A New Vehicle Motion Model for Improved Predictions and Situation Assessment

Joakim Sörstedt, Lennart Svensson, Fredrik Sandblom, and Lars Hammarstrand

***Abstract*—Reliable and accurate vehicle motion models are of vital importance for automotive active safety systems for a number of reasons. First of all, these models are necessary in tracking algorithms that provide the safety system with information. Second, the motion model is often used by the safety application to make long-term predictions about the future traffic situation. These predictions are then part of the basic data used by the system to determine if, when, and how to intervene. In this paper, we suggest a framework for designing accurate vehicle motion models. The resulting models differ from conventional models in that the expected control input from the driver is included. By also providing a methodology for a formal treatment of the uncertainties, a model structure well suited, e.g., in a tracking algorithm, is obtained. To utilize the framework in an application will require careful design and validation of submodels to calculate the expected driver control input. We illustrate the potential of the framework by examining the performance for a specific model example using real measurements. The properties are compared with those of a constant acceleration model. Evaluations indicate that the proposed model yields better predictions and that it has an ability to estimate the prediction uncertainties.**

***Index Terms*—Active safety systems, automotive, driver models, motion model, optimal control, predictions, tracking.**

I. INTRODUCTION

A CURRENT trend in today's automotive industry is to equip vehicles with increasingly more active safety systems. These have the objective of aiding the driver in different accident-prone situations, such as unintentional lane departures or dangers caused by distracted drivers. Based on sensor data, active safety systems try to assess the situation and detect dangerous scenarios. Some examples of active safety systems that are currently available on the market are given in [1]. To improve these systems in the future, detailed vehicle motion models will be most valuable. The reasons for this are at least threefold.

1) The motion model is an integral part of the tracking system that provides the safety system with information regarding the surrounding environment. More precise motion models would improve the tracking system's ability to handle complex traffic scenarios.
2) An active safety system needs to decide when and how to intervene. In theory, an accurate motion model enables the safety applications to make better long-term predictions and thereby allow decisions to be made earlier and with greater confidence [2].
3) Careful modeling of specific situations of interest, e.g., related to different driver maneuvers, can help the safety application to classify and understand the traffic situation.

The motion models predominantly used in tracking systems are derived from the fundamental laws of dynamics and assume that the input is approximately white Gaussian noise. Such models have extensively been employed in numerous applications and often with satisfactory results [3]–[5]. However, there are reasons to believe that traditional models can be improved to better meet the demands of future safety systems. For instance, more accurate long term predictions would be most useful for a decision making algorithm.

A main weakness in traditional models is that they do not capture the influence the driver has on the vehicle motion. Under normal conditions, it is reasonable to believe that the vehicle is completely controlled by the driver. Attempts to model driver behavior have previously appeared in the literature. In [6], a driver behavior model is used to design an adaptive cruise controller, and in [7], similar ideas have been used for threat assessment. Other examples of driver behavior models have been developed for micro traffic simulators [8] and in the ongoing European Union projects integrated human modeling and simulation to support human error risk analysis of partially autonomous driver assistance systems (ISI-PADAS) [9] and Information Technology for error remediation and trapping emergencies (ITERATE) [10]. A closely related problem exists in the field of robotics, where algorithms are designed to find a collision-free path through a partly unknown environment, see, e.g., [11] and the references therein. We believe that neither of these models fulfill the requirements from a future active safety system. For instance, the models in [6] and [11] are not suitable from a tracking perspective as they do not provide a stochastic description of the vehicle motion. Similarly, the idea in [7] is not well suited for decision making as it does not capture the variations in driving styles and driving preferences among different drivers. Finally, since the models for micro process simulators are designed for simulation purposes, they lack a description of uncertainties in predictions as well as the ability to adapt model parameters to observed data. These properties are essential for filtering and decision making purposes and

Manuscript received September 9, 2010; revised March 31, 2011; accepted April 22, 2011. Date of publication July 22, 2011; date of current version December 5, 2011. This work was supported by the Swedish Intelligent Vehicle Safety Systems (IVSS) program and was a part of the SEnsor Fusion for Safety Systems (SEFS) project. The Associate Editor for this paper was H. Dia.

J. Sörstedt and L. Hammarstrand are with Volvo Car Corporation, 405 31 Gothenburg, Sweden (e-mail: jsorsted@volvocars.com; lhammar5@volvocars.com).

L. Svensson is with the Department of Signals and Systems, Chalmers University of Technology, 412 96 Gothenburg, Sweden (e-mail: lennart.svensson@chalmers.se).

F. Sandblom is with Volvo 3P, 405 08 Gothenburg, Sweden (e-mail: fredrik.sandblom.2@volvo.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2160342

make these models inappropriate alternatives in their current form. The same arguments also hold for the models so far presented in the ISI-PADAS and ITERATE projects.

In this paper, we continue to develop the model framework introduced in [12] and [13]. The idea is that the driver controls the vehicle by making a tradeoff between different objectives, such as the desire to travel fast and to travel comfortably. To describe the expected driver control signal, we use a cost function that reflects the driver objectives. By minimizing this cost function, the most likely control signal can be calculated. Compared with [12] and [13], the model framework is extended here to cover a more generic model structure. More importantly, it is also adapted to handle uncertainties in the cost function parameters. This allows us to design models that not only contain knowledge regarding typical driver behavior but also able to capture variations among different drivers and to adapt as driver behavior changes.

The main contribution of this paper is the model structure described in Section III. This structure suggests a method for including the expected driver input in a model structure and provides guidelines on how to construct cost functions. In contrast to other commonly used methods, our approach has the advantage that model parameters have a clear interpretation. Having said that, a significant amount of work still remains before this approach can be used by an application. First of all, the cost function needs to be adjusted and validated for the intended application. Second, an efficient implementation for minimizing the cost function has to be developed. To illustrate the performance of the suggested model structure, we present one cost function and evaluate it using real measurements. In the studied examples, the proposed model explains the data better and provides better predictions compared with a constant acceleration (CA) model.

This paper is organized as follows: Section II presents the notation and some of the most commonly used motion models in tracking applications. In Section III, a general description of the proposed modeling framework is provided. A specific design of the cost function is described in Section IV and later evaluated on measured data in Section V. Finally, Section VI contains the conclusions.

II. Background

The vehicle motion model describes the evolution of a time-discrete vector $\mathbf{x}_k \in \mathcal{R}^{n_x}$ that is referred to as the state vector of the model. For vectors and matrices (indicated by boldface), we use the subindex k as notation for a discrete-time instant with a continuous counterpart in t_k, whereas the notation $x(k)$ is used for scalars. The continuous time interval between two samples is constant and denoted as $T_s = t_k - t_{k-1}$. We consider a state vector partitioned as

$$\mathbf{x}_k = \left[(\mathbf{x}_k^h)^T \ (\mathbf{z}_k^t)^T \ \mathbf{r}_k^T\right]^T \tag{1}$$

where $\mathbf{x}_k^h$ is the state vector for the host vehicle, $\mathbf{z}_k^t$ is the state vector for the nearest target vehicle in front of the host vehicle, and the vector $\mathbf{r}_k$ contains parameters describing the road. Our main interest in this paper is to derive a model for the vector $\mathbf{x}_k^h$. For $\mathbf{z}_k^t$ and $\mathbf{r}_k$, we use standard models, which are described in this section. The model we propose for the host vehicle can also be used for one or several target vehicles, but for simplicity, we apply it only on the host vehicle.

We restrict the scope to Markovian models of the form

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{e}_{k-1}) \tag{2}$$

where $\mathbf{f}_{k-1}(\cdot)$ is a possibly time-varying and nonlinear function of $\mathbf{x}_{k-1}$ and a stochastic noise vector $\mathbf{e}_{k-1} \in \mathcal{R}^{n_e}$. The noise process accounts for model uncertainties and is traditionally modeled as an independent identically distributed process with zero mean.

An important subgroup of the general model structure (2) is the family of linear motion models defined by the matrices $\mathbf{A}_{k-1} \in \mathcal{R}^{n_x \times n_x}$ and $\mathbf{B}_{k-1} \in \mathcal{R}^{n_x \times n_e}$, such as

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{e}_{k-1}. \tag{3}$$

These motion models are popular in tracking applications because they enable simple and efficient algorithms such as the Kalman filter [14] to estimate and predict the states.

A frequently used linear motion model is the CA model. In this model, the state vector $\mathbf{z}_k^t$ is parameterized as

$$\mathbf{z}_k^t = \left[\eta_x^t(k) \ \eta_y^t(k) \ \dot{\eta}_x^t(k) \ \dot{\eta}_y^t(k) \ \ddot{\eta}_x^t(k) \ \ddot{\eta}_y^t(k)\right]^T \tag{4}$$

where $(\eta_x^t(k), \eta_y^t(k))$ is the center position of the vehicle, and $(\dot{\eta}_x^t(k), \dot{\eta}_y^t(k))$ and $(\ddot{\eta}_x^t(k), \ddot{\eta}_y^t(k))$ are the first and second order time derivatives of $\eta_x^t(k)$ and $\eta_y^t(k)$. In the version of the CA model employed here, the noise process $\mathbf{e}_k$ determines the jerk between sample instances. The acceleration, velocity, and position increments are thus given by $\mathbf{e}_k T_s$, $\mathbf{e}_k T_s^2/2$ and $\mathbf{e}_k T_s^3/6$, respectively. The system matrices then take the form

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T_s & 0 & \frac{T_s^2}{2} & 0 \\ 0 & 1 & 0 & T_s & 0 & \frac{T_s^2}{2} \\ 0 & 0 & 1 & 0 & T_s & 0 \\ 0 & 0 & 0 & 1 & 0 & T_s \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{T_s^3}{6} & 0 \\ 0 & \frac{T_s^3}{6} \\ \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \tag{5}$$

where the noise process is modeled as $\mathbf{e}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_e)$, where $\mathbf{C}_e = \mathrm{diag}[\sigma_{\ddot{\eta}_x}^2 \ \sigma_{\ddot{\eta}_y}^2]$.

In essence, the CA model is a kinematic particle model [15], where the object motion is decoupled in different dimensions. Other examples, where the object's motion in the different dimensions is instead coupled, are the coordinated turn models [3], [4]. More detailed vehicle motion models, which, for instance, take wheel slip and wheel angle into account, have been derived for applications such as vehicle stability and traction control. In [16], more information is provided on dynamic vehicle models and describes the single track model or bicycle model that has been used for vehicle tracking, at least in a simplified form; see, e.g., [17].

In this paper, we use a curved road coordinate system such that the positions of the host vehicle (η_x^h, η_y^h) and the target vehicle (η_x^t, η_y^t) are given relative to the road. This coordinate system, which was previously used, e.g., in [18], is defined

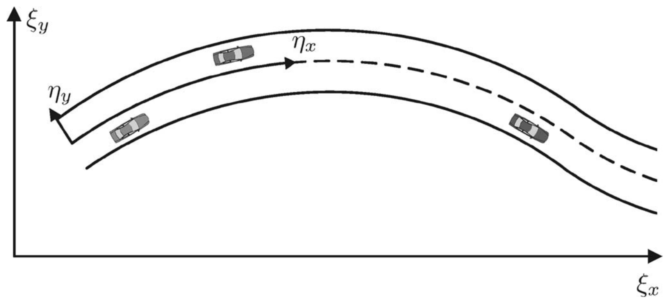

Fig. 1. Global Cartesian coordinate system (ξ_x, ξ_y) and curved road coordinate system (η_x, η_y).

in Fig. 1, which also contains a Cartesian coordinate system (ξ_x, ξ_y) that will be needed later in Section IV. To find a relation between these coordinate systems, a model of the road curvature is required. Here, we follow suggestions from, e.g., [19] and use a clothoid model. The road curvature is then described by a local linear approximation about the host vehicle, i.e.,

$$c\left(\eta_x^h + \Delta\eta_x\right) = c_0 + \Delta\eta_x c_1. \tag{6}$$

The parameter c_0 is the local curvature at the center of the road, and c_1 is the curvature change rate as a function of distance $\Delta\eta_x$ from the host vehicle. We include the clothoid parameters $c_0(k)$ and $c_1(k)$ and the lane width $L_w(k)$ in the road state vector

$$\mathbf{r}_k = \left[L_w(k)\ c_0(k)\ c_1(k)\right]^T. \tag{7}$$

The evolution of $\mathbf{r}_k$ is modeled as

$$\mathbf{r}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & T_s\dot{\eta}_x(k-1) \\ 0 & 0 & 1 \end{bmatrix} \mathbf{r}_{k-1} + \mathbf{v}_{k-1}^r \tag{8}$$

where $\mathbf{v}_{k-1}^r \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_r})$.

III. Motion Model Framework

In this section, we describe how the motion model is extended to incorporate the effect of the driver. We base the modeling idea on a set of postulates that are highlighted in the text. The equations for calculating the driver input are also presented, and important model properties are discussed.

A. *Model Structure*

We will here motivate and present the structure and some key components in the proposed model.

Postulate 1. *The driver controls the motion of the vehicle by steering and adjusting the acceleration.*

As the classical motion models fail to acknowledge the influence of the driver, there is, at least theoretically, room for substantial improvements if we can model the driver actions. Naturally, the vehicle obeys the laws of physics, and the models in Section II are still appropriate although the input is no longer white noise. In a general parameterization, we suggest a model

$$\mathbf{x}_k^h = \mathbf{f}_{k-1}\left(\mathbf{x}_{k-1}^h, \mathbf{u}_{k-1}(\mathbf{x}_{k-1}), \mathbf{v}_{k-1}\right) \tag{9}$$

where $\mathbf{u}_{k-1}(\mathbf{x}_{k-1})$ is the driver input signal, and $\mathbf{v}_{k-1}$ is a white noise process. Modeling the driver input $\mathbf{u}_{k-1}(x_{k-1})$ is a key component here and the topic of Sections III-B and IV.

The complete state vector for the host vehicle is given by $\mathbf{x}_k^h$, but the kinematic state of the host vehicle is described by the vector $\mathbf{z}_k^h$, which is a subvector of $\mathbf{x}_k^h$. The propagation of this kinematic state vector is described by the model

$$\mathbf{z}_k^h = \mathbf{f}_{k-1}^h\left(\mathbf{z}_{k-1}^h, \mathbf{u}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}^z\right). \tag{10}$$

Here, we use a modified CA model

$$\mathbf{z}_k^h = \mathbf{A}\mathbf{z}_{k-1}^h + \mathbf{B}\left(\mathbf{u}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}^z\right) \tag{11}$$

where

$$\mathbf{u}_{k-1}(\mathbf{x}_{k-1}) = \begin{bmatrix} \dddot{\eta}_x^h(k) \\ \dddot{\eta}_y^h(k) \end{bmatrix} \tag{12}$$

and $\mathbf{v}_{k-1}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_z})$. There are other model structures, like the bicycle model, which often describe vehicle motion more accurately than the chosen CA model. However, as we will see, the CA model leads to a relatively simple implementation.

There is reason to believe that the variations in behavior between different drivers can be large. We therefore allow the function $\mathbf{u}_{k-1}(\mathbf{x}_{k-1})$ to be parameterized by a vector $\boldsymbol{\theta}_{k-1}$, which is intended to capture variations in driver preferences and driving styles. For an example of what parameters $\boldsymbol{\theta}_{k-1}$ may contain, see Section IV.

The evolution of $\boldsymbol{\theta}_k$ is described by the model

$$\boldsymbol{\theta}_k = \mathbf{g}_{k-1}\left(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}^\theta\right) \tag{13}$$

i.e., it is assumed to be independent of all the other elements in $\mathbf{x}_{k-1}^h$. We use

$$\mathbf{g}_{k-1}\left(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}^\theta\right) = \gamma\boldsymbol{\theta}_{k-1} + (1-\gamma)\boldsymbol{\mu} + \mathbf{v}_{k-1}^\theta \tag{14}$$

where $\mathbf{v}_{k-1}^\theta \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\theta)$, whereas $\gamma \in [0, 1)$, $\boldsymbol{\mu}$, and $\mathbf{Q}_\theta$ are design parameters. The proposed process model is such that when an element in $\boldsymbol{\theta}_{k-1}$ is not observable, its posterior distribution approaches a prior distribution representing the set of all drivers. To illustrate the convergence of the first two moments, we study

$$E[\boldsymbol{\theta}_{k+l}] = \gamma^l E[\boldsymbol{\theta}_k] + \boldsymbol{\mu}(1-\gamma^l) \tag{15}$$

$$Cov[\boldsymbol{\theta}_{k+l}] = \gamma^{2l} Cov[\boldsymbol{\theta}_k] + \frac{1-\gamma^{2l}}{1-\gamma^2}\mathbf{Q}_\theta. \tag{16}$$

Thus, γ determines the convergence speed, whereas $\boldsymbol{\mu}$ and $\mathbf{Q}_\theta/(1-\gamma^2)$ are the limit values of the first two moments as the prediction length $l \to \infty$.

The task of modeling driver behavior in terms of $\mathbf{u}_{k-1}(\mathbf{x}_{k-1})$ and $\mathbf{v}_{k-1}$ is an extensive and difficult problem. To simplify the problem, we assume that it is divided into subproblems for which the uncertainties are smaller and $\mathbf{v}_{k-1}$ has a unimodal distribution.

Postulate 2. *In normal traffic, the actions of a driver can be divided into a set of different categories.*

Each category typically corresponds to a driver intention, such as changing lanes, following the lane, or turning left, but it could also capture other aspects of the situation, such as if the driver is distracted or not. Given the intention of the driver, it is easier to capture the driver behavior. We use a scalar parameter $m(k)$ to indicate the driver action that is currently active.

The complete state vector for the host vehicle is now given by

$$\mathbf{x}_k^h = \left[(\mathbf{z}_k^h)^T \; \boldsymbol{\theta}_k^T \; m(k) \right]^T. \tag{17}$$

Note that the function $\mathbf{u}_{k-1}(\mathbf{x}_{k-1})$ depends on both $\boldsymbol{\theta}_k$ and $m(k)$, and the implications of this are further discussed in Section III-C.

B. Calculating the Driver Input $\mathbf{u}_k(\mathbf{x}_k)$

In the suggested framework, the driver operates the vehicle such that three main objectives, or driving rules, are fulfilled.

Postulate 3. *The driver strives to control the car in a safe and comfortable manner and at a preferred velocity.*

The safety preference corresponds to the driver's desire to control the vehicle such that it stays in the preferred lane and at a safe distance to other vehicles and objects. The distance is here both spatial and temporal measures. Simultaneously, the driver asks for a comfortable journey and is reluctant to be put under large accelerating forces and jerks. The third objective captures the driver's desire to maintain a preferred velocity.

Based on this assumption, we can mathematically represent the driver preferences using a cost function

$$\text{cost}(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}). \tag{18}$$

Suggestions for cost functions that penalize trajectories that are in conflict with the foregoing preferences are given in Section IV. As we will see, the cost function is implicitly dependent on the vector $\boldsymbol{\theta}_{k-1}$.

An important driver ability is to make decisions based on both the present and upcoming situations.

Postulate 4. *The driver plans ahead and tries to find the optimal route such that the driver preferences are considered for the near—and not only the immediate—future.*

We wish to incorporate this property in our model such that the expected driver input at time $k-1$ enables a desirable journey also for the near future. The cost function is therefore evaluated for a trajectory of driver input signals and corresponding state vectors. By selecting the trajectory with the lowest cost, we can produce a prediction of the vehicle motion that incorporates knowledge regarding the road, other objects, speed limits, etc.

In a sense, we are hereby approximating the driver by an optimal controller [20], where the optimality criterion is defined by the cost function. The optimal driver input is obtained as $\mathbf{u}_{k-1}(\mathbf{x}_{k-1}) \triangleq$

$$\underset{\mathbf{u}_{k-1}}{\arg\min} \quad \underset{\{\mathbf{u}_k, \ldots, \mathbf{u}_{k+N-2}\}}{\min} \sum_{n=k-1}^{k+N-2} \text{cost}(\mathbf{u}_n, \mathbf{x}_n) \tag{19}$$

where the future state vectors $\mathbf{x}_n$, $n > k-1$ are calculated using the model (9). The driver input $\mathbf{u}_{k-1}$ is usually a nonlinear function of the state $\mathbf{x}_{k-1}$, which turns the modified CA model in (11) into a nonlinear motion model. Here, we assume that the driver completely controls the car such that there are no uncertainties in the motion model; we set $\mathbf{v}_n = \mathbf{0}$ for $n = k-1, \ldots, k+N-3$.

To solve (19), we take an approach that is very common in optimal control and that is advocated, e.g., in [21]. Instead of parameterizing the problem in the free variables $\mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k+N-2}$, we solve for the optimal sequence of state vectors $\mathbf{z}_k, \ldots, \mathbf{z}_{k+N-2}$. The motion model (11) (with $\mathbf{v}_{n-1}^z = \mathbf{0}$ for $n = k, \ldots, k+N-2$) now enters as linear constraints on the state sequence. As a consequence, the optimization problem becomes more high dimensional but with much nicer properties. In this alternative form, the optimization can conveniently be solved using a built-in routine in TOMLAB (http://tomopt.com/tomlab/) called SNOPT. SNOPT solves the optimization problem using an efficient sequential quadratic programming method [22]. Note, however, that depending on the choice of cost function, the optimization problem (19) may be of different complexity. The properties of the optimization problem hence need to be considered when designing the cost functions. For the cost functions presented in Section IV, which are designed so that their first and second derivatives both exist and are continuous, and for the test scenario described in Section V, SNOPT computes the optimal trajectory in less than 50 ms on a conventional computer.

C. Model Features

The idea of approximating the driver as an optimal controller according to an objective function has previously been discussed, e.g., in [6]. An important difference in the new motion model is the inclusion of the parameters $\boldsymbol{\theta}_k$ and $m(k)$. In this section, we discuss the vital role these parameters have for situation assessment and to produce reliable predictions.

1) Driver Preferences $\boldsymbol{\theta}_k$: The purpose of the parameter $\boldsymbol{\theta}_k$ is to capture the individual driving style and allow the driver preferences to vary over time. Naturally, additional parameters add complexity to the problem, but the benefits obtained by introducing the vector $\boldsymbol{\theta}_k$ are arguably more significant.

First of all, as time goes by, the system gains information regarding $\boldsymbol{\theta}_k$ by tracking the vehicle trajectory $\mathbf{z}_1^h, \mathbf{z}_2^h, \ldots, \mathbf{z}_k^h$. Due to improved knowledge about $\boldsymbol{\theta}_k$, predictions become more accurate, and biases are reduced. The objective is, of course, to attain this performance for (virtually) all driving styles.

Second, by describing the uncertainties in $\boldsymbol{\theta}_k$, we can obtain a parameterized and improved description of the uncertainties in $\mathbf{z}_k^h$. While the cost function framework provides an estimate of the expected control signal, the random vector $\boldsymbol{\theta}_k$ also affects the variance of $\mathbf{u}_{k-1}(x_{k-1})$. This is a key advantage of the model framework, which owes to the fact that the uncertainties in $\boldsymbol{\theta}_k$ can be propagated to $\mathbf{z}_k^h$ according to (10). In situations where the optimal trajectory is essentially independent of $\boldsymbol{\theta}_k$, i.e., when most drivers would act the same, the predictions are more reliable. On the other hand, in other circumstances, the optimal trajectory may be highly dependent on the driving style, which would lead to a lower confidence in the predictions as $\boldsymbol{\theta}_k$ is not fully known. As we will see in Section V, the variations in prediction uncertainties coincide fairly well with the observed prediction errors.

2) Driver Intention $m(k)$: It may seem like a drawback to include the hypotheses parameter $m(k)$ into the state vector as it leads to several difficulties. First, to provide a complete description of the model in (9), one must design a process model for $m(k)$ given by the probability function $P(m(k)|\mathbf{x}_{k-1})$. A second complication is that a predetermined list of possible hypotheses is also needed; of course, depending on the position of the vehicles, not all of these hypotheses may be likely. Nonetheless, for many applications, it is also potentially the greatest advantage with the framework since it enables straightforward and formal derivations of the posterior distribution of $m(k)$, as demonstrated in [13]. In particular, once the intention specific motion models have been developed, information regarding $m(k)$ is extracted simply by examining the past trajectories of the vehicles. Thereby, one may obtain a powerful tool to draw conclusions regarding $m(k)$ and, thus, the traffic situation.

However, it is beyond the scope of this paper to fully develop and illustrate cost functions for multiple intentions. Thus, although we would like to point out the potential gains with including $m(k)$, it is not properly investigated here. The state vector used in the remainder of this paper is therefore reduced to $\mathbf{x}_k^h = [(\mathbf{z}_k^h)^T \boldsymbol{\theta}_k^T]^T$.

IV. Designing Cost Functions

The cost functions should be designed to capture the typical behavior of the driver. We have partly based our choices on behavior studies used to design roads [23] and common practice for driving taught at driving schools. As was argued in Postulate 2, the actions of the driver can be separated into different categories. Each of these will typically require different cost functions. In this paper, we focus on the hypothesis that the driver intends to follow the right lane of the road. Designing cost functions to describe the normal driver behavior is different compared with most previous contributions, which have been focused on predicting dangerous situations [6], [24], [25].

We divide the cost function into four different components related to the longitudinal velocity $c_{lo}(\mathbf{u}_n, \mathbf{x}_n)$, the lateral positioning of the vehicle $c_{la}(\mathbf{u}_n, \mathbf{x}_n)$, the comfort of the trajectory $c_c(\mathbf{u}_n, \mathbf{x}_n)$, and the interaction with other vehicles $c_{in}^1(\mathbf{u}_n, \mathbf{x}_n)$ and $c_{in}^2(\mathbf{u}_n, \mathbf{x}_n)$. Among these, $c_{la}(\mathbf{u}_n, \mathbf{x}_n)$, $c_{in}^1(\mathbf{u}_n, \mathbf{x}_n)$, and $c_{in}^2(\mathbf{u}_n, \mathbf{x}_n)$ cover the safety aspect discussed in Postulate 3. Similarly, $c_c(\mathbf{u}_n, \mathbf{x}_n)$ ensures that the trajectory is comfortable and $c_{lo}(\mathbf{u}_n, \mathbf{x}_n)$ that the preferred speed of the driver is maintained. A weighted sum of the different components constitutes the total cost

$$\begin{aligned}\text{cost}(\mathbf{u}_n, \mathbf{x}_n) = \alpha_{lo} c_{lo}(\mathbf{u}_n, \mathbf{x}_n) + \alpha_{la} c_{la}(\mathbf{u}_n, \mathbf{x}_n)\\ + \alpha_c c_c(\mathbf{u}_n, \mathbf{x}_n) + \alpha_{in} c_{in}(\mathbf{u}_n, \mathbf{x}_n) \quad (20)\end{aligned}$$

where the parameters α_{lo}, α_{la}, α_c, and α_{in} decide how the driver gives priority to the different costs.

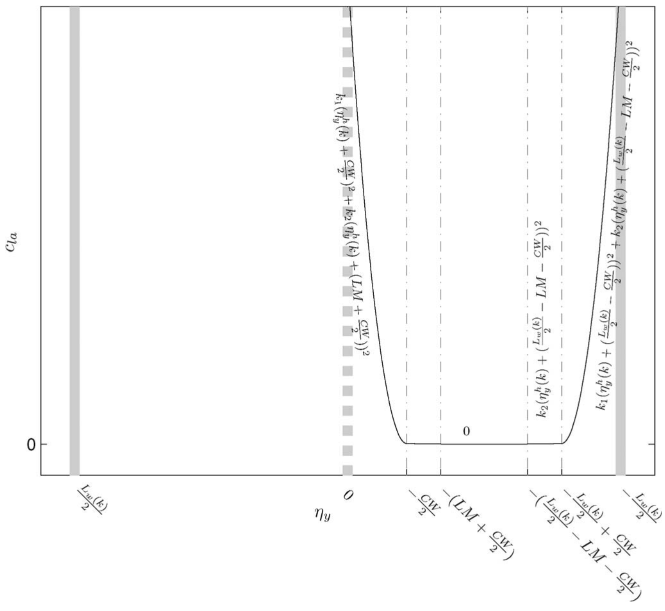

Fig. 2. Lateral cost function for following the right lane. The figure specifies the mathematical expressions for the different road regions.

As mentioned in previous sections, driving preferences and thus driving style are characterized by the vector $\boldsymbol{\theta}_k$. In our implementation, $\boldsymbol{\theta}_k$ determines, not only α_{lo} and α_{in}, but the parameters $\dot{\eta}_{\text{x-ref}}^h(k)$ and $\bar{t}_h(k)$ related, respectively, to $c_{lo}(\mathbf{u}_n, \mathbf{x}_n)$ and $c_{in}(\mathbf{u}_n, \mathbf{x}_n)$ subsequently defined as well. We will return to the parameterization of $\boldsymbol{\theta}_k$ and its process model in Section V-B. The remaining weight parameters are assumed constant and are given the values $\alpha_{la} = \alpha_c = 1$.

A. Longitudinal Cost Function $c_{lo}(\cdot, \cdot)$

The longitudinal cost function reflects the driver's desire to maintain a specific speed given by the parameter $\dot{\eta}_{\text{x-ref}}^h$. To punish trajectories for which the speed of the host vehicle deviates from the reference speed, we use

$$c_{lo}(\mathbf{u}_n, \mathbf{x}_n) = \left(\dot{\eta}_x^h(n) - \dot{\eta}_{\text{x-ref}}^h(n)\right)^2. \quad (21)$$

If this cost function is omitted, the optimal driver tends to slow down as that leads to a safe and comfortable journey. Like the other parameters in $\boldsymbol{\theta}_k$, $\dot{\eta}_{\text{x-ref}}^h(k)$ is partially unknown initially. However, $\dot{\eta}_{\text{x-ref}}^h(k)$ is special since it sometimes varies significantly over time, e.g., when the vehicle exits or enters a highway.

B. Lateral Cost Function $c_{la}(\cdot, \cdot)$

The objective with the lateral cost function is to associate large costs to trajectories outside the driver's preferred lane; a cost related to the desire to drive safely. Fig. 2 defines how $c_{la}(\mathbf{u}_n, \mathbf{x}_n)$ depends on the lateral position η_y. The background in Fig. 2 consists of a road portrayed with thick grey lines divided into different cost segments. In the expressions, CW is the width of the car, and LM is a margin to the lane edge, whereas k_1 and k_2 are design parameters used to tune the cost function. We use $k_1 = 1000$ and $k_2 = 3$.

C. Comfort Cost Function $c_c(\cdot,\cdot)$

The comfort cost corresponds to the driver's desire to steer the vehicle in a smooth and comfortable manner. Smooth trajectories are accomplished by increasing the cost for large accelerations and jerks. However, the acceleration perceived by the driver is not equal to the acceleration of the vehicle expressed in curved road coordinates. Instead, the global Cartesian coordinate system (ξ_x, ξ_y) is used. It is easy to construct a nonlinear mapping $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ from road coordinates to Cartesian coordinates, i.e., T is such that $T(\eta_x, \eta_y) = [\xi_x \ \xi_y]^T$. Using the mapping T, it is also possible to compute $(\ddot{\xi}_x^h(n), \ddot{\xi}_y^h(n))$ and $(\dddot{\xi}_x^h(n), \dddot{\xi}_y^h(n))$ from $\mathbf{z}_n^h$, $\mathbf{r}_n$, and $(\dddot{\eta}_x^h(n), \dddot{\eta}_y^h(n))$.

To simplify the expressions, we introduce the notation $\ddot{\xi}^h(n) = \sqrt{\ddot{\xi}_x^h(n)^2 + \ddot{\xi}_y^h(n)^2}$ and $\dddot{\xi}^h(n) = \sqrt{\dddot{\xi}_x^h(n)^2 + \dddot{\xi}_y^h(n)^2}$. The cost function is divided into two parts

$$c_{\mathrm{c}}\left(\dddot{\eta}_x(n), \dddot{\eta}_y^h(n), \mathbf{z}_n^h, \mathbf{r}_n\right) = c_{\mathrm{a}}\left(\ddot{\xi}(n)\right) + c_{\mathrm{j}}\left(\dddot{\xi}(n)\right) \tag{22}$$

where $c_{\mathrm{a}}(\cdot)$ is related to the acceleration, and $c_{\mathrm{j}}(\cdot)$ is related to the jerk.

Empirical studies indicate that people are fairly insensitive to jerks and accelerations up to a certain level, above which the motions feel more unpleasant (possibly with the exception of accelerating forces). We use functions $c_{\mathrm{a}}(\cdot)$ and $c_{\mathrm{j}}(\cdot)$ of the form

$$c(x) = \begin{cases} \frac{x^2}{g^2}, & \text{when } x < g \\ \left(\frac{5}{6} + \frac{x^2}{6g^2}\right)^6, & \text{otherwise.} \end{cases} \tag{23}$$

As a consequence, we get cost functions that increase more rapidly for $x > g$ and have continuous derivatives. Based on information from [23], we set $g = 2$ for $c_{\mathrm{a}}(\cdot)$ and $g = 1.5$ for $c_{\mathrm{j}}(\cdot)$.

D. Cost Functions for Vehicle Interaction $c_{in}(\cdot,\cdot)$

An important aspect in the proposed motion model is that it not only incorporates the interaction between the vehicle and the road, e.g., in $c_{la}(\mathbf{u}_n, \mathbf{x}_n)$ but the interaction with other vehicles as well. Here, we define a cost function that seeks to penalize trajectories where vehicles are too close. We limit our design in this paper to situations when there is one more vehicle close to the host vehicle. The vehicle interaction cost contains two parts

$$c_{in}(\mathbf{u}_n, \mathbf{x}_n) = c_{in}^1(\mathbf{u}_n, \mathbf{x}_n) + c_{in}^2(\mathbf{u}_n, \mathbf{x}_n) \tag{24}$$

which concern the temporal and spatial distances to the other vehicle, respectively.

To measure the temporal distance, we introduce the time headway

$$t_h(k) = \frac{\eta_x^t(k) - \eta_x^h(k)}{\dot{\eta}_x^h(k)} \tag{25}$$

which reflects the time it takes for the host vehicle to reach the current position of the target vehicle. A typical driver is reluctant to allow the time headway to become too small, whereas it matters less if t_h is, say, 4 or 10 s. The cost function

$$c_{\mathrm{int}}^1(\mathbf{u}_n, \mathbf{x}_n) = \begin{cases} \frac{(\bar{t}_h - t_h(k))^2}{t_h(k)}, & \text{if } 0 < t_h(k) \leq \bar{t}_h \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

is selected to enable our model to capture this behavior, and $\bar{t}_h$ is included in the state vector $\boldsymbol{\theta}_k$.

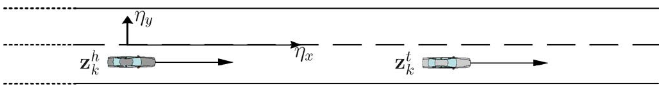

Fig. 3. Test scenario. Host vehicle described by a state vector $\mathbf{z}_k^h$ traveling on a straight road behind a second vehicle described by $\mathbf{z}_k^t$.

The spatial distance is here defined as the distance between the longitudinal positions of the two vehicles

$$d(k) = \eta_x^t(k) - \eta_x^h(k). \tag{27}$$

In this case, we design the cost function for a scenario where the target vehicle is initially positioned ahead of the host vehicle on a single lane road. A reasonable cost function for the spatial distance should be large when $d(k)$ is small (and even larger when $d(k)$ is negative). We use the cost function

$$c_{\mathrm{int}}^2(\mathbf{u}_n, \mathbf{x}_n) = \begin{cases} 0, & \text{if } \overline{d} \leq d \\ \frac{(\overline{d} - d)^2}{(\overline{d} - \underline{d})^2}, & \text{if } \underline{d} \leq d \leq \overline{d} \\ \left(\frac{5}{6} + \frac{(\overline{d} - d)^2}{6(\overline{d} - \underline{d})^2}\right)^6, & \text{if } d \leq \underline{d}. \end{cases} \tag{28}$$

Note the resemblance to (23) if we set $x = \overline{d} - d$ and $g = \overline{d} - \underline{d}$. The parameter values used in our evaluations are $\overline{d} = 5$ and $\underline{d} = 2$.

V. Evaluation Results

To evaluate our model, we study how well it explains (predicts) two different driving sequences. The sequences are real traffic situations collected using the same driver on straight busy two-lane roads at two different places in a city. The host vehicle is positioned in the middle of the road and is traveling behind a target vehicle (see Fig. 3). The two vehicles are closely spaced such that the host vehicle has to slow down when the target vehicle slows down (see Figs. 4 and 5).

The host vehicle is equipped with a 77-GHz radar sensor that measures the range and range rate to the target vehicle. Based on these and internal measurements on the host vehicle speed and acceleration, we estimate the trajectories $\mathbf{Z}_L^h = [\mathbf{z}_1^h, \ldots, \mathbf{z}_L^h]$ and $\mathbf{Z}_L^t = [\mathbf{z}_1^t, \ldots, \mathbf{z}_L^t]$ using a Kalman smoother. As benchmark, we use a CA model also for the host vehicle. For notation, we introduce

$$\begin{aligned} \mathcal{M}_1 : &\quad \text{constant acceleration model} \\ \mathcal{M}_2 : &\quad \text{proposed motion model} \end{aligned}$$

to indicate the choice of model.

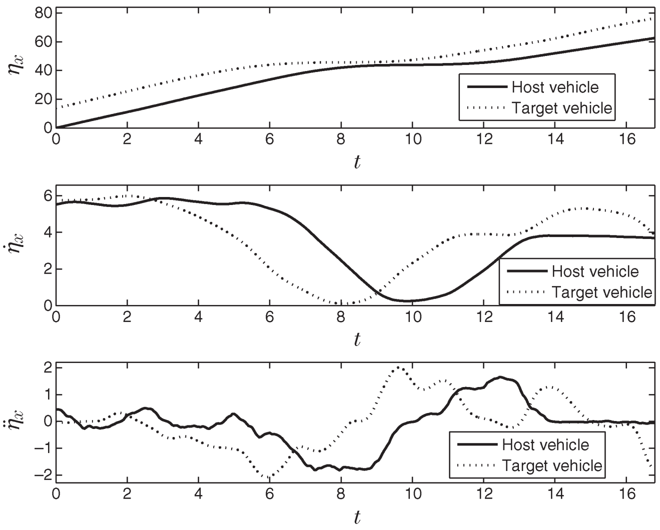

Fig. 4. Positions, velocities, and accelerations of the two vehicles, in the first sequence, as functions of time.

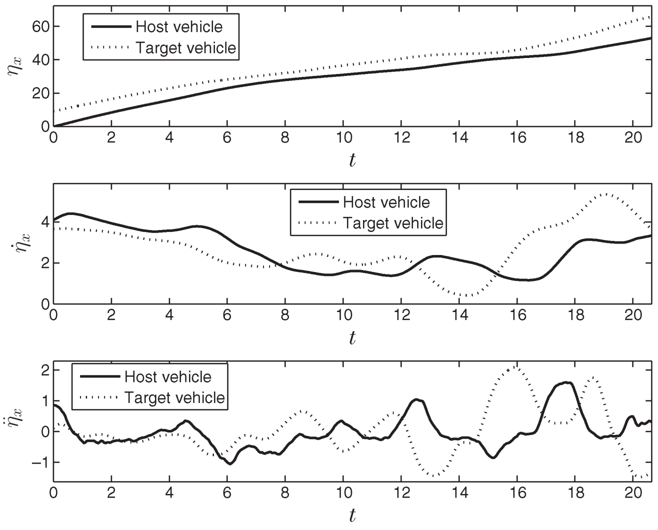

Fig. 5. Positions, velocities, and accelerations of the two vehicles in the second sequence.

A. Evaluation Criteria

We will mainly study two different properties of the model, where the first concerns the prediction capability and the variations in prediction uncertainties. More specifically, we will compare the mean and the variance of the density[1] $p(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_{k-1}^t, \mathcal{M}_i)$ for $i = 1$ and 2 to the observed trajectories.

The second property is the model likelihood

$$p\left(\mathbf{Z}_L^h, \mathbf{Z}_L^t|\mathcal{M}_i\right) = p\left(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_i\right) p\left(\mathbf{Z}_L^t|\mathcal{M}_i\right) \tag{29}$$

which is a standard criterion in model testing (see [2]). As we are mainly interested in the ratio between the two likelihoods, our subsequent evaluations study $p(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_i)$ since $p(\mathbf{Z}_L^t|\mathcal{M}_i)$ does not depend on $\mathcal{M}_i$.

B. Filtering $\boldsymbol{\theta}_k$

In this section, we describe how $\boldsymbol{\theta}_k$ is treated in the evaluation.

1) Motivation: We notice that both criteria involve the computation of densities of the host state vector either $p(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_{k-1}^t, \mathcal{M}_i)$ or $p(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_i)$. For the CA model, these densities are completely determined by the process noise covariance matrix $\mathbf{C}_e = \text{diag}[\sigma^2_{\dddot{\eta}_x}\ \sigma^2_{\dddot{\eta}_y}]$. In the evaluations, we select $\sigma^2_{\dddot{\eta}_x}$ and $\sigma^2_{\dddot{\eta}_y}$ to maximize $p(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_1)$.

For the proposed motion model, these densities are, in principal, evaluated in the same way. We will describe the evaluation steps for the density $p(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_2)$ first. Later, at the end of Section V-B3, we explain how the algorithm can be adapted to instead compute $p(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_{k-1}^t, \mathcal{M}_2)$. To simplify the calculations, we decompose the likelihood as

$$p\left(\mathbf{Z}_L^h|\mathbf{Z}_L^t, \mathcal{M}_2\right) = \prod_{k=1}^{L} p\left(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2\right) \tag{30}$$

and our primary goal is, therefore, to find $p(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2)$. Note that this expression is not conditioned on $\boldsymbol{\theta}_k$. By marginalizing the dependence on $\boldsymbol{\theta}_k$, the variance of $\boldsymbol{\theta}_k$ will influence the uncertainties in the predictions of $\mathbf{z}_k^h$.

To evaluate

$$p\left(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2\right) = \int p\left(\mathbf{z}_k^h|\boldsymbol{\theta}_{k-1}, \mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2\right) \times p\left(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2\right) d\boldsymbol{\theta}_{k-1} \tag{31}$$

we approximate $p(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2)$ as a Gaussian density and then apply the unscented transform [26] to obtain a Gaussian approximation of (31). In Section V-B3, we describe how to obtain $p(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2)$ using the unscented Kalman filter (UKF), but first we describe $\boldsymbol{\theta}_k$ and its model in more detail.

2) Parameterization of $\boldsymbol{\theta}_k$: The time evolution of $\boldsymbol{\theta}_k$ was already described in (13) and (14). We have previously mentioned that $\boldsymbol{\theta}_k$ defines α_{lo}, α_{in}, $\dot{\eta}^h_{\text{x-ref}}(k)$, and $\bar{t}_h(k)$, and we now wish to specify how. For some of the parameters, it appears equally likely that they should increase by, say 30%, or decrease by the same amount. For the process model in (13) and (14) to agree with this, we use the parameterization

$$\boldsymbol{\theta}_k = \begin{bmatrix} \log \frac{\alpha_{lo}(k)}{\tilde{\alpha}_{lo}} \\ \log \frac{\alpha_{in}(k)}{\tilde{\alpha}_{in}} \\ \log \frac{\bar{t}_h(k)}{\tilde{t}_h} \\ \dot{\eta}^h_{\text{x-ref}}(k) \end{bmatrix} \tag{32}$$

where $\tilde{\alpha}_{lo} = 1$, $\tilde{\alpha}_{in} = 70$, and $\tilde{t}_h = 2$. The other model parameters are $\boldsymbol{\mu} = [0\ \ 0\ \ 0\ \ 6]^T$, $\gamma = 0.95^{T_s}$, and $\mathbf{Q}_\theta = \text{diag}[1\ 1\ 1\ 0.08]/80$.

3) Filter Recursion: Here, we present a recursive algorithm to compute $p(\boldsymbol{\theta}_k|\mathbf{Z}_k^h, \mathbf{Z}_L^t, \mathcal{M}_2)$ from $p(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_{k-1}^h, \mathbf{Z}_L^t, \mathcal{M}_2)$. We use the UKF, which is a Kalman-like technique to

[1] In this section, all densities are conditioned on the initial states $\mathbf{z}_0^h$ and $\mathbf{z}_0^t$, even though these variables are omitted for brevity.

approximate the updated density[2] $p(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_k^h,\mathbf{Z}_L^t,\mathcal{M}_2)$ and the predicted density $p(\boldsymbol{\theta}_k|\mathbf{Z}_k^h,\mathbf{Z}_L^t,\mathcal{M}_2)$.

In the update step, we adjust the prior density with the information obtained from the likelihood function[3]

$$p\left(\mathbf{z}_k^h|\boldsymbol{\theta}_{k-1},\mathbf{z}_{k-1}^h,\mathbf{Z}_L^t\right) \propto\mathcal{N}\left(\mathbf{B}^\dagger\left[\mathbf{z}_k^h-\mathbf{A}\mathbf{z}_{k-1}^h\right];\mathbf{u}_{k-1}\left(\boldsymbol{\theta}_{k-1},\mathbf{z}_{k-1}^h,\mathbf{Z}_L^t\right),\mathbf{Q}_z\right). \qquad (33)$$

In (33), we have used the singularity of $p(\mathbf{z}_k^h|\boldsymbol{\theta}_{k-1},\mathbf{z}_{k-1}^h,\mathbf{Z}_L^t)$ (there are only uncertainties in some dimensions) to reduce the dimension by multiplying with the Moore–Penrose pseudoinverse $\mathbf{B}^\dagger$. To employ the UKF, it is more convenient to express (33) as

$$\mathbf{B}^\dagger\left[\mathbf{z}_k^h-\mathbf{A}\mathbf{z}_{k-1}^h\right]=\mathbf{u}_{k-1}\left(\boldsymbol{\theta}_{k-1},\mathbf{z}_{k-1}^h,\mathbf{Z}_L^t\right)+\mathbf{v}_k^z \qquad (34)$$

where $\mathbf{B}^\dagger[\mathbf{z}_k^h-\mathbf{A}\mathbf{z}_{k-1}^h]$ is regarded as the measurement equation. The update step of the standard UKF algorithm can now be applied based on (34), see [26] and [27]. We employ the UKF parameterization defined in [27] with $[\alpha\ \beta\ \kappa]=[1.05\ 2\ 0]$.

In the prediction step, we assume that we have a Gaussian prior density

$$p\left(\boldsymbol{\theta}_{k-1}|\mathbf{Z}_k^h,\mathbf{Z}_L^t,\mathcal{M}_2\right)\approx\mathcal{N}(\boldsymbol{\theta}_{k-1};\hat{\boldsymbol{\theta}}_{k-1|k},\mathbf{P}_{k-1|k}). \qquad (35)$$

Since the process model for $\boldsymbol{\theta}_k$, which is described in (14), is linear and Gaussian, the predicted density is also Gaussian, i.e.,

$$p\left(\boldsymbol{\theta}_k|\mathbf{Z}_k^h,\mathbf{Z}_L^t,\mathcal{M}_2\right)\approx\mathcal{N}(\boldsymbol{\theta}_k;\hat{\boldsymbol{\theta}}_{k|k},\mathbf{P}_{k|k}) \qquad (36)$$

where $\hat{\boldsymbol{\theta}}_{k|k}=\gamma\hat{\boldsymbol{\theta}}_{k-1|k}+(1-\gamma)\boldsymbol{\mu}$, and $\mathbf{P}_{k|k}=\gamma^2\mathbf{P}_{k-1|k}+\mathbf{Q}_z$.

From an implementation perspective, the UKF is very convenient to use but has the disadvantage that it requires us to evaluate $\mathbf{u}_{k-1}(\boldsymbol{\theta}_{k-1},\mathbf{z}_{k-1}^h,\mathbf{Z}_L^t)$ several times (once for each sigma point; since $\boldsymbol{\theta}_{k-1}$ contains four elements, we have $2\times 4+1=9$ sigma points).

We have here written the densities conditional on $\mathbf{Z}_L^t$. The calculations when we instead condition on $\mathbf{Z}_k^t$ are analogous but with a slight difference in the update step. As we have seen, the function $\mathbf{u}_{k-1}$ takes $\mathbf{Z}_L^t$ as input to know how to compensate for the vehicle interaction $c_{in}(\cdot,\cdot)$. When $\mathbf{z}_k^t,\mathbf{z}_{k+1}^t,\ldots,\mathbf{z}_L^t$ are unknown, we predict these based on $\mathbf{z}_{k-1}^t$. The approach taken here is to use the predicted mean of the CA model $\hat{\mathbf{z}}_{k-1+l}^t=\mathbf{A}^l\mathbf{z}_{k-1}^t$. (Note that t stands for target, whereas $\mathbf{A}$ is raised to the power of l.)

C. Results

As previously mentioned, we observe two cars driving on a straight road in a city. We evaluate the criteria in Section V-A on two different sequences (see Figs. 4 and 5). Both sequences are interesting since there is an interaction between the two vehicles; the host vehicle is forced to slow down when the target vehicle slows down. Measurement data are obtained with the frequency of 20 Hz. However, to speed up the optimization algorithm, we assume $T_s=0.3$ and a prediction horizon $N=10$ [see (19)] during our calculation of $\mathbf{u}_{k-1}(\cdot)$.

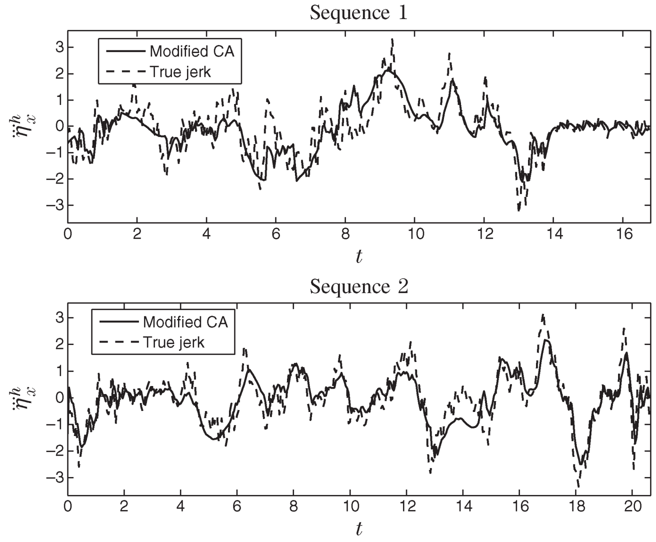

Fig. 6. Predicted and true jerk for the two trajectories.

1) Prediction Accuracy: We now wish to evaluate how well the predicted density $p(\mathbf{z}_k^h|\mathbf{Z}_{k-1}^h,\mathbf{Z}_{k-1}^t,\mathcal{M}_2)$ corresponds to the observed trajectories. The motion model in (11) is driven only by the jerk values between the sample instances. We will therefore compare the predicted jerk values (obtained from $\mathbf{u}_{k-1}(\cdot)$) with the values computed from the measurements. For the CA model, we use the values $\sigma^2_{\dddot{\eta}_x}=0.95$ and $\sigma^2_{\dddot{\eta}_x}=1.08$ for the first and second sequences, respectively, since these values maximize the likelihood (see Section V-B1).

Fig. 6 shows the jerk predictions obtained from our model $E\{\dddot{\eta}_x^h(k)|\mathbf{Z}_{k-1}^h,\mathbf{Z}_{k-1}^t,\mathcal{M}_2\}$, which is denoted here as modified CA, together with the observed jerk values. Note that the jerk predictions provided by the CA model are zero at all times. As we can see, the jerk predictions are fairly accurate most of the time. In the first sequence, the predictions are less accurate at around $t=6$ when the host vehicle temporarily accelerates although the target vehicle slows down. Similarly, for the second sequence, the model incorrectly predicts an increased deceleration at $t\approx 14$. However, overall, the predictions from the modified CA model are substantially better than those from the standard CA model.

For the predicted density to be accurate, it is also important that it has a reasonable variance. Let us denote the error between the predicted and observed jerks by $\Delta\dddot{\eta}_x^h(k)$. Ideally, we would like the variance of the predicted density $\mathrm{Var}\{\dddot{\eta}_x^h(k)|\mathbf{Z}_{k-1}^h,\mathbf{Z}_{k-1}^t,\mathcal{M}_2\}$ to be identical to $(\Delta\dddot{\eta}_x^h(k))^2$ on the average. As previously discussed, we hope that the model also has an ability to know when the predictions are reliable and when they are more uncertain (see Section III-C1).

The variance of the predictions (the uncertainties in the predictions) and the prediction errors are given in Figs. 7 and 8 for the respective data sequences. On the average, the variance

[2] As $\mathbf{z}_k^h$ is related more directly to $\boldsymbol{\theta}_{k-1}$ than to $\boldsymbol{\theta}_k$, the suggested order (first update, then predict) is more straightforward; normally, one would instead first predict, compute $p(\boldsymbol{\theta}_k|\mathbf{Z}_{k-1}^h,\mathbf{Z}_L^t,\mathcal{M}_2)$, and then do the update and calculate $p(\boldsymbol{\theta}_k|\mathbf{Z}_k^h,\mathbf{Z}_L^t,\mathcal{M}_2)$.

[3] In previous sections, $\mathbf{u}_{k-1}(\cdot)$ only took $\mathbf{x}_{k-1}$ as input. However, when $\mathbf{Z}_L^t$ is known, we use it as an additional input. To understand how this influences the computations, see the discussion at the end of Section V-B3.

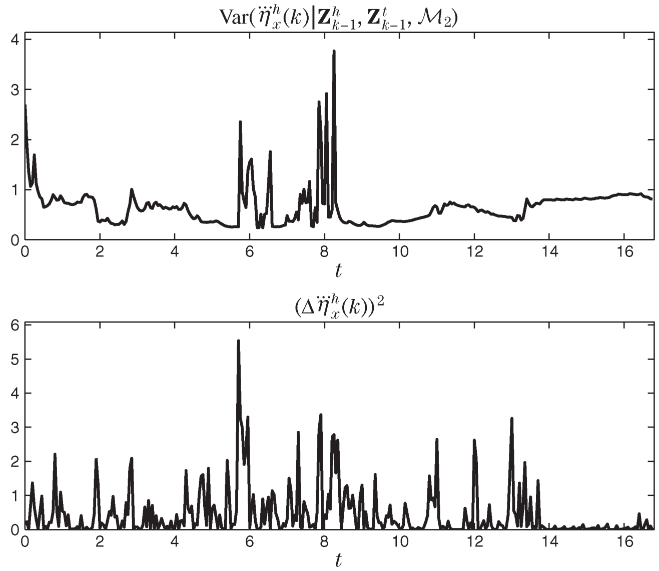

Fig. 7. Prediction uncertainties and prediction errors for the first sequence.

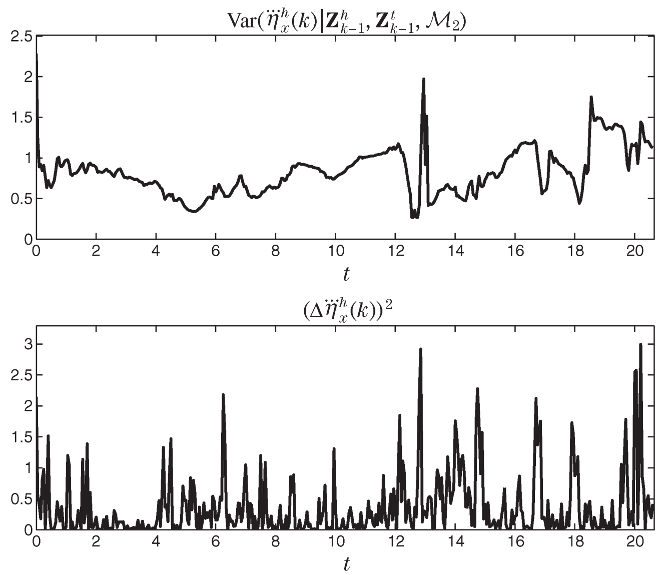

Fig. 8. Prediction uncertainties and prediction errors for the second sequence.

is 0.65 and 0.83 in the first and second sequences, respectively, whereas the corresponding errors $(\Delta\dddot{\eta}_x^h(k))^2$ are 0.45 and 0.36. The model is thus overestimating the uncertainties, but at least the order of magnitude is correct. At several occasions, like at around $t = 6$ and $t = 8$ in the first sequence and at $t \approx 13$ in the second sequence, larger prediction variances coincide with larger errors. Since an increased prediction variance can only be caused by uncertainties in $\boldsymbol{\theta}_k$, this is a good sign in favor of the filter described in Section V-B. Although the evaluation is limited, this is still a promising indication.

2) Model Likelihood: We previously stated that we wish to compute $p(\mathbf{Z}_L^h, \mathbf{Z}_L^t|\mathcal{M}_i)$ for $\mathcal{M}_1$ and $\mathcal{M}_2$ as a measure of how well the models explain the data. However, by studying

$$\log p\left(\mathbf{Z}_k^h|\mathbf{Z}_L^t,\mathcal{M}_i\right)=\sum_{j=1}^{k}\log p\left(\mathbf{z}_j^h|\mathbf{Z}_{j-1}^h,\mathbf{Z}_L^t,\mathcal{M}_i\right) \qquad (37)$$

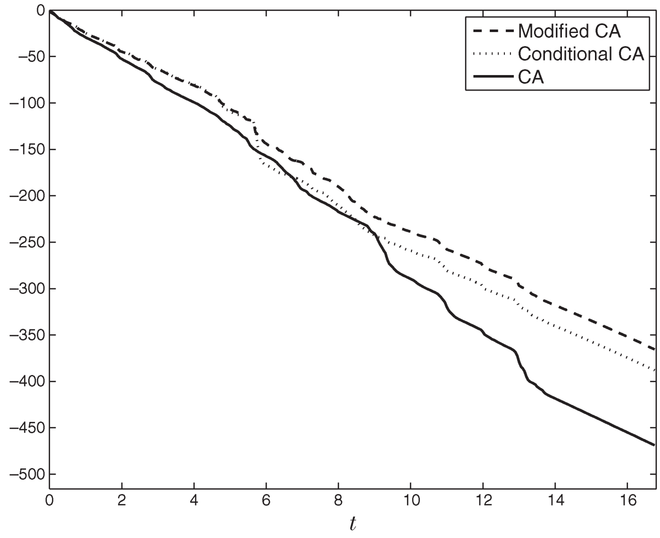

Fig. 9. Log-likelihood functions for the first data sequence.

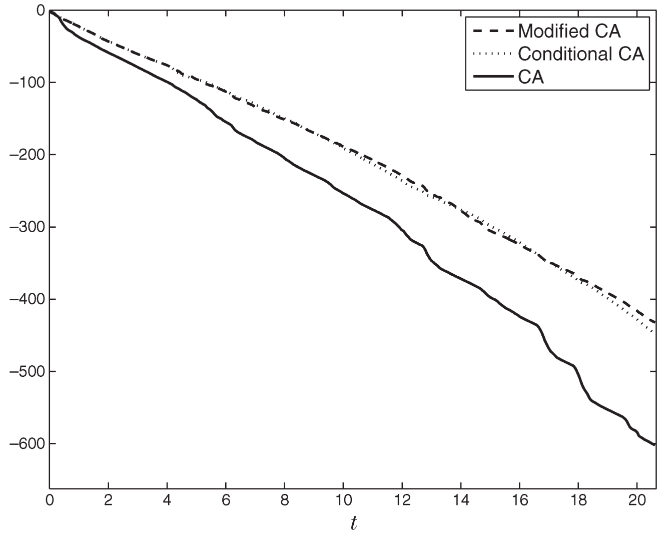

Fig. 10. Log-likelihood functions for the second data sequence.

for $k = 1, \ldots, L$, we also get information about how well the models explain different parts of the sequences (the logarithm is merely included to enable us to illustrate functions of different magnitudes in the same figure). In addition, we also include the function

$$\sum_{j=1}^{k}\log p\left(\mathbf{z}_j^h|\mathbf{Z}_{j-1}^h,\mathbf{Z}_j^t,\mathcal{M}_2\right) \qquad (38)$$

which is made up of the predictive densities discussed in Section V-C1. In the figures, we refer to $\log p(\mathbf{Z}_k^h|\mathbf{Z}_L^t,\mathcal{M}_1)$ as CA, $\log p(\mathbf{Z}_k^h|\mathbf{Z}_L^t,\mathcal{M}_2)$ as conditional CA and $\sum_{j=1}^{k}\log p(\mathbf{z}_j^h|\mathbf{Z}_{j-1}^h,\mathbf{Z}_j^t,\mathcal{M}_2)$ as modified CA.

The three log-likelihood functions are shown in Figs. 9 and 10 for the first and second sequences, respectively. Generally speaking, modified CA and conditional CA are similar and significantly larger (better) than CA. Compared with the proposed model, here represented by modified CA and conditional CA, CA explains the data very poorly when the acceleration changes notably; its likelihood function quickly decreases at those times, whereas both modified CA and conditional CA are much smoother. The only clear exception is the conditional CA, which substantially decreases around $t = 6$ in the first sequence. The reason for this is that the model expects the driver to break harder when there is instead a slight acceleration; the same phenomenon was discussed previously related to Fig. 6. To put the numbers in Figs. 9 and 10 into perspective, Jeffreys [28] states that a log-likelihood difference greater than 2 should be viewed as decisive evidence as to what model is true.

VI. Conclusion and Future Work

We have presented a general framework for modeling vehicle motion. By including the influence of the driver into the model, an improved prediction ability is obtained. Moreover, with a formalized treatment of uncertainties in the underlying model parameters, the description of the prediction uncertainties is improved. Specific attention was here given to adapting the model to when the driver follows the right lane of the road. Accurate data from real driving situations were collected to study the model properties. For the evaluated test scenarios, we have shown that a model developed using our framework describes reference data considerably better than the commonly used CA model.

There are several possibilities to further develop the motion model. First of all, more driver intentions should be added to capture interesting scenarios, such as lane changes, overtakings, and distracted drivers. For these scenarios, the main challenge is to design suitable cost functions that accurately represent the driver behavior and still result in a manageable optimization problem. Further, a more extensive validation is needed using more test scenarios and several different drivers. The validation data should be used to investigate the model's capability to adapt to different driving styles, as well as to be used to fine tune the cost functions presented in this paper. Finally, it would be interesting to explore the usefulness of the prediction capability of the model in a decision making algorithm, for instance, in a collision avoidance application or for run off road detection.

References

[1] "Requirements for preventive safety applications," PReVENT IP Deliverable 4, Apr. 16, 2005. [Online]. Available: www.ip-prevent.org

[2] C. P. Robert, *The Bayesian Choice: A Decision-Theoretic Motivation*. New York: Springer-Verlag, 1994.

[3] X. R. Li and V. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. New York: Wiley, 2001.

[5] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.

[6] R. Mobus, M. Baotic, and M. Morari, "Multi-object adaptive cruise control," in *Proc. 6th Int. Workshop HSCC*, Prague, Czech Republic, Sep. 2003.

[7] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Intell. Vehicles Symp.*, Las Vegas, NV, 2005, pp. 319–324.

[8] P. Gipps, "A behavioural car-following model for computer simulation," *Transp. Res. Part B*, vol. 15, no. 2, pp. 105–111, 1981.

[9] [Online]. Available: http://www.isi-padas.eu/

[10] [Online]. Available: http://www.iterate-project.eu/

[11] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[12] L. Svensson and J. Gunnarsson, "A new motion model for tracking of vehicles," in *Proc. 14th IFAC Symp. Syst. Identification*, Newcastle, Australia, Mar. 2006.

[13] J. Gunnarsson, L. Svensson, F. Bengtsson, and L. Danielsson, "Joint driver intention classification and tracking of vehicles," in *Proc. Nonlinear Stat. Signal Process. Workshop*, Cambridge, U.K., Sep. 2006, pp. 95–98.

[14] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Ser. D. J. Basic Eng.*, vol. 82, pp. 24–45, 1960.

[15] J. Meriam and L. Kraige, *Engineering Mechanics*. New York: Wiley, 1993.

[16] U. Kiencke and L. Nielsen, *Automotive Control Systems for Engine, Driveline and Vehicle*. New York: Springer-Verlag, 2000.

[17] N. Kaempchen, K. Weiss, M. Schaefer, and K. Dietmayer, "IMM object tracking for high dynamic driving maneuvers," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2004, pp. 825–830.

[18] A. Eidehall, "An automotive lane guidance system," Licentiate Thesis no. 1122, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2004.

[19] E. Dickmanns and A. Zapp, "A curvature-based scheme for improving road vehicle guidance by computer vision," in *Proc. SPIE Conf. Mobile Robots*, 1986, pp. 161–168.

[20] G. Leitmann, *The Calculus of Variations and Optimal Control*. New York: Plenum, 1981.

[21] J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA: SIAM, 2001.

[22] W. Murray, P. E. Gill, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM J. Optim.*, vol. 12, no. 4, pp. 979–1006, 2002.

[23] ISSN: 1401-9612 Vgu, vv publikation 2004:80, grundvärden, Jun. 2004.

[24] E. Bertolazzi, F. Biral, and M. Da Lio, "Future advanced driver assistance systems based on optimal control: The influence of "risk functions" on overall system behavior and on prediction of dangerous situations," in *Proc. Intell. Vehicles Symp.*, Parma, Italy, 2004, pp. 386–391.

[25] E. Bertolazzi, F. Biral, M. Da Lio, A. Saroldi, and F. Tango, "Supporting drivers in keeping safe speed and safe distance: The SASPENCE subproject within the European framework programme 6 integrating project PReVENT," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 525–538, Sep. 2010.

[26] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense: 11th Int. Symp. Aerosp./Defense Sens., Simul. Controls, Multi Sensor Fusion, Tracking Resour. Manage. II*, 1997, pp. 182–193.

[27] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.

[28] H. Jeffreys, *Theory of Probability*, 3rd ed. London, U.K.: Oxford Univ. Press, 1961.

Joakim Sörstedt was born in Skara, Sweden, on April 3, 1976. He received the M.S. and Ph.D. degrees in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 2001 and 2007, respectively.

Since 2007, he has been with Volvo Car Corporation, Gothenburg, where he has worked on developing active safety systems. His research interests are in the area of signal processing, specifically subspace-based estimation methods and object tracking.

Lennart Svensson was born in Älvängen, Sweden, in 1976. He received the M.S. and Ph.D. degrees in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 1999 and 2004, respectively.

He is currently an Associate Professor with the Signal Processing Group, Chalmers University of Technology. His research interests include Bayesian inference in general and nonlinear filtering and tracking in particular.

Fredrik Sandblom was born in Mölndal, Sweden, in 1979. He received the M.S. degree in electrical engineering and the Licentiate of Engineering degree from Chalmers University of Technology, Gothenburg, Sweden, in 2004 and 2008, respectively.

Since 2005, he has been with Volvo 3P, Gothenburg, where he works on active safety system research as a part of his Ph.D. studies. His research interests concern object tracking and sensor data fusion, particularly methods for estimating statistical moments.

Lars Hammarstrand received the M.Sc. and Ph.D. degrees in electrical engineering from Chalmers University of Technology, Gothenburg, Sweden, in 2004 and 2010, respectively.

He is currently with the Active Safety and Chassis Department, Volvo Car Corporation, Gothenburg, where he works on sensor data fusion development. His main research interests are in the fields of target tracking and radar sensor modeling, particularly with application to active safety systems.