Halmstad University Post-Print

Model-based estimation of driver intentions using particle filtering

Kristoffer Lidström and Tony Larsson

N.B.: When citing this work, cite the original article.

©2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Lidström K, Larsson T. Model-based estimation of driver intentions using particle filtering. In: 11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008. IEEE; 2008. p. 1177-1182.

DOI: <http://dx.doi.org/10.1109/ITSC.2008.4732623>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2303>

Model-based Estimation of Driver Intentions Using Particle Filtering

Kristoffer Lidström, *Student member, IEEE*, and Tony Larsson, *Member, IEEE*

Abstract—Proactive vehicle alert systems that warn the driver about dangerous situations must be able to reason about, and predict, likely future states of the traffic environment. Our prediction method is based on a combination of a fuzzy logic model for intersection turning behavior and Gipps model for car following behavior. The stochastic models are used together with a particle filter to recursively approximate the state probability distribution as measurements are received over time. Estimates of the unobservable part of the state are used to predict path choice and thus driver intentions. The approach is evaluated on trajectory data gathered from video footage of an intersection, however it is also relevant for trajectories acquired through vehicle-to-vehicle communication.

I. INTRODUCTION

THE step from designing passive vehicle safety systems to designing proactive vehicle safety systems is also a step from designing safety functions based on facts about what has already happened, to designing systems based on belief about what is going to happen. As the temporal horizon is extended the potential benefits increase. Systems that detect hazardous situations early on not only give the driver more time to react but also allow for less aggressive corrections of for example heading and speed.

The range of services that can be provided by a proactive alert system is large; however some definition must be adopted for the sake of reasoning and we suggest the following:

A proactive alert system is a system that takes as input relevant observations which include other vehicles, driver state and vehicle state as well as a-priori knowledge, such as traffic rules and road geometry. Based on current and previous inputs the system predicts one or several possible future states of the environment and issues warnings or recommendations to the driver in order to minimize the likelihood of a future hazardous state.

Notably, this definition includes the driver as part of the control loop, both when gathering inputs and as an actuator of control recommendations. Thus the assumption is that the driver is ultimately in control.

This work was carried out within the Vehicle Alert System (VAS) project at Halmstad University supported in part by the Knowledge Foundation, SP Technical Research Institute of Sweden, Volvo Technology AB and Free2Move AB.

K. Lidström and T. Larsson are with the Centre for Research on Embedded Systems, Halmstad University, Halmstad, Box 823, 30118, Sweden (phone: (+46)035-167385; e-mail: {kristoffer.lidstrom, tony.larsson}@hh.se).

Observations of the environment have traditionally been gathered through a number of onboard sensor systems such as radar, lidar and cameras. One main drawback of these types of sensors is that their functionality is often limited to line-of-sight conditions. However, early research [1] in safety systems based on vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communication suggested ways of handling these problems. With the advancement of the enabling technologies, embedded computing and wireless communication, the research community has regained interest in V2V and V2I systems. Although we in this paper use an infrastructure-based camera to gather vehicle positions over time, this type of information might as well have been transmitted directly from vehicle to vehicle via a radio link. We are concerned about what decisions to take based on the data rather than how it was acquired.

Instead of a machine-learning approach, where patterns in observed behavior serve to predict future behavior, we define explicit models that we then compare with observations. This approach allows for specifying rare behaviors for which it would otherwise be hard to gather training data as well as enables the explicit integration of regulations such as traffic laws.

Predicting the future in traffic scenarios is a notoriously difficult task that is influenced by a number of factors that are commonly divided into three main groups; *driver*, *vehicle* and *environment*. In the remainder of this section we briefly describe these groups in the context of predicting future traffic states. Section II describes our state space model involving vehicle speed, position and driver intentions. The state space is evolved between time steps according to driver behavioral models given in section III. The presented models describe the acceleration of vehicles during car-following and when intending to turn. In sections IV, V, and VI we integrate observations and models using particle filters in order to estimate the most likely intention of the driver. Finally we test the approach using recorded traffic data.

A. Environment

In everyday traffic common sets of rules enable drivers to predict each others behavior in a number of situations. This shared set of conventions makes it possible to define allowed and prohibited behavior. Implicit is also an assumption that the allowed behavior will be the norm among drivers. In the ideal case a high degree of coordination between drivers could be achieved using a highly detailed and restrictive set

of traffic rules. In reality safety is sacrificed for efficiency and generality, e.g. one does not always have to stop at a yield sign unless another vehicle is approaching.

We argue that an alert system must be able to model these types of artificial rules in order to accurately predict the future intentions of drivers. The system must also be able to incorporate the individual variations in how drivers choose to follow the rules. Traffic rules specified by, and for, humans are mainly of a qualitative nature. They describe general situations such as the relative position of vehicles to each other or to road geometry rather than explicitly stating exact distances or angles. Consider the following simplified statement describing the Swedish right-hand rule:

“At an unregulated intersection a vehicle must yield to other vehicles approaching from the right”.

From this short rule we can extract several properties that must be expressible if we want to model traffic rules, such as road geometry and topology (e.g. the intersection and road segments), time (definition of approaching) as well as the artificial concept of right-of-way.

In order to express these natural language rules that contain both discrete and continuous concepts we have chosen to use Fuzzy Logic for one part of our driver behavior model. In section III we describe the resulting control surface for velocity change when turning, constructed using this approach. The other part of the behavioral model, car-following, is defined using difference equations. The ability to integrate several models implemented using disparate techniques is one key problem that we address in this paper.

B. Vehicle

On a shorter timescale the physical characteristics of a vehicle play an important role in predicting its future location. Basic Newtonian physics relating to mass and velocity limit the type of maneuvers that can be carried out and are often expressed as differential equations. In [2] the authors design such a warning system around the extended Kalman filter, without taking into account the road geometry and with a prediction horizon of 2-3 seconds.

C. Driver

The third and final key component is the driver which, if observable, can indicate his or her intended control actions for example through body pose or gaze. Predicting human behavior is a large research area, with applications in many fields. Within the vehicular domain various machine learning approaches have been utilized, such as Hidden Markov Models [3] and Support Vector Machines [4].

Approaches that are based on machine learning often perform well, depending on the training data. However in traffic there are many situations for which it is very difficult to obtain training data such as most accident scenarios. Despite this, there is often a lot of expert knowledge formulated as explicit models. Our approach is to utilize this explicit domain knowledge in order to detect certain types of

driver behaviors.

In this paper we focus on the environment and vehicle components and try to infer what the driver intends to do. The assumption is that only basic vehicle kinematics is observable through for example position and velocity measurements, i.e. the effects of driver actions on the state of the vehicle. Much richer measurements can certainly be utilized and previous work has explored directly observing such parameters as driver pose and gaze [5]. However, the ability of groups of drivers to coordinate their actions despite being largely unaware of the detailed state of other drivers suggests that observations of only vehicle and environment state play an important role. An approach similar to ours for estimating driver intentions is taken in [6] where particle filters are used to estimate the lane-changing intentions of drivers.

II. STATE SPACE MODEL

Vehicles are modeled by describing the evolution of a state vector, consisting of vehicle kinematics and driver intentions, over time. By viewing the state as a random variable we are able to incorporate uncertainty in both the state evolution models as well as the observations. This allows for individual variations in how drivers implement the behaviors expressed by the model components defined in section III.

The road network is modeled as a directed acyclic graph, G , consisting of a number of links between vertices (Fig. 1). Links represent straight road segments between the vertices which are defined by two dimensional coordinates. Vehicles

Fig. 1. A frame from the video sequence (above) and the graph representation of the road geometry (below), the numbers correspond to link identifiers. A vehicle traveling west would exit in the top left corner of the frame on link 4.

are assumed to move only along paths in G which means that we map the two-dimensional measurements of vehicle location onto the closest point that lies on a link in G . Our system thus only considers longitudinal movements along the links in G . The driver of a vehicle approaching from the south on link 1 can thus have either the intention to turn left, (1,7,4), or to turn right, (1,2). Disregarding the position of the vehicle relative the center of the lane does have an adverse effect on the classification accuracy in some situations as is seen in section VI and should be weighed against the computational benefits of reducing the state vector dimensionality.

The most likely value of the unobservable part of the state vector, the driver intentions, is estimated recursively by first simulating a large number of evolutions of the state vector into the next time step and then comparing which evolutions most closely match the received measurement. More specifically the state of vehicle i at time k is given by the state vector

$$x_k^i = \begin{bmatrix} d_k^i \\ v_k^i \\ p_k^i \end{bmatrix}, \quad (1)$$

where d and v are the distance from the beginning of, and velocity along, the intended path p . The intended path p is a discrete variable indicating the road links that the driver intends to follow, e.g. (1,2) for a vehicle approaching from the south and then making a right turn. Thus, when we refer to the drivers *intentions* we mean a list of links that the driver plans to traverse.

The distance parameter d for each vehicle state vector is evolved based on the velocity at the previous time step and process noise is added

$$x_{k+1}^i = f(x_k^i) + \phi_k = \begin{bmatrix} d_k^i + v_k^i T \\ g(x_k^i, Y_k) \\ p_k^i \end{bmatrix} + \phi_k, \quad (2)$$

where $f(\cdot)$ is the process model, T is the sampling interval, and ϕ_k is zero mean Gaussian noise. The parameter p does not change between time steps, i.e. the noise added to it has zero mean and zero variance. The speed evolution, $g(x_k^i, Y_k)$, depends on the current vehicle state as well as the observations of all other vehicles at time k , Y_k , and is determined from the two model components described in the following section. Observations of the vehicle location are received every T seconds and relate to the state vector parameters as

$$y_k^i = \begin{bmatrix} d_k^i + \omega_k \\ h(p_k^i) \end{bmatrix}, \quad (3)$$

where ω_k is zero mean Gaussian noise and $h(p_k^i)$ are all links of the path up to and including the current one. Thus the velocity and future path choice remain hidden.

III. BEHAVIORAL MODELS

The speed evolution model consists of two main components, a car following component and an intersection turning component. The components each take as input an estimate of the state at a time step and output an estimate of the state at the next time step.

A. Car-following component

The car following component describes how a driver of a follower vehicle adjusts the own vehicle velocity to the velocity of a lead vehicle in front as well as how the velocity is adjusted when there is no lead vehicle, so-called free-flow conditions.

We have chosen the well-established Gipps model [7] to express car-following behavior. The model consists of two difference equations that describe the velocity evolution in free-flow (4a) and constrained conditions (4b) respectively. The minimum of the two difference equations is used as the velocity for the next time step (4c). Thus, in congested traffic drivers adjust their speed so that they are able to stop safely based on the estimated maximum deceleration of the lead vehicle.

$$v_f^a(t+T) = v_f(t) + 2.5A_f T \left(1 - \frac{v_f(t)}{v_f^{desired}} \right) \left(0.025 + \frac{v_f(t)}{v_f^{desired}} \right)^{1/2}, \quad (4a)$$

$$v_f^b(t+T) = -B_f T + \sqrt{B_f^2 T^2 + B_f \left[2[d_l(t) - d_f(t) - S_l] - T v_f(t) + \frac{v_l(t)^2}{\hat{B}_l} \right]}, \quad (4b)$$

$$v_{follow}(t+T) = \min(v_f^a, v_f^b), \quad (4c)$$

The parameters of equations (4a-4c) are explained in Table I together with any constant values used in our experiments.

Distance to the lead vehicle is calculated based on the *measured* position of the lead vehicle. A vehicle is said to be the lead vehicle if it is the closest vehicle in front of the follower along the followers intended path, indicated by the p variable in the state vector.

TABLE I
GIPPS MODEL PARAMETERS

Parameter	Description	Value used
T	Sampling interval	0.5s
A_f	Maximum desired acceleration for follower	2 m/s ²
v_f	Velocity of follower	Dynamic
$v_f^{desired}$	Desired velocity for follower	12 m/s
B_f	Maximum desired deceleration for follower	-2 m/s ²
$\hat{B}_l$	Followers estimate of lead vehicles maximum desired deceleration	-2 m/s ²
d_l, d_f	Position along a path in the road network graph G for lead and follower.	Dynamic
S_l	Lead vehicle length and minimum distance between vehicles.	3m

B. Turning component

The intersection turning component models the braking behavior of drivers as they approach a situation where they intend to make a turn. In order to decide whether the future path of the vehicle contains a turn, the path variable p of the state vector is used in combination with the graph G . The variable p can thus be seen as specifying the regime, or mode, that the system is in. Turns are identified by considering the difference in angle between segments on the intended path. If the angle difference is larger than a threshold it is considered a turn and we associate a preferred velocity v_{pref} with it.

The velocity evolution when approaching a turn is expressed as a set of fuzzy-logic rules that relate the antecedents, i.e. the time until reaching the turn t_{turn} and the preferred velocity v_{pref} to the consequent, brake force f_{brake} .

The resulting brake force is used together with the maximum deceleration parameter to calculate the component output.

$$v_{turn}(t+T) = v_k^i + B_f f_{brake} T \quad (5)$$

The control surface of the two-input one-output fuzzy controller is shown in Fig 2. It can be seen that the maximum brake force is exerted when the vehicle is very close to the intended turn and travelling at least 5 m/s faster than the recommended speed for that turn.

Fig. 2. The control surface of the fuzzy controller relates the output, brake force f_{brake} to the inputs, time until turn t_{turn} and the difference between the vehicle speed and the recommended speed for the turn.

C. Combining the components

If $f_{brake} = 0$ then $g(x_k^i, Y_k) = v_{follow}$, i.e. the velocity is determined fully by Gipps model as the driver is not braking for a turn.

If $f_{brake} > 0$ then $g(x_k^i, Y_k) = \min(v_{follow}, v_{turn})$, i.e. the driver is braking for a turn. In free-flow conditions the

velocity will be determined completely by the turning component, while if there is a lead vehicle the velocity will be the minimum of the two components outputs.

IV. INFERRING DRIVER INTENTIONS

The evolution of the state vector from one point in time to the next is defined by the model components described in the previous section.

In more general recursive Bayesian estimation terms our goal is to sequentially approximate the density $p(x_k^i | Y_{1..k})$ of vehicle i at each time step k in order to form estimates of the route choice variable p . Since our state evolution models are nonlinear we approximate the posterior non-Gaussian density at each time step using a set of weighted samples, commonly referred to as particle filtering [8].

Particles evolve between time steps according to the state evolution function (2), i.e. we use the transition prior $p(x_k^i | x_{k-1}^i)$ as the importance density. As measurements are received the particle weights are updated so that particles that behave likely with respect to the measurements are weighted more heavily. The weights are computed based on the difference between the distance parameter of the state vector and the distance parameter of the measurement vector. The likelihood function used is a normal distribution centered on the state vector distance parameter.

Over time the variance of the weights will increase for our choice of importance density, as shown in [9]. The standard solution to this is re-sampling, where particles with low weights are removed and replaced by copies of particles with high weights. In our experiments we have used the systematic resampling algorithm [10].

As particles with high weights are replicated the more probable areas of the state space will be approximated with a greater number of particles. However since the state vector contains a static component, the driver intentions, there is a risk that all particles with a specific value of the intention parameter are completely removed in the resampling step should they have low weights. Since the intention part of the state does not change between time steps, there will no longer be any particles describing that specific part of the probability space. To counteract this problem we divide the particles into groups with identical intention parameters and only resample within each group. This ensures that there will always be particles for each value of the intention parameter. The drawback is that particles with very low weights are kept in some cases. The number of particles is adjusted so that there are always a fixed number of particles per vehicle.

This approach is similar to the interacting multiple-model (IMM) particle filter [11] used for jump Markov nonlinear systems (JMNLs). The JMNLs IMM particle filter essentially runs a filter for each possible mode of the system and mixes the estimates based on the mode-change probabilities. In our system these mode-change probabilities are zero, i.e. the path parameter never changes, which means that the estimates for the mode-matched filters are not mixed.

The benefit, however, is that we can dedicate a fixed number of particles for each value of p in order to avoid the degeneracy mentioned earlier.

V. EXPERIMENT

The test data consists of two dimensional vehicle trajectories at a three-way, non-signalized, intersection. The trajectories have been manually extracted from a thirty minute video sequence of the intersection shot from an elevated location. In total the data consists of about 6700 data points that describe the motion of in total 251 vehicles with a resolution of two measurements per vehicle and second. The data was recorded during a weekday around noon and the driving conditions were good with daylight and unrestricted visibility.

The trajectories in camera oriented coordinates were projected onto an aerial photograph of the intersection and a representation of the road segments using the graph notation presented in section II was created (Fig. 1). Characteristics of the data are presented in Table II. A sequence of link identifiers are used to indicate a path through the graph, for example, the sequence 1, 7, 4 indicates a vehicle approaching from the south and turning left towards the west.

TABLE II
NUMBER OF OBSERVATIONS AND AVERAGE SPEED

Path	Number of vehicles	Average speed (km/h)
3, 9, 6	24	18.3
3, 8, 4	98	36.3
5, 10, 2	109	41.1
5, 6	2	24.7
1, 2	17	19.3
1, 7, 4	1	24.7
All	251	35.4

VI. RESULTS

The particle filter (using 400 particles per vehicle) was run on the collected data and for each time step the weights corresponding to a specific path choice for each vehicle were summed. A final decision about the intentions of a specific vehicle was made when the sum of weights for a particular path choice exceeded a fixed threshold α . A ground truth path choice was obtained using the first and last measurements for each vehicle. As a vehicle first enters the scene the system determines the initial segment it is on. Particle distance and velocity parameters are initially instantiated according to a normal distribution centered on the location and velocity of the vehicle when it was first measured. The intention parameter is drawn from a uniform distribution over all possible path choices that can be made from the initial location.

Table III indicates the number of correct and false classifications of path choices for $\alpha=0.9$. Only classifications that were made prior to reaching the intended segment are included. For the indicated value of α the

TABLE III
CONFUSION MATRIX FOR INTENTION CLASSIFICATIONS, $\alpha=0.9$

True	Classification					
	3,9,6	3,8,4	5,10,2	5,6	1,2	1,7,4
3, 9, 6	13	3	-	-	-	-
3, 8, 4	12	80	-	-	-	-
5, 10, 2	-	-	86	14	-	-
5, 6	-	-	0	2	-	-
1, 2	-	-	-	-	0	1
1,7,4	-	-	-	-	0	0

system correctly predicts the driver intentions in 181 cases, incorrectly in 30 cases and does not make a prediction in 40 cases. Situations when no prediction is made occur when no estimated state has a probability above the threshold α before actually entering onto the intended segment. Classifying the intended segment after entering onto it is trivial, and indeed all 40 previously undecided cases are correctly classified after passing this point. Table III also indicates that the system poorly predicts whether a driver is about to turn left or right when approaching the intersection on link 1. In this case, since a turn must be made, the braking behavior according to the model is very similar regardless of whether the driver is making a right or left turn. Knowledge of, and support in the model for, the lateral position of the vehicles would most likely improve the functionality of the system in this case as most drivers position themselves differently depending on whether they are turning left or right. Another way is to elicit information about the intended path from the driver directly, as is already done today through the use of turn indicators. A third option would be to specify more detailed recommended speeds for the two turns based on the observed data, in our experiment we have chosen the same recommended speed for all turns (3 m/s). However, as the models are tuned for a specific dataset the risk of overfitting increases.

Fig. 3 indicates the classification accuracy for various values of the threshold parameter. By varying the threshold it can be seen that if we want to make classifications early then we sacrifice accuracy, indeed if the classification is made even before the driver has started to adapt the vehicles speed to the maneuver then we expect very poor accuracy

Fig. 4 shows how increasing the threshold forces the system to wait longer before making a decision.

VII. CONCLUSIONS

We have shown how explicitly formulated models of driver behavior can be utilized to predict driver intentions. Differences in driver behavior are incorporated by simulating a large number of stochastic state evolutions and updating based on how close they resemble the received measurements. The main advantage of our approach when compared to learning systems is that the models are simple, transparent and lend themselves to analysis by domain experts. The interpretability of the models is important as the system is meant not only to predict behavior but also to dictate it. For example, if drivers continuously violate the

Fig. 3. The fraction of true and false classifications as well as the fraction of cases when no hypothesis was above the threshold before performing the intended maneuver.

Fig. 4. The average distance before reaching the start of a correctly predicted next link.

right-hand rule at an intersection the system still needs to have a notion that this is disallowed behavior, albeit the norm.

We have implemented a prototype of the system and carried out experiments on a set of recorded traffic data, the results show that even with rather simple and intuitive models of driver behavior the system is often able to predict the intended path of the driver through an intersection based only on change in speed and the vehicle location relative to the road geometry and other vehicles in front of it. Our prototype implementation does not run in real-time. However the problem size, i.e. the number of vehicles in a scene at any given time, should be considered a constant in comparison to the expected increase in computational performance of the platforms used in vehicular applications.

In future work we would like to explore the benefits that can be gained by extending the state vector to include more than one vehicle, i.e. estimating the intentions of a group of vehicles. The computational performance required to

estimate a state vector of high dimensionality is a key problem and finding ways to partition vehicles into smaller related groups is a further continuation of our work in this direction.

Our work focuses on designing proactive safety systems for networks of intelligent vehicles and thus another important research direction is how to formulate warning criteria based on the types of intention predictions we present in this paper. Threat assessment mechanisms based on Monte Carlo simulation of driver models have recently been suggested [12], [13], and the ability of the approach to act as a generalized warning system has been emphasized. Intuitively the ability of drivers to predict the future traffic situation is a good measure of safety, if an intention prediction system like the one presented in this paper is unable to choose between hypotheses then that could act as an indicator of a potentially hazardous, or unpredictable, situation.

REFERENCES

- [1] Williams, M., 'PROMETHEUS-The European research programme for optimising the Road Transport System in Europe', in *Driver Information, IEE Colloquium on*, 1988, pp. 1-9.
- [2] H.-S. Tan and J. Huang, "DGPS-Based Vehicle-to-Vehicle Cooperative Collision Warning: Engineering Feasibility Viewpoints," in *IEEE Trans. Intelligent Transportation Systems*, vol. 7, Dec. 2006, pp. 415-428.
- [3] R. Simmons, B. Browning, Y. Zhang and V. Sadekar "Learning to Predict Driver Route and Destination Intent," *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 127-132.
- [4] H. M. Mandalia and D. D. Salvucci, "Using support vector machines for lane change detection," in *Proc. Human Factors and Ergonomics Society 49th Annual Meeting*, 2005, pp. 1965-1969.
- [5] S. Y. Cheng and M. M. Trivedi, "Turn-Intent Analysis Using Body Pose for Intelligent Driver Assistance," in *IEEE PERSASIVE COMPUTING*, vol. 5, Oct.-Dec. 2006, pp. 28-37.
- [6] J. Gunnarsson, L. Svensson, E. Bengtsson, and L. Danielsson, "Joint Driver Intention Classification and Tracking of Vehicles," in *IEEE Nonlinear Statistical Signal Processing Workshop*, Sept. 2006, pp. 95-98.
- [7] P.G. Gipps, "A behavioural car-following model for computer simulation," in *Transp. Res. B*, vol. 15, April 1981, pp. 105-111.
- [8] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," in *IEEE Trans. Signal Processing*, vol. 50, issue 2, Feb. 2002, pp. 174-188.
- [9] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," in *Statistics and Computing*, vol. 10, no. 3, Jul. 2000, pp. 197-208.
- [10] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," in *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, Mar. 1996, pp. 1-25.
- [11] H. Driessen and Y. Boers, "Efficient particle filter for jump Markov nonlinear systems," in *IEE Proc. Radar, Sonar and Navigation*, vol. 152, no. 5, Oct. 2005, pp. 323-326.
- [12] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo Road Safety Reasoning," in *Proc. Intelligent Vehicles Symposium*, Jun. 2005, pp. 319-324.
- [13] S. Danielsson, L. Petersson, and A. Eidehall, "Monte Carlo Based Threat Assessment: Analysis and Improvements," in *IEEE Intelligent Vehicles Symposium*, Jun. 2007, pp. 233-238.