005). Analysis and comparison of microscopic traffic flow models with real traffic microscopic data. Transportation Research Record No. 1934.

Ranjitkar, P., Nakatsuji, T., & Kawamura, A. (2005). Experimental analysis of car-following dynamics and traffic stability. Transportation Research Record No. 1934, pp. 22–32.

Simonelli, F., Bifulco, G. N., Martinis, V. D., & Punzo, V. (2009). Human-like adaptive cruise control systems through a learning machine approach. *Applications of soft computing* (pp. 240–249). Springer-Verlag.

Strang, G. (1986). *Introduction to applied mathematics*. Wellesley, MA: Wellesley-Cambridge Press.

Sultan, B., Brackstone, M., & McDonald, M. (2004). Drivers' use of deceleration and acceleration information in car-following process. Transportation Research Record No. 1883, pp. 31–39.

Wilmink, I., Kunder, G., & van Arem, B. (2007). Traffic flow effects of integrated full-range speed assistance. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 1204–1210).

Yang, Q. (1997). *A simulation laboratory for evaluation of dynamic traffic management systems*. Ph.D. Thesis, Massachusetts Institute of Technology.