

# Statistical Threat Assessment for General Road Scenes Using Monte Carlo Sampling

Andreas Eidehall and Lars Petersson

**Abstract**—This paper presents a threat-assessment algorithm for general road scenes. A road scene consists of a number of objects that are known, and the threat level of the scene is based on their current positions and velocities. The future driver inputs of the surrounding objects are unknown and are modeled as random variables. In order to capture realistic driver behavior, a dynamic driver model is implemented as a probabilistic prior, which computes the likelihood of a potential maneuver. A distribution of possible future scenarios can then be approximated using a Monte Carlo sampling. Based on this distribution, different threat measures can be computed, e.g., probability of collision or time to collision. Since the algorithm is based on the Monte Carlo sampling, it is computationally demanding, and several techniques are presented to increase performance without increasing computational load. The algorithm is intended both for online safety applications in a vehicle and for offline data analysis.

**Index Terms**—Decision making, Monte Carlo, road vehicle safety, threat assessment.

## I. INTRODUCTION

WHILE MOST research within automotive collision warning and avoidance is carried out on sensor technology, e.g., radar, laser, and vision systems, threat assessment is attracting much interest. Threat-assessment algorithms are currently moving from deterministic models toward stochastic models which take many possible scenarios into account.

### A. Deterministic Threat Assessment

Deterministic threat assessment refers to methods that predict a single future trajectory for all objects and uses these to compute various threat measures, for instance, time to collision (TTC) [1]–[3], predicted minimum distance, predicted time to minimum distance [4], etc. Note that the predicted trajectories are often based on statistical estimation methods such as the Kalman filter, but only a point estimate is then used in the threat assessment.

Jansson [5] provides an overview of different deterministic threat-assessment methods. An approach when systematically

choosing warning thresholds in different deterministic methods is presented by Yang *et al.* [6].

### B. Stochastic Threat Assessment

A stochastic threat-assessment algorithm is demonstrated by Jansson [5] in a collision mitigation by braking system. The estimated probability density from the Kalman filter is predicted using Kalman filter time updates; thus, it also predicts the distribution and then uses this distribution to find the future point in time where the probability of collision is equal to one. However, this method is only suitable for short-time predictions since the uncertainty quickly increases. Polychronopoulos *et al.* [4] suggest in using several different models for prediction, representing possible future maneuvers. The different possibilities are then condensed into a single prediction without considering them as distinct possibilities. Broadhurst *et al.* [7] take this one step further and consider many (thousands) different possibilities for all objects by trying to approximate the true probability density function (pdf) of the predictions with Monte Carlo sampling. The current state of all objects is assumed to be known, and the control inputs that determine the future trajectories are modeled as stochastic variables which can then be used to compute a pdf for future positions.

This paper is based on the framework presented in [7], with some modifications and uses this to define a new threat-assessment method. One of the contributions is a new way to create a more efficient use of the samples. This is achieved not only by using a dynamic model that is designed with the Monte Carlo sampling in mind but also by an iterative-sampling process which removes and replaces samples that generate collisions at an early stage. We have also added visibility constraints to highlight the fact that drivers have most of their attention focused forward and, thus, are less likely to detect and consider objects in other regions.

The main goal of the threat-assessment algorithm is to detect general threats, i.e., threats that are not necessarily connected to a particular safety function. For example, an extension compared to most conventional safety systems is the ability to detect indirect threats. These are threats from objects that may not be on a direct collision course but are forced onto a collision course by other elements in the traffic situation. An example illustrating this is when an oncoming vehicle has to swerve to avoid an obstacle, and because of that, it becomes a threat to the host vehicle. The vehicles may not be on a direct collision course from the beginning, but the situation still poses a threat because of the obstacle. This example is shown

Manuscript received November 28, 2006; revised March 1, 2007, June 25, 2007, and September 20, 2007. This work was supported in part by the Intelligent Vehicle Safety Systems of the Swedish National Road Authorities and in part by Volvo Car Corporation. The Associate Editor for this paper was W. Scherer.

A. Eidehall is with Linköping University, 581 83 Linköping, Sweden, and also with Volvo Car Corporation, 405 08 Göteborg, Sweden.

L. Petersson is with the Vision Science, Technology, and Applications Group, National ICT Australia, Sydney, N.S.W. 1430, Australia.

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.2007.909241



Fig. 1. Situation where the host vehicle (lower of the two vehicles) is moving from the left to right in a straight line, and an oncoming vehicle (upper of the two vehicles) is moving from the right to the left. The system predicts that it is likely that the oncoming vehicle will have to swerve left in order to avoid the obstacle; thus, it becomes a threat to the host vehicle.

in Fig. 1. Another example is a tailgating situation. Again, the vehicles are not on a collision course from the beginning, but in this case, there is a certain probability that the leading vehicle will brake.

In order to be able to use the threat-assessment algorithm with general road scenes, a demonstration is provided showing how it can be combined with a road-aligned curved coordinate system. This significantly simplifies the model when the road bends. Another contribution is a modified dynamic model which allows accelerations and dynamic forces to be accurately computed directly in the curved coordinate system. By using this, acceleration limitations due to road friction or engine torque can be incorporated without having to switch to Cartesian coordinates.

The result is an algorithm that currently runs offline in Matlab and has been evaluated, both using a small set of simulated scenarios and using a large set of recorded sensor data. The intended use is not only for online safety applications with execution at every time update of the system but also for offline analysis of the recorded sensor data.

The statistical model is formulated in Section II, and the road-aligned coordinate system is presented in Section II-E. Implementation aspects, such as time discretization and collision detection, are discussed in Section IV. Results and conclusions are given in Sections V and VI, respectively.

## II. MODEL

### A. Stochastic Model

The future paths of other objects are determined by their current position and their future control inputs such as steering, braking, etc. In a real application, their current positions can be measured using a sensor or a combination of sensors. However, their future control inputs are unknown, which is why, in this paper, they are modeled using a stochastic variable

$$\mathcal{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$$

which consists of the control input for  $m$  number of objects in the scene.  $\mathbf{u}_i$  contains the control input for a time interval  $I_t = [0, T_{\max}]$  for object  $i$ , i.e.,  $\mathbf{u}_i = (u_1(t), \dots, u_{n_c}(t))_i$ .  $T_{\max}$  is the prediction horizon, i.e., the period of time when the predictions are made, and  $n_c$  is the number of control inputs for each object. This means that, given a control input  $\mathcal{U}$ , the entire system can be simulated, using motion models for all objects, to reach a state  $X(\mathcal{U})$ .  $X(\mathcal{U})$  will contain the position and other

states for all objects for the entire time interval  $I_t$ , given the control input  $\mathcal{U}$ , and can be written as

$$X(\mathcal{U}) = [\mathbf{x}_1(\mathbf{u}_1), \dots, \mathbf{x}_m(\mathbf{u}_m)].$$

In this paper, a prior distribution  $\pi(\mathcal{U})$  is used to model driver preference, i.e., that certain maneuvers are more likely than others. Details are given in Section II-C.

This framework is inspired by the work done by Broadhurst *et al.* [7] in that the control inputs of other objects are considered stochastic and that we also use the Monte Carlo sampling to compute the threat quantities. However, the method suggested in [7] does not consider the fact that almost always, drivers try to avoid colliding with other objects. This is further addressed in Section III.

### B. Dynamic Model

1) *Cars/Bicycles*: A car or a bicycle is geometrically described as a rectangle,<sup>1</sup> and two control inputs ( $u_1, u_2$ ) are used for longitudinal and lateral control, respectively.

In any Monte Carlo application, there is a tradeoff between computational performance and accuracy. More samples means higher accuracy but also higher computational load. In this application, the samples will be spread out over an interval of the control inputs, and we claim that it is important to have the sampling process in mind at an early stage when choosing the dynamic model.

For example, in [7], Broadhurst *et al.* propose using steering-wheel angles as control input. However, for high velocities, the limiting factor for lateral movement is not the steering angle but rather a tyre-to-road friction. In order to deal with this, Broadhurst *et al.* [7] propose to simply remove samples with higher lateral acceleration than what is physically allowed. We believe that better results could be obtained by distributing the samples according to this maximum level from the beginning, and thus, we get a higher concentration of samples within the allowed control-input set.

A simple friction model is used, where the maximal friction force is proportional to the normal force and thus limits the acceleration. To simplify the model, the accelerations in the lateral and longitudinal directions are treated separately. The model that will be used is

$$\dot{x} = v \cos \theta \quad (1a)$$

$$\dot{y} = v \sin \theta \quad (1b)$$

$$\dot{v} = \begin{cases} u_1 a_f, & \text{if } v \leq v_{\text{long}} \\ u_1 \frac{k/v + a_f}{2} + \frac{k/v - a_f}{2}, & \text{if } v > v_{\text{long}} \end{cases} \quad (1c)$$

$$\dot{\theta} = \begin{cases} v \sin(\varphi_{\max} u_2) / L, & \text{if } v \leq v_{\text{lat}} \\ a_f u_2 / v, & \text{if } v > v_{\text{lat}} \end{cases} \quad (1d)$$

where  $(x, y)$  is the position,  $v$  is the velocity, and  $\theta$  is the heading angle of the modeled vehicle. Furthermore, the maximum

<sup>1</sup>Figures for a typical car: The size is  $1.8 \times 4.8$  m,  $L = 2.4$  m,  $\varphi_{\max} = 0.5$  rad,  $a_f = 9.1 \text{ m} \cdot \text{s}^{-2}$ , and  $k = 66.6 \text{ m}^2 \cdot \text{s}^{-3}$ . This gives  $v_{\text{lat}} = 6.76 \text{ m/s}$  and  $v_{\text{long}} = 7.32 \text{ m/s}$ . For a bicycle, the size is  $0.6 \times 2$  m,  $L = 1.6$  m,  $\varphi_{\max} = 0.5$  rad,  $a_f = 4 \text{ m} \cdot \text{s}^{-2}$ , and  $k = 0.75 \text{ m}^2 \cdot \text{s}^{-3}$ . This gives  $v_{\text{lat}} = 3.7 \text{ m/s}$  and  $v_{\text{long}} = 0.2 \text{ m/s}$ .

acceleration due to road friction is denoted as  $a_f$ ,  $\varphi_{\max}$  is the maximum steering angle,  $L$  is the wheelbase, and  $k$  is a parameter describing engine power.  $v_{\text{long}}$  and  $v_{\text{lat}}$  are the breakpoint velocities at which the limit of the longitudinal acceleration switches from tire-to-road friction to engine power and from maximum steering angle to tyre-to-road friction. By using this model, the control-input sets  $u_1, u_2 \in [-1, 1]$  can be used for all velocities.

2) *Pedestrians*: The motion model for pedestrians is a simple constant-velocity model

$$\begin{aligned}\dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= a_f u_1 \\ \dot{v}_y &= a_f u_2.\end{aligned}$$

The input set  $u_1, u_2 \in [-1, 1]$  can also be used in this case.

3) *Obstacles*: To emulate other obstacles and road boundaries, general convex polygons will be used. Each polygon also has a motion vector, which allows moving objects to be generated.

### C. Prior Distribution

The prior distribution  $\pi(\mathcal{U})$  is used to model the fact that drivers have a goal with their driving. For instance, they want to get from points  $A$  to  $B$  and maintain a desired velocity. Typically, drivers also strive for a comfortable ride by minimizing acceleration forces in all directions. In [7], Broadhurst *et al.* suggest a driver-preference distribution that models four different aspects:

- 1) distance to intended path;
- 2) deviation from desired velocity;
- 3) longitudinal acceleration;
- 4) steering angle.

We use the same distribution except that the steering angle is replaced with lateral acceleration, which seems to be a more relevant measure. The distribution is defined as

$$\pi(\mathcal{U}) = a e^{-f(\mathcal{U}, X(\mathcal{U}))} \quad (2)$$

where  $a$  is a normalizing constant, and

$$f(\mathcal{U}, X(\mathcal{U})) = \sum_{i=1}^m \omega_i g(\mathbf{u}_i, \mathbf{x}_i(\mathbf{u}_i)). \quad (3)$$

The sum runs over all objects in the scene, i.e.,  $f$  is the total cost of the maneuvers of all objects. The weights  $\omega_i$  can be used to prioritize the prior driver preference between different objects, something that is utilized when implementing the visibility constraints in Section II-D. We then define

$$\begin{aligned}g(\mathbf{u}_i, \mathbf{x}_i(\mathbf{u}_i)) &= \int_{I_t} [(l_x x(t) + l_y y(t) - l_z)^2 \lambda_1 + (v(t) - v_0)^2 \lambda_2 \\ &\quad + a_{\text{long}}(t)^2 \lambda_3 + a_{\text{lat}}(t)^2 \lambda_4] dt \quad (4)\end{aligned}$$

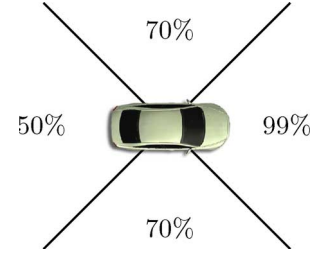


Fig. 2. Probability that the host vehicle will observe/regard a vehicle within different regions.

which represents the cost of a maneuver for a single object over the entire time interval  $I_t$ , where  $l_x x(t) + l_y y(t) - l_z$  measures the distance to the line  $l = (l_x, l_y, l_z)$ , and  $v_0$  is the initial (desired) velocity. The line  $l$  represents the desired path and can be any line, but it usually coincides with the tangent of the object at time  $t = 0$ . Note that for the pedestrian object class,  $v(t)$  has to be computed from  $v_x(t)$  and  $v_y(t)$ . The signals  $a_{\text{lat}}(t)$  and  $a_{\text{long}}(t)$  are computed from the control inputs. The weights  $\lambda_i$  are chosen to balance the costs of these different behavioral aspects in (4) and have so far been calibrated by visually inspecting the distributions.<sup>2</sup> These values are significantly higher than those presented in [7] which gives a narrower distribution. In order to avoid numerical problems at the “tails” of the distribution, a logarithmic representation of  $\pi(\mathcal{U})$  has been implemented. A more systematic approach to calibrating these parameters, for example, by adapting the distribution to authentic recorded data, would certainly be interesting, and this may lie ahead in our future work.

### D. Visibility Constraints

In this section, a method is formulated for incorporating the fact that drivers have most of their attention directed forward and are more likely to detect other objects in this region. Furthermore, drivers are much more inclined to adapt their driving behavior with respect to other vehicles in front of them rather than behind. For example, in Fig. 3, if vehicle 1 approaches vehicle 2 from behind with a high relative velocity, then vehicle 1 is much more likely to change course in order to avoid colliding with vehicle 2, compared to vehicle 2 that changes course in order to let vehicle 1 pass, even though this might have been a more comfortable maneuver for both vehicles overall.

The first step is to define regions around each object and then assign an attention level to each region. The regions and corresponding attention levels we have chosen are shown in Fig. 2. These values are just an example and are not scientifically justified; the main contribution here is the way these constraints can be implemented. A matrix  $\mathbf{V}$  is constructed which contains the visibility levels within all object pairs, where element  $\mathbf{V}_{ij}$  indicates how well object  $i$  can be seen or how

<sup>2</sup> $\lambda_1 = 60/T_{\max}$ ,  $\lambda_2 = 0.5/T_{\max}/(1 + |v_0|)$ ,  $\lambda_3 = 1/T_{\max}/a_f^2$ ,  $\lambda_4 = 75/T_{\max}/\varphi_{\max}$ , where  $T_{\max}$  is the length of the time interval  $I_t$ .

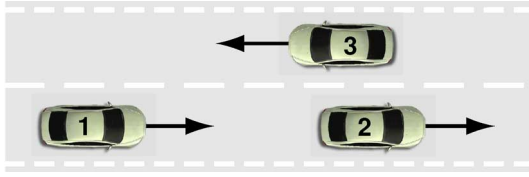


Fig. 3. Scenario used as an example to illustrate the method for incorporating visibility constraints.

much it is registered by object  $j$ . The example in Fig. 3 will generate the following  $\mathbf{V}$  matrix:

	$j = 1$	$j = 2$	$j = 3$
$i = 1$	—	50%	99%
$i = 2$	99%	—	70%
$i = 3$	99%	70%	—

The weights  $\omega_i$  in (3) are then chosen as

$$\omega_i = \sum_{j \neq i} \hat{\mathbf{V}}_{ij} \quad (5)$$

where  $\hat{\mathbf{V}}$  is a normalized version of  $\mathbf{V}$ , i.e., the sum of all elements is equal to one. For example, this means that in a situation where one vehicle follows another, the leading vehicle will have a high weight in order to represent the fact that this vehicle is more likely to follow its own preferences rather than adapting its behavior to the vehicle approaching from behind.

#### E. Road-Aligned Coordinates and Tracking

Many automotive tracking systems recently presented have proposed using a curved road-aligned coordinate system for tracking [4], [8], [9]. Such a curved coordinate system would also be very suitable to use with the threat-assessment algorithm proposed in this paper. For instance, if the road boundaries of a curved road are to be modeled in a Cartesian coordinate system, they would either have to be approximated with a number of polygon shapes, or a more general class of obstacles with curved edges would have to be introduced. If a road-aligned coordinate system is used, the road boundary could be modeled as a straight line instead. Furthermore, the driver-preference function, which is represented by the prior in Section II-C, could also be greatly simplified. The straight line representing the desired path would automatically follow the curved road in such a coordinate system.

However, before the threat-assessment algorithm can be implemented, the effects of the curved coordinate system on the physical limitations of the vehicle and the dynamic model need to be investigated. For example, when driving along a curve, a certain amount of the lateral force available is already used just to follow the road, thus reducing the amount of remaining friction force in that direction. This is not automatically incorporated, and how to handle this is analyzed in Section II-E2.

1) *Tracking*: The position of objects in the road coordinates is often already available since tracking is typically done in that frame of reference. It has been shown that tracking in the curved road-aligned coordinates has many advantages [8], [9]. The main feature is that the estimate of the road shape

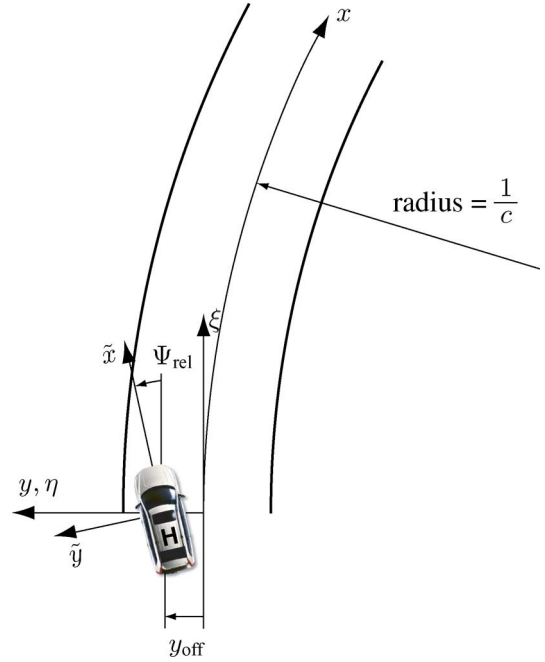


Fig. 4. Definition of the coordinate systems  $(x, y)$ ,  $(\tilde{x}, \tilde{y})$ , and  $(\xi, \eta)$  and the parameters  $c$ ,  $y_{\text{off}}$ , and  $\Psi_{\text{rel}}$ .

can be improved by studying the motion of leading vehicles and by using a model of their motion in relation to the road. There are also other advantages when it comes to modeling and prediction. In the system presented, a road model with the shape of a circle segment is used, as is suggested in [8].

The positions of objects are expressed in  $(x, y)$  coordinates, where  $x$  is the longitudinal position along the road, and  $y$  is the lateral position measured from the center of the host-vehicle lane. In order to carry out tracking, these positions need to be related to the coordinate systems of the sensors attached to the host vehicle, which is why a host-vehicle coordinate system  $(\tilde{x}, \tilde{y})$  is introduced (see Fig. 4).

A relationship between  $(x, y)$  and  $(\tilde{x}, \tilde{y})$  can be expressed as

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \mathbf{R}(-\Psi_{\text{rel}}) \begin{pmatrix} (1 + cy) \sin(cx) \\ (1 + cy) \cos(cx) - 1 - cy_{\text{off}} \end{pmatrix} \frac{1}{c}$$

where  $\mathbf{R}$  is a rotational matrix, i.e.,

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

and the other variables are defined in Fig. 4. This relationship is then used to construct a measurement equation in an extended Kalman filter. For the purpose of tracking, a simpler dynamic model than the one described in Section II-B is used. One of the advantages of the road-aligned coordinate system is that modeling motion in relation to the road becomes greatly simplified. For example, the equation  $\dot{y} = 0$  can be used to express the fact that vehicles will keep to their lanes even if the road bends. For longitudinal dynamics, the equation  $\ddot{x} = 0$  is used, i.e., constant velocity.

2) *Vehicle Dynamics in Curved Coordinates*: In this section, the effects from the curved coordinate system on the dynamics of the vehicles are analyzed, which is one of the main

contributions of this paper. A problem with a non-Cartesian frame of reference is that Newton's laws of motion cannot be directly applied. Instead, they need to be applied in Cartesian coordinates and then transformed to the curved space. Thus, we introduce Cartesian coordinates  $(\xi, \eta)$  which are attached at the center of the host-vehicle lane and coincide with  $(x, y)$  at the origin (see Fig. 4). Then, the accelerations  $\ddot{x}$  and  $\ddot{y}$  as functions of  $\xi$  and  $\eta$  are computed. The physical limits in the  $(\xi, \eta)$  space can then be translated to the  $(x, y)$  space. Starting with  $\xi$ , the relationship for velocity is found as

$$\dot{\xi} = \frac{\partial \xi}{\partial x} \dot{x} + \frac{\partial \xi}{\partial y} \dot{y}.$$

Acceleration can then be obtained as

$$\begin{aligned} \ddot{\xi} &= \frac{d}{dt} \left( \frac{\partial \xi}{\partial x} \dot{x} + \frac{\partial \xi}{\partial y} \dot{y} \right) \\ &= \left( \frac{\partial^2 \xi}{\partial x^2} \dot{x} + \frac{\partial^2 \xi}{\partial x \partial y} \dot{y} \right) \dot{x} + \frac{\partial \xi}{\partial x} \ddot{x} \\ &\quad + \left( \frac{\partial^2 \xi}{\partial x \partial y} \dot{x} + \frac{\partial^2 \xi}{\partial y^2} \dot{y} \right) \dot{y} + \frac{\partial \xi}{\partial y} \ddot{y} \\ &= \frac{\partial^2 \xi}{\partial x^2} \dot{x}^2 + \frac{\partial^2 \xi}{\partial y^2} \dot{y}^2 + 2 \frac{\partial^2 \xi}{\partial x \partial y} \dot{x} \dot{y} + \frac{\partial \xi}{\partial x} \ddot{x} + \frac{\partial \xi}{\partial y} \ddot{y} \\ &= \mathbf{v}^T \mathbf{H}_\xi \mathbf{v} + \mathbf{J}_\xi \mathbf{a} \end{aligned} \quad (6)$$

where  $\mathbf{v}$  and  $\mathbf{a}$  represent the velocity and acceleration in the  $(x, y)$  coordinates, i.e.,

$$\mathbf{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix}.$$

$\mathbf{J}_\xi$  is the Jacobian

$$\mathbf{J}_\xi = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \end{pmatrix}$$

and  $\mathbf{H}_\xi$  is the Hessian matrix

$$\mathbf{H}_\xi = \begin{pmatrix} \frac{\partial^2 \xi}{\partial x^2} & \frac{\partial^2 \xi}{\partial x \partial y} \\ \frac{\partial^2 \xi}{\partial x \partial y} & \frac{\partial^2 \xi}{\partial y^2} \end{pmatrix}.$$

The  $\eta$  coordinate can be treated analogously

$$\ddot{\eta} = \mathbf{v}^T \mathbf{H}_\eta \mathbf{v} + \mathbf{J}_\eta \mathbf{a}. \quad (7)$$

Equations (6) and (7) show a general relationship between accelerations in two coordinate systems. A circular road segment has the following coordinate transformation:

$$\begin{pmatrix} \xi \\ \eta \end{pmatrix} = \begin{pmatrix} (1 + cy) \sin(cx) \\ (1 + cy) \cos(cx) - 1 \end{pmatrix} \frac{1}{c}.$$

Inserting this into (6) and (7), preferably using a symbolic software package, and collecting the trigonometric terms give

$$\begin{aligned} \ddot{\xi} &= \sin(cx) [\ddot{y} - c\dot{x}^2(1 + cy)] \\ &\quad + \cos(cx) [(1 + cy)\ddot{x} + 2c\dot{x}\dot{y}] \end{aligned} \quad (8)$$

$$\begin{aligned} \ddot{\eta} &= \cos(cx) [\ddot{y} - c\dot{x}^2(1 + cy)] \\ &\quad - \sin(cx) [(1 + cy)\ddot{x} + 2c\dot{x}\dot{y}]. \end{aligned} \quad (9)$$

The movement of objects along the  $y$ -axis is small (about  $\pm 5$  m) compared to the typical road radius (typically above 100 m), which implies that  $|cy| \ll 1$ , and thus,  $1 + cy \approx 1$ . Furthermore, (8) and (9) can jointly be expressed as a rotation, again using the rotational matrix  $\mathbf{R}$

$$\begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} = \mathbf{R}(-cx) \begin{pmatrix} \ddot{x} + 2c\dot{x}\dot{y} \\ \ddot{y} - c\dot{x}^2 \end{pmatrix} \Leftrightarrow \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \mathbf{R}(cx) \begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} + \begin{pmatrix} -2c\dot{x}\dot{y} \\ c\dot{x}^2 \end{pmatrix}. \quad (10)$$

However,  $\ddot{x}$  and  $\ddot{y}$  do not necessarily coincide with the longitudinal and lateral directions for the individual vehicles. That is why a rotation of the accelerations  $\ddot{x}$  and  $\ddot{y}$  with  $\theta$  radians is needed, where  $\theta$  is the heading angle of the vehicle relative to the road, as explained in Section II-B. The Cartesian accelerations  $\ddot{\xi}$  and  $\ddot{\eta}$  also need to be rotated for the same reason since the acceleration limitations are expressed in the vehicle reference frame. These need to be rotated: first with  $cx$  radians to be aligned with the road,  $cx$  being the angle between the  $x$ - and  $\xi$ -axes, and then with an additional  $\theta$  rad to become aligned with the vehicle. The following variables are introduced:

$$\begin{pmatrix} a_{\text{long}} \\ a_{\text{lat}} \end{pmatrix} = \mathbf{R}(\theta) \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} a'_{\text{long}} \\ a'_{\text{lat}} \end{pmatrix} = \mathbf{R}(\theta + cx) \begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} \quad (12)$$

where “long” and “lat” refer to the longitudinal and lateral directions of the vehicle modeled. To explain the notation,  $a_{\text{long}}$  and  $a_{\text{lat}}$  are accelerations in the curved coordinate system, and  $a'_{\text{long}}$  and  $a'_{\text{lat}}$  are the corresponding accelerations in a fixed Cartesian coordinate system, where Newton's laws of motion can be applied. Multiplying (10) with  $\mathbf{R}(\theta)$  immediately gives

$$\begin{pmatrix} a_{\text{long}} \\ a_{\text{lat}} \end{pmatrix} = \begin{pmatrix} a'_{\text{long}} \\ a'_{\text{lat}} \end{pmatrix} + \mathbf{R}(\theta) \begin{pmatrix} -2c\dot{x}\dot{y} \\ c\dot{x}^2 \end{pmatrix} \quad (13)$$

and by introducing the variables

$$\begin{pmatrix} a_{\text{long,off}} \\ a_{\text{lat,off}} \end{pmatrix} = \mathbf{R}(\theta) \begin{pmatrix} -2c\dot{x}\dot{y} \\ c\dot{x}^2 \end{pmatrix}. \quad (14)$$

Equation (13) can be rewritten as

$$\begin{cases} a_{\text{long}} = a'_{\text{long}} + a_{\text{long,off}} \\ a_{\text{lat}} = a'_{\text{lat}} + a_{\text{lat,off}} \end{cases} \quad (15)$$

The subscript “off” suggests that these act as an offset on the accelerations to compensate for the fact that the coordinate system is non-Cartesian. By using  $\dot{x} = v \cos \theta$  and  $\dot{y} = v \sin \theta$ , the acceleration offsets can be computed by evaluating (14)

$$a_{\text{long,off}} = -\cos^2 \theta \sin \theta c v^2 \quad (16)$$

$$a_{\text{lat,off}} = (\cos^3 \theta - 2 \cos \theta \sin^2 \theta) c v^2. \quad (17)$$

To illustrate these offsets, consider a special case where a vehicle drives along a curve and stays at a constant lateral position in relation to the lane. In this case,  $\dot{y} = \theta = 0$ , and thus,



$a_{\text{long,off}} = 0$  and  $a_{\text{lat,off}} = cv^2$ .  $cv^2$  acts as an acceleration offset in the lateral direction, exactly describing the phenomenon in the example mentioned earlier, i.e., when negotiating a curve, a certain amount of the available friction force in one direction is already used to stay in the lane, thus reducing the remaining room for maneuverability in the same direction. According to typical guidelines for road construction [10], a 90-km/h road has a minimum radius of 550 m, which gives  $a_{\text{off}} = 1.25 \text{ m/s}^2$ . A road of 50 km/h has a minimum radius of 140 m, which gives  $a_{\text{off}} = 1.38 \text{ m/s}^2$ . Thus, under good road conditions, the offset can be up to about 15% of the lateral acceleration available, which indicates that this effect should not be neglected.

The minimum turning radius is also affected by the fact that the coordinates are curved. A similar offset for the turning rate is thus introduced

$$\dot{\theta} = \dot{\theta}' + \dot{\theta}_{\text{off}} \quad (18)$$

where  $\dot{\theta}$  is the turning rate in the curved coordinates, and  $\dot{\theta}'$  is the turn rate in the fixed Cartesian coordinates. First, define

$$\mathbf{v}' = \begin{pmatrix} \dot{\xi} \\ \dot{\eta} \end{pmatrix} \quad \text{and} \quad v' = |\mathbf{v}'|. \quad (19)$$

The following well-known relationships between turn rate and acceleration under a circular motion in the two reference frames are used:

$$a_{\text{lat}} = v\dot{\theta} \quad \text{and} \quad a'_{\text{lat}} = v'\dot{\theta}'. \quad (20)$$

Note that  $a'_{\text{lat}}$  can be computed from  $a_{\text{lat}}$  using (15). In order to find  $v'$ , the Jacobian is evaluated as

$$\mathbf{J}_{\xi,\eta} = \begin{pmatrix} \mathbf{J}_{\xi} \\ \mathbf{J}_{\eta} \end{pmatrix} = \begin{pmatrix} (1+cy)\cos(cx) & \sin(cx) \\ -(1+cy)\sin(cx) & \cos(cx) \end{pmatrix}. \quad (21)$$

Again, by using  $1+cy \approx 1$ , it is noted that  $\mathbf{J}_{\xi,\eta} \approx \mathbf{R}(-cx)$ . Thus, the following relationship holds:

$$\mathbf{v}' = \mathbf{J}_{\xi,\eta}\mathbf{v} \approx \mathbf{R}(-cx)\mathbf{v} \Rightarrow v = |\mathbf{v}| \approx |\mathbf{v}'| = v'. \quad (22)$$

By using this and inserting (20) into (18) give

$$a_{\text{lat}}/v = a'_{\text{lat}}/v + \dot{\theta}_{\text{off}} \quad (23)$$

and thus, from the definition of  $a_{\text{lat,off}}$

$$\dot{\theta}_{\text{off}} = (a_{\text{lat}} - a'_{\text{lat}})/v = a_{\text{lat,off}}/v. \quad (24)$$

The model can then be modified by adding  $a_{\text{long,off}}$  to the longitudinal dynamics and  $a_{\text{lat,off}}$  and  $\dot{\theta}_{\text{off}}$  to the lateral dynamics, replacing (1c) and (1d) with

$$\dot{v} = \begin{cases} u_1 a_f + a_{\text{long,off}}, & \text{if } v \leq v_{\text{long}} \\ u_1 \frac{k/v+a_f}{2} + \frac{k/v-a_f}{2} + a_{\text{long,off}}, & \text{if } v > v_{\text{long}} \end{cases} \quad (25)$$

$$\dot{\theta} = \begin{cases} v \sin(\varphi_{\text{max}} u_2)/L + \dot{\theta}_{\text{off}}, & \text{if } v \leq v_{\text{lat}} \\ (a_f u_2 + a_{\text{lat,off}})/v, & \text{if } v > v_{\text{lat}}. \end{cases} \quad (26)$$

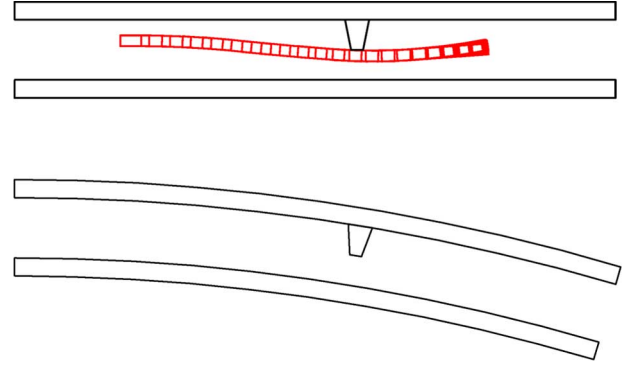


Fig. 5. Vehicle approaching an obstacle at a high velocity. In the top scenario, the vehicle can barely pass the obstacle. The lower scenario has no conflict-free solution since the lateral friction force available is not sufficient to avoid the obstacle.

To demonstrate how these modifications affect the model, consider the example in Fig. 5. A vehicle approaches an obstacle at high velocity and has to swerve right in order to avoid it. When the road is straight, this is possible, but if the same scenario takes place on a road that bends to the right, the lateral friction force available is not sufficient.

### III. THREAT ASSESSMENT

In this section, “host vehicle” refers to the vehicle equipped with the potential safety system, i.e., the vehicle that is warned of potential danger. The host vehicle is modeled as a deterministic object, not in the stochastic framework. The reason for this is that even though the future actions of the host vehicle are not known, the threat of the current trajectory can be computed. The driver of the host vehicle can then, for example, be informed if he or she needs to act or not in order to avoid a collision.

As was discussed earlier, drivers most often try to avoid collisions. Trying to incorporate this in the model, we will create a mix of two distributions: one where other vehicles try to avoid the host vehicle and one where they do not see or consider the host vehicle. These distributions are then weighted based on the visibility relation between the host vehicle and the objects. We start by computing the posterior distribution of  $\mathcal{U}$ , given that there are no collisions during  $I_t$ . Mathematically, a collision-free event is expressed as the complement  $\mathcal{C}^c$ , where  $\mathcal{C}$  is the event of a collision. The posterior distribution of  $\mathcal{U}$  can then be computed using Bayes theorem

$$P(\mathcal{U}|\mathcal{C}^c) = \frac{P(\mathcal{C}^c|\mathcal{U})\pi(\mathcal{U})}{\int_{\mathcal{X}_M} P(\mathcal{C}^c|\mathcal{U})\pi(\mathcal{U})d\mathcal{U}} \quad (27)$$

where  $\mathcal{X}_M$  is the set of physically allowed steering inputs,  $P(\mathcal{C}|\mathcal{U}) \in \{0, 1\}$  is the probability of collision given a steering input, and  $\pi(\mathcal{U})$  is the prior distribution of  $\mathcal{U}$ , which was explained in Section II-C.

In order to compute the final distribution of  $\mathcal{U}$ , a certain risk that other objects might disregard or not see the host vehicle will be included. The following events are defined:  $\mathcal{C}_A$  = collision between any objects, including the host

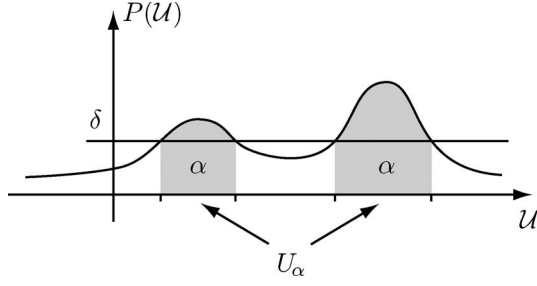


Fig. 6. How the set  $U_\alpha$  is defined.

vehicle, and  $\mathcal{C}_B$  = collision between any objects, excluding the host vehicle. A new merged distribution for  $\mathcal{U}$  is then defined as

$$P(\mathcal{U}) = w_A P(\mathcal{U}|\mathcal{C}_A^c) + w_B P(\mathcal{U}|\mathcal{C}_B^c). \quad (28)$$

The weights are chosen based on the visibility calculations in Section II-D in order to capture the fact that the probability of the host vehicle being disregarded or not seen by another vehicle depends on their relative positions and orientations.

We propose the following weights, where  $k$  is the object index of the host vehicle

$$w_A = \min_j \{\hat{\mathbf{V}}_{kj} : j \neq k\} \quad (29)$$

and then  $w_B$  is chosen such that  $w_A + w_B = 1$ . This makes sense since  $w_A$  is then the minimum visibility of the host vehicle by other objects, i.e., it represents the worst case of how little the host vehicle is regarded or how little it is seen by others.

Whether a situation is judged to be dangerous or not depends on how much of the probability mass of  $\mathcal{U}$  is conflict-free. This will be determined by forming a set  $U_\alpha \subset \mathcal{X}_M$  which is defined to be the most likely set of control inputs with probability mass  $\alpha$ . Mathematically, it is obtained by first defining  $U(\delta) = \{\mathcal{U} \in \mathcal{X}_M : P(\mathcal{U}) > \delta\}$  and then  $\delta_\alpha = \sup\{\delta \in \mathbb{R}^+ : P(U(\delta)) > \alpha\}$ . This means that  $U_\alpha = U(\delta_\alpha) = \{\mathcal{U} \in \mathcal{X}_M : P(\mathcal{U}) > \delta_\alpha\}$  has  $P(U_\alpha) \geq \alpha$ , but depending on the behavior of the distribution, we often actually have  $P(U_\alpha) = \alpha$ . Furthermore, we are guaranteed that  $U_\alpha$  does not have a lesser probability than  $\alpha$ . The set  $U_\alpha$  is shown in Fig. 6.

A final threat level of the host vehicle can then be computed as

$$P(\mathcal{C}_B|U_\alpha) \in \{0, 1\} \quad (30)$$

by choosing  $\alpha$  at an appropriate level, depending on how much probability mass to include. This can be interpreted as the amount of probability mass the host vehicle should avoid, i.e., by using  $\alpha = 99\%$ , the probability of avoiding a collision is greater than 99%.

If a threat is detected, i.e.,  $P(\mathcal{C}_B|U_\alpha) = 1$ , TTC can be computed and used as a warning level for taking countermeasures or alerting the driver.

## IV. IMPLEMENTATION

### A. Collision Detection and Time Discretization

This section deals with the computation of  $P(\mathcal{C}|\mathcal{U})$  in (27), i.e., given a combination of control inputs  $\mathcal{U}$ , compute whether it will result in a collision. The first step is to insert the time variable  $t$  using the marginalization formula

$$P(\mathcal{C}|\mathcal{U}) = \int_{I_t} P(\mathcal{C}|\mathcal{U}, t) P(t) dt \quad (31)$$

where  $P(t)$  is constant, since all times in  $I_t$  will appear with equal probability. However, computing  $P(\mathcal{C}|\mathcal{U}, t)$  for a continuous time variable  $t$  is difficult; thus, discretization of the time variable is needed. First, define a sampling time  $T_s$  and a time interval  $I_s(n) = [nT_s, (n+1)T_s]$ ; then, (31) can be rewritten as

$$P(\mathcal{C}|\mathcal{U}) = \sum_n \int_{I_s(n)} P(\mathcal{C}|\mathcal{U}, t) P(t) dt \quad (32)$$

where the sampling time  $T_s$  is chosen to be small enough so that the integral can be approximated as

$$\int_{I_s(n)} P(\mathcal{C}|\mathcal{U}, t) dt \approx P(\mathcal{C}|\mathcal{U}, t = nT_s). \quad (33)$$

The appropriate size of  $T_s$  depends on the expected sizes and velocities of objects; if it is too large, there is a chance that objects can pass through each other and that the detection of a collision is missed. In order to compute  $P(\mathcal{C}|\mathcal{U}, t = nT_s)$ , the system is simulated, using the dynamic model in Section II-B, to reach time  $nT_s$ . Then, the polygon shape of all objects is computed, and a polygon-intersection algorithm is applied. If a nonempty intersection between any two polygons is detected, we have  $P(\mathcal{C}|\mathcal{U}, t = nT_s) = 1$ .

Similarly, a time discretization for the control input is defined, with a sampling time  $T_c$  and a corresponding time interval  $I_c(k) = [kT_c, (k+1)T_c]$ . The reason for discretizing the control input differently is that the frequency content of a typical driver input is much lower in comparison to the sampling frequency required to detect collisions.<sup>3</sup> The discrete time trajectories of the dynamic model from Section II-B are computed using the Runge–Kutta method.

### B. Monte Carlo Sampling

In order to evaluate the integral (27), the Monte Carlo sampling is used, where  $N$  samples of the control-input space  $\mathcal{X}_M$  are generated. The prior probability discussed in Section II-C is then computed, and the collision-detection algorithm from Section IV-A is applied for each sample. The result is a set of samples that represent the distribution of  $\mathcal{U}$ , which can then be used to compute the threat quantities.

We found that a uniform sampling worked well; the main reason for this is a technique referred to as iterative sampling,

<sup>3</sup>We use  $T_s = 0.1$  s and  $T_c = 0.5$  s.

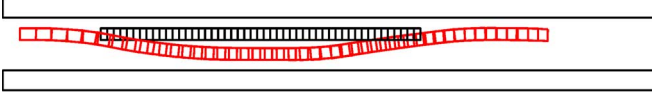


Fig. 7. Scenario where two vehicles are moving from the left to the right, with one approaching the other from behind with a higher velocity. The large long rectangles are used to model road boundaries. This overtaking maneuver was found as the most likely path, i.e.,  $\arg \max_{\mathcal{U} \in \mathcal{X}_M} P(\mathcal{U}|\mathcal{C}^c)$ , among 1000 samples using iterative sampling with uniform sampling only.

discussed in Section IV-C, which makes the final sample density very high. It thus becomes unnecessary to emphasize special maneuvers or sets of the control-input space, which has to be done in the method proposed in [7]. Fig. 7 shows an example of an overtaking maneuver that was found as the most likely path, i.e.,  $\arg \max_{\mathcal{U} \in \mathcal{X}_M} P(\mathcal{U}|\mathcal{C}^c)$ , using the uniform sampling only. Typically, without the iterative-sampling process, 1000 samples would not be enough to find a single collision-free sample while the iterative process generates 600 to 700 samples.

Moreover, a Monte Carlo-based threat-assessment algorithm, such as the one proposed, would be an ideal complement to a Monte Carlo-based filter, i.e., the particle filter [11], [12]. In a particle filter, the distribution of the current positions of objects is already represented by samples. These samples could then be fed straight into the threat-assessment algorithm, creating a seamless statistical connection between filtering and threat assessment.

### C. Iterative-Sampling Process

A straightforward approach in obtaining a set of conflict-free samples would be as follows: first, to generate a random set of control inputs; second, to compute the trajectories of all objects using the dynamic model; and, finally, to apply the collision-detection scheme proposed in Section IV-A. Here, we propose an iterative process which will increase the density of the final set of samples by iteratively removing and replacing samples that generate conflicts at an early stage in the simulation.

Let  $\mathcal{X}_M$  be the set of physically allowed control inputs, and let  $\mathcal{X}_F^k$  be the set of inputs free from conflict at time step  $k$ , which refers to the time interval  $I_c(k)$ , as defined in Section IV-A. Formally

$$\mathcal{X}_F^k = \{\mathcal{U} \in \mathcal{X}_M : P(\mathcal{C}|\mathcal{U}, t \in I_c(k)) = 0\}. \quad (34)$$

The algorithm starts by generating a set  $U_1 \in \mathcal{X}_M$  of  $N$  control inputs for time step  $k = 1$ . Then, for  $k = 1, 2, \dots$ , repeat the following steps.

- 1) Simulate the system during time interval  $k$  using the control inputs  $U_k$ .
- 2) Compute

$$\tilde{U}^k = U^k \cap \mathcal{X}_F^k \Rightarrow |\tilde{U}^k| = N_k \leq N \quad (35)$$

that is, remove the control inputs that will generate a conflict in the time interval  $k$ .

- 3) Now, form  $U^{k+1}$  by resampling  $\tilde{U}^k$  such that  $|U^{k+1}| = N$ . We propose using a mixed distribution where new samples are drawn both from a uniform distribution and

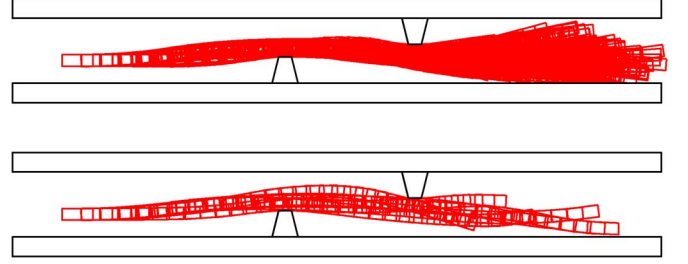


Fig. 8. Set of conflict-free particles based on 500 initial samples obtained (top) with and (bottom) without iterative sampling. The iterative-sampling algorithm generates much denser sample sets.

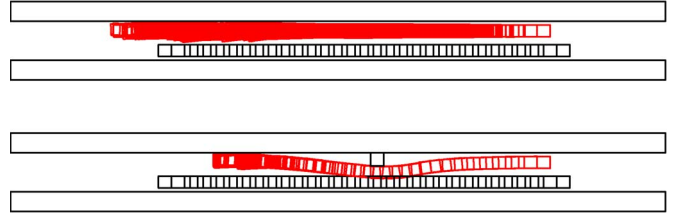


Fig. 9. Revisiting the scenario in Fig. 1, in these two simulations, the host vehicle (lower of the two vehicles) moves from left to right in a straight line, and an oncoming vehicle (upper of the two vehicles) moves from the right to the left. In the top scenario, no danger is detected, whereas in the bottom scenario, there is an obstacle that the oncoming vehicle has to avoid; thus, it becomes a threat to the host vehicle. In the bottom scenario, a threat is detected. TTC is 1.4 s.

from the prior distribution. The resampling is done using a replacement, which means that there will be multiple copies of some elements in  $U^{k+1}$ . Then, new random control inputs in  $\mathcal{X}_M$  for time step  $k + 1$  are generated and adjoined to the control inputs from  $U^{k+1}$  so that even if some control inputs are repeated up to time step  $k$ , they are all different at time step  $k + 1$ .

The algorithm is terminated after Step 2) when the final time step has been reached. An example of the difference between iterative and noniterative samplings can be seen in Fig. 8.

## V. RESULTS

The results are given as the output of the threat-assessment algorithm for a number of simulated and real traffic scenarios. The algorithm currently runs on a standard desktop PC in a nonoptimized Matlab code, and the threat level of a typical scenario using 1000 samples takes 2 to 3 s to compute. In an optimized C code, increasing the performance by a factor of 100 to reach 30–50 Hz does seem realistic. In all these experiments, 1000 samples and a prediction horizon of 3 s were used.

### A. Simulated Data

Fig. 9 shows a scenario where an oncoming vehicle is forced into the lane of the host vehicle by an obstacle. Fig. 10 shows that the algorithm correctly identifies a situation as dangerous where the host vehicle is tailgating another vehicle. The two vehicles are not on a direct collision course, but there is a certain probability that the host vehicle will brake. Fig. 11 shows a situation where the host vehicle drives next to a bicycle.



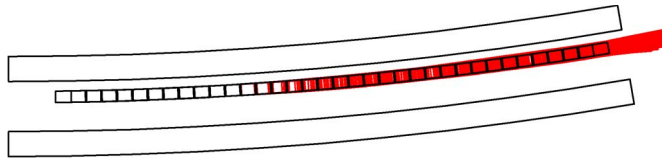


Fig. 10. Host vehicle is driving extremely close to another vehicle, both moving from left to right in the image. The system recognizes this as a dangerous situation even though the vehicles have the same velocity and are, thus, not on a direct collision course. The reason for this is that there is a certain probability that the leading vehicle will brake. Expected TTC is 2.8 s.

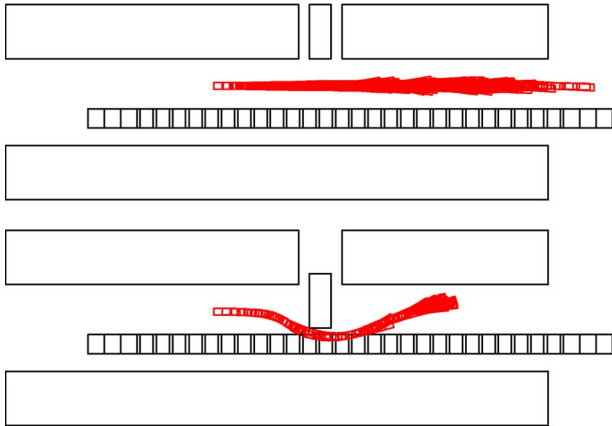


Fig. 11. Simulation in which a car and a bicycle travel from the left to the right. In the top scenario, there is no threat, but in the lower scenario, a vehicle is reversing, and the bicycle has to swerve to avoid it and, thus, becomes a threat to the host vehicle. In the bottom scenario, a threat is detected by the algorithm. TTC is 1.5 s.

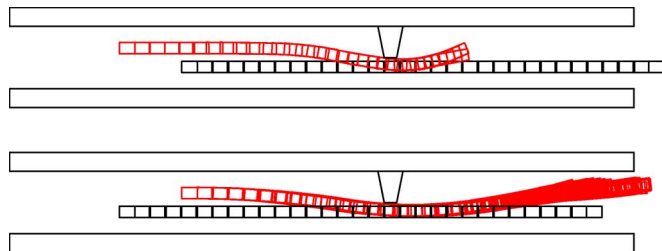


Fig. 12. Simulation demonstrating the properties of the visibility constraints. The (lower) host vehicle and (upper) another vehicle approach a narrow passage, moving right in the images. In the top scenario, no threat is identified since the host vehicle can be seen by the other vehicle and is likely to adapt its behavior to avoid a collision. The result is that the other vehicle stops or slows down to let the host vehicle pass first. In the bottom scenario, there is a higher probability that the host vehicle is not seen by the other vehicle and that it will steer toward the host vehicle. Here, a threat is detected, and the TTC in this scenario is 1.6 s.

The bicycle passes a parked vehicle, which suddenly starts to reverse, and the bicycle has to swerve and, thus, becomes a threat to the host vehicle. Fig. 12 shows the effects of the visibility constraints. Two vehicles approach a narrow passage, and the threat depends on whether the host vehicle can be expected to be seen by the other vehicle.

### B. Real Data

The system has been tested on recorded sensor data to evaluate the performance in authentic traffic situations. The data contain an information about the host vehicle, such as

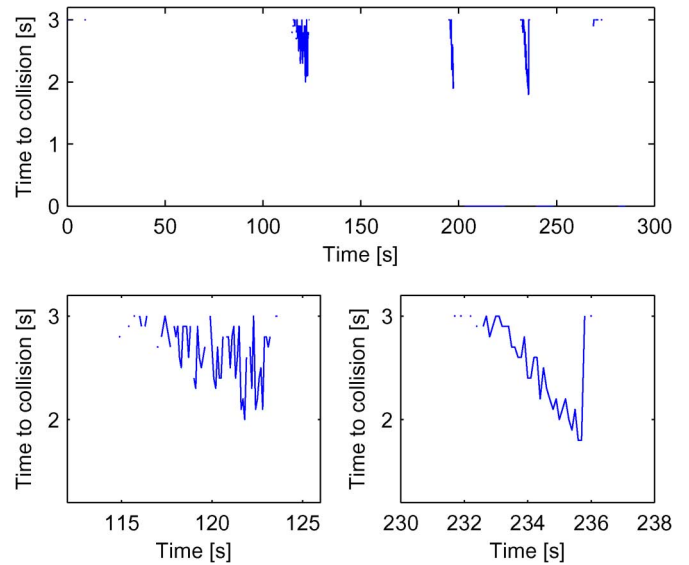


Fig. 13. Example of a 5-min data sequence. The top plot shows the threat levels for the entire time interval, and the two bottom plots show the detailed behavior for two interesting time intervals.

velocity and yaw rate, positions of other objects obtained from a fused vision/radar system, and information about the road shape and road boundaries that is given by the vision system. The radar system is a standard 77-GHz unit mounted in the grille of the vehicle that is typically used for adaptive cruise control application. The vision system is a black-and-white monosystem mounted at the rear-view mirror inside the windscreen. The data are filtered using an extended Kalman filter based on dynamic models of vehicles and the road (see, for instance, [9]). Currently, only vehicles can be detected with the radar/vision sensor combination, i.e., no other type of obstacle, which somewhat limits the variation in the types of scenarios that are assessed. All data are sampled at 10 Hz.

Another “problem” is that, although we are very thankful for it, the data do not contain any accidents. Instead, the idea is to get the threat-assessment algorithm to identify threatening situations, for example, situations where the host vehicle is driving close to a leading vehicle or is about to change lanes in the direction of a vehicle in the adjacent lane. A correct alarm is thus defined as an alarm that coincides with a traffic situation that might not be very dangerous but still poses some sort of threat. An alarm that does not correspond to a threatening situation is defined as a false alarm.

Fig. 13 shows an example of a 5-min sequence from the data set, where three threatening situations have been identified. A video clip of one of them is shown in Fig. 14, which is a situation where the host vehicle approaches a leading vehicle at a high relative velocity. The stochastic nature of the Monte Carlo-based algorithm becomes evident here. The threat value clearly has an increasing trend as the traffic scene evolves, but there is a large variance in the threat values. In fact, when applying the threat algorithm to the same scenario several times, the output threat level will vary slightly. The reason for this is the limited number of samples that are used. An example where more samples have been added is shown in Fig. 15.



Fig. 14. Video clip at about time 235 s from the data set in Fig. 13. The host vehicle is approaching a leading vehicle as it is about to overtake it.

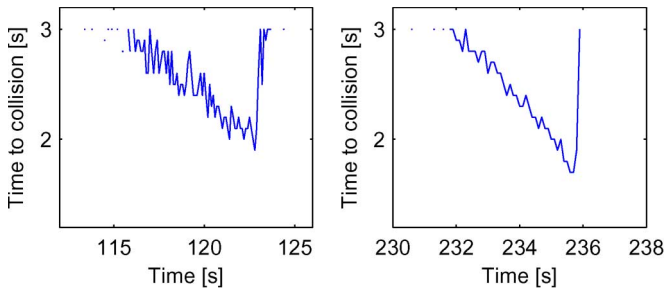


Fig. 15. Two sequences from Fig. 13 here run with 10 000 samples. The stochastic behavior is significantly reduced but, of course, at an increased computational cost.

TABLE I  
LONG-TERM PERFORMANCE

TTC threshold [s]	3.0	2.0
Correct alarms	156	67
False alarms	15	1
$T_{\text{correct}}$ [minutes]	1.70	3.96
$T_{\text{false}}$ [hours]	0.29	4.42

For 10 000 samples, the stochastic behavior is significantly reduced, but of course, the computational cost is much higher; it increases linearly with the number of samples. In a real application, however, these problems can be reduced by the fact that the system will have access to the threat levels from subsequent time steps. Thus, time averaging can be used to reduce noise.

For long-term performance evaluation, the algorithm has been run on a recorded data set of 4.5 h from a dense German autobahn traffic. A prediction horizon of 3 s and a sample set of 1000 samples were used. The number of alarms is shown in the first column in Table I. The first and second rows state the total number of correct and false alarms. The quantities  $T_{\text{correct}}$  and  $T_{\text{false}}$ , which are the average times between correct and false alarms, are presented on the last two rows. However, many of these alarms have a relatively high TTC value; thus, they do not pose a very high threat. If, for example, only alarms with TTC lower than 2 s are considered, the number of alarms will be significantly reduced (see column two of Table I).

A limitation in this paper is that there is not a lot of variation in the data set since all of it was taken from motorway traffic.

A result of this is that all the alarms that were given can be put into three different categories.

- 1) The host vehicle approaches a leading vehicle at high velocity.
- 2) The host vehicle drives close to a leading vehicle at about the same speed.
- 3) The host vehicle drifts out of its lane in the direction of a vehicle in the adjacent lane.

The false alarms are typically caused by vehicles in adjacent lanes. Their future positions are represented by a stochastic distribution, and sometimes, this distribution is a bit too spread out in the lateral direction. It is interesting to note that the already-existing automotive safety functions (forward collision warning [3] and emergency lane assist [13]) appear as special cases within this framework. These correspond to the aforementioned categories 1 and 3.

However, the reader is reminded that this paper does not suggest a new collision warning system; it merely demonstrates that the statistical framework that has been derived can be used to distinguish between threatening and nonthreatening situations in some respect. In a real-life application, one alarm every 4 min during normal driving is unacceptable.

## VI. CONCLUSION

A general threat-assessment algorithm is proposed that is able to incorporate many future hypotheses by regarding the control inputs of other vehicles as stochastic variables. Monte Carlo sampling is then used in order to compute a threat level. A method for creating dense sample sets without increasing computational load is also presented. For example, a result of this is that it becomes unnecessary to include domain-specific knowledge, such as emphasizing specific maneuvers or sets of maneuvers for vehicles when sampling.

The algorithm was tested both on simulated data and on authentic sensor data, and it successfully distinguishes between threatening and nonthreatening road scenes.

It was also noted that already existing automotive safety functions, such as forward collision warning and emergency lane assist, emerge as special cases within the general framework. This paper argues that reaching real-time performance regarding computational complexity seems feasible. The framework would thus be suitable both for online use in a safety application and for offline use to distinguish between threatening and nonthreatening situations in recorded data sets.

The first thing that would be interesting to improve is how the evaluation on real data is done. Whether a road scene is considered dangerous or not is currently based on a visual inspection of the recorded video, and it is, thus, not completely objective. A clear definition of what a dangerous situation is would be very desirable. Another alternative would be to have a panel of several people independently judging the same situations.

Another aspect that needs further investigation is the distribution of samples. The iterative-sampling process iteratively removes samples with conflicts and replaces them with copies of the conflict-free ones. Occasionally, a few of the initial samples are overrepresented in the final distribution. This means that

even though the number of samples in the final distribution is high, many of the samples are based on the same “parents,” and thus, the statistical variation can be somewhat limited. This is one of the sources of the stochastic noise in the threat level. How it is possible to reduce these problems and to create better variation in the final distributions is an interesting topic for further investigations. There are also many other possible future research topics, for instance, evaluating the algorithm on data with varying road and weather conditions. It would also be interesting to include “soft” obstacles to model lane markings. Another interesting topic would be to investigate if the visibility map could be improved using studies of driver behavior.

#### ACKNOWLEDGMENT

National ICT Australia is funded by the Australian Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australia’s ability and the ICT Centre of Excellence Program.

#### REFERENCES

- [1] R. Labayrade, C. Royere, and D. Aubert, “A collision mitigation system using laser scanner and stereovision fusion and its assessment,” in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, Jun. 2005, pp. 441–446.
- [2] E. Dagan, O. Mano, G. Stein, and A. Shashua, “Forward collision warning with a single camera,” in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun. 2004, pp. 37–42.
- [3] *Automotive Collision Avoidance System Field Operational Test*, 2000. Washington DC: Nat. Highway Traffic Safety Admin. [Online]. Available: [www.nhtsa.dot.gov/people/injury/research/pub/ACAS/ACAS-fieldtest](http://www.nhtsa.dot.gov/people/injury/research/pub/ACAS/ACAS-fieldtest)
- [4] A. Polychronopoulos, M. Tsogas, A. Amditis, U. Scheunert, L. Andreone, and F. Tango, “Dynamic situation and threat assessment for collision warning systems: The euclidean approach,” in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun. 2004, pp. 636–641.
- [5] J. Jansson, “Collision avoidance theory with application to automotive collision mitigation,” Ph.D. dissertation, Linköping Univ., Linköping, Sweden, Jun. 2005.
- [6] L. Yang, J. H. Yang, E. Feron, and V. Kulkarni, “Development of a performance-based approach for a rear-end collision warning and avoidance system for automobiles,” in *Proc. IEEE Intell. Veh. Symp.*, Columbus, OH, Jun. 2003, pp. 316–321.
- [7] A. E. Broadhurst, S. Baker, and T. Kanade, “Monte Carlo road safety reasoning,” in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, Jun. 2005, pp. 319–324.
- [8] A. Eidehall and F. Gustafsson, “Combined road prediction and target tracking in collision avoidance,” in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun. 2004, pp. 619–624.
- [9] Z. Zomotor and U. Franke, “Sensor fusion for improved vision based lane recognition and object tracking with range-finders,” in *Proc. IEEE Conf. Intell. Transp. Syst.*, Nov. 1997, pp. 595–600.
- [10] T. Bergh, J. Moberg, A. Strömberg, and L. Fridh, “Vägutformning 94 version s-2,” Swedish Nat. Road Admin., Borlänge, Sweden, 1994. Tech. Rep. 2002:113.
- [11] A. Eidehall, T. B. Schon, and F. Gustafsson, “The marginalized particle filter for automotive tracking applications,” in *Proc. IEEE Intell. Veh. Symp.*, Las Vegas, NV, Jun. 2005, pp. 369–374.
- [12] N. Gordon, D. Salmond, and A. Smith, “A novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *Proc. Inst. Electr. Eng.—F, Radar Signal Process.*, vol. 140, no. 3, pp. 107–113, Apr. 1993.
- [13] A. Eidehall, J. Pohl, and F. Gustafsson, “A new approach to lane guidance systems,” in *Proc. IEEE Intell. Transp. Syst.*, Vienna, Austria, Sep. 2005, pp. 108–112.



**Andreas Eidehall** received the M.Sc. degree in applied physics and electrical engineering and the Ph.D. degree in automatic control from Linköping University, Linköping, Sweden, in 2002 and 2007, respectively.

He is currently with the Department of Active Safety Functions, Volvo Car Corporation, Göteborg, Sweden. He is currently focusing on active safety applications and, in particular, collision warning, mitigation, and avoidance systems. His research interests lie in the area of statistical signal processing

with applications to safety-critical systems.



**Lars Petersson** received the M.Sc. degree in engineering physics and the Ph.D. degree in robotics and autonomous systems from the Royal Institute of Technology, Stockholm, Sweden, in 1997 and 2002, respectively.

Prior to joining the Vision Science, Technology, and Applications Group, National ICT Australia, Sydney, Australia, as a Project Leader for the Smart Cars project, he was a Researcher with the Australian National University, Canberra, Australia. His research interests are mainly in the

area of advanced driver-assistance systems as well as computer-vision-based perceptive systems.