<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*.

Citation for the original published paper:

Annell, S., Gratner, A., Svensson, L. (2016)

Probabilistic collision estimation system for autonomous vehicles

In:

<https://doi.org/10.1109/ITSC.2016.7795597>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-249496>

Probabilistic Collision Estimation System for Autonomous Vehicles

Stefan Annell^{1,2,3}, Alexander Gratner^{1,2,3} and Lars Svensson¹

Abstract—Nearly 1.3 million people die each year in traffic-related accidents, whereas an additional 20-50 million people are injured. Introducing autonomous vehicles would aim to reduce these numbers by removing the driver from the loop entirely and thus removing the human error. Intersections are considered a complex traffic situation for autonomous vehicles. Functions which could accurately foresee future events in those situations, mimicking the situation awareness of humans, would improve autonomous systems and increase traffic safety.

To address this a system is designed with two main functionalities: estimate the movements of a observed vehicles in a general traffic situation and predict the probability of a collision, given the current ego trajectory. This system could either be used as information and feedback for a trajectory planner or as a support for decision making at higher level system monitoring.

The main contributions are the robust system design, that robustly and consistently estimates the likelihood of a collision and thus preventing future collision, and the intention estimation which determines the probability of which route through an intersection an observed vehicle will take through an intersection by using its current state. The system is validated by controlling the ego vehicle's velocity with a Velocity Planning Controller to avoid colliding. It is shown that in terms of robustness to noise the system successfully avoids collision.

I. INTRODUCTION

Nearly 1.3 million people die each year in traffic-related accidents, whereas an additional 20-50 million people are injured or disabled [1]. These numbers serve as motivation for initiatives such as the Zero Vision approach which originated in Sweden, stating that *no loss in life is acceptable* [2], or the No Fatal Accidents by 2020 goal set by Volvo Cars' [3].

Collision avoidance systems for autonomous vehicles can be divided into two methodologies; sensor-based and Intelligent Transportation Systems (ITS)-based. Sensor based approaches rely on on-board sensors and perception algorithms to retrieve information of the vehicles surroundings while ITS-based systems utilize either Vehicle to Vehicle (V2V) or Vehicle to Infrastructure (V2I) communication to retrieve information. According to [4], both methodologies face several challenges in order for them to be viable alternatives in future intelligent vehicles. Sensor based systems are limited to field-of-view and are still sensitive to unfavorable weather conditions. ITS systems are on the other hand prone to errors when sending information and this also opens up a channel that brings security issues. Which is why automotive manufacturers have developed the Dedicated Short-Range Communications (DSRC) standards for communication in

Fig. 1. An overview showing the proposed system, its inputs and outputs.

such systems [5]. The information collected using either of the two methodologies described above is then used to predict/assume future behaviors of vehicles present in the scene.

The proposed system is a supporting system for a fully autonomous vehicle, as seen in Fig. 1. It avoids dangerous traffic situation by using the observed state of a vehicle and make long-term trajectory predictions along intention paths. The two outputs are a Probability Field that could be used by a trajectory planner and a Collision Probability which could be used in a higher level part of the system or as a controller. As a proof of concept, the collision estimation is used as feedback through a Velocity Planning Controller (VPC) that modifies the ego vehicle's velocity to avoid collisions.

The first main contribution is the system design with its probabilistic approach to estimate possible future positions of an observed vehicle as a distribution in time and therefore is also able to predict when and where a collision could occur. This is validated in terms robustness to noise in the object tracking. The second contribution is the intention estimation that estimates the most likely intention in a intersection of the observed vehicle depending on its current state.

¹KTH – Royal institute of Technology

²ÅF Technology AB

³These authors have done the same amount of contribution to this article.

Fig. 2. A sequential image of the system, where the top row shows the future estimated positions of the ego and observed vehicle as ellipses and potential collision points as red dots. The bottom row shows the collision probability at future time-steps t_{vir} given different control inputs u_{ego} applied to the ego vehicle over the time horizon.

TABLE I
DESCRIPTION OF THE SYMBOLS THAT ARE USED IN THE SYSTEM.

Symbol	Description
x_t	Vehicle state ($x, y, \theta, \dot{x}, \dot{\theta}$)
ξ	Map data
$\tilde{\psi}(i)$	A set of Intention path probabilities
$\tilde{\Omega}(t_{vir})$	A set of Vehicle trajectories
$\tilde{G}(t_{vir})$	A set of Gaussian distributions
$\hat{P}(t_{vir})$	A set of Probability fields
$\hat{P}(t_{vir})$	Collision probability along given ego trajectory

This paper is organized as follows. Section II presents a discussion of the related work. Section III describes the proposed system which estimates future collisions for the vehicle. In Section IV, we discuss the different subsystems. The system setup is described in Section V and the results of two different cases are presented in Section VI. Section VII summarizes the work and defines its limitations.

II. RELATED WORK

Various system structures related to future prediction algorithms have previously been proposed by numerous authors. In [6], the system obtains vehicle and environmental evidence, such as vehicle states and lane lines, in order to detect probable maneuvers for each vehicle in the scene using bayesian inference. These results are then fed forward to a manoeuvre specific prediction algorithm which estimates future motions in order to later on calculate collision probability estimates and criticality metric calculations. A similar system is presented by [7] whereas target tracking and road marking is provided to a trajectory prediction system which combines a kinematic vehicular model with a manoeuvre recognition module. In [8], a situation hypothesis selection framework is presented which selects the ego manoeuvre based on situation recognition and risk minimization.

In the research presented in [7], [9], it is argued that the trajectory gained from a kinematic model is accurate for a short time prediction while the trajectory obtained from a manoeuvre identification, such as changing lane, is more accurate for long time predictions. Another similar approach is presented in [10]. In their work, the dynamic model is used for short term predictions and the road geometry is explicitly used for long term predictions.

A method to predict vehicle movement using a particle filter, namely [11]. [12] continued on his work and developed it further, suggesting different kinds of models for the vehicles and also different weighting methods. Another approach is to apply a Kalman Filter (KF) as in [13] and make a prediction along a trajectory. However, only the collision estimation part is evaluated, the modified Kalman filter is only shown that it works and it is not investigated in terms of for example robustness. In [14] the intention for a detected vehicle is determined by using a multilayer Hidden Markov Model, the distribution of possible movements is calculated with Gaussian Process and the risk estimation is achieved by sampling from the Gaussian Processes and comparing it with the ego trajectory.

III. SYSTEM DESCRIPTION

There are two main functionalities of the system:

- 1) Generate a probability field representing possible future positions of the observed vehicle. This can be used as a support for the trajectory planner in the ego vehicle to be able to create better trajectories that avoids colliding with observed objects. The probability field is defined as $\hat{P}(t_{vir})$ and $t_{vir} = [0, t_{horizon}]$, t_{vir} is the predicted time and $t_{horizon}$ is the predicted time horizon. And $\hat{P}(t_{vir})$ contains Gaussian distributions describing the possible future states of the observed vehicle.

Fig. 3. The estimated probability of a collision for the ego vehicle along a trajectory.

- 2) Estimate how likely a collision is. Collision probability uses the generated probability field and a given trajectory and estimates how likely a collision is and when it would occur, given that the trajectory is followed. The collision probability is defined as $\bar{P}(t_{vir})$.

A sequential image scenario which shows how the system works is shown in Fig. 2. The probability field is applied to both the ego vehicle and the observed vehicle. The VPC tests different control inputs and chooses the one with lowest collision probability. The top row shows the probability field of the observed vehicle in orange and the chosen control input for the ego is the green probability field. The estimated collision points for all control options are also shown. The bottom row of the figure shows the collision predictions and from there the controller input is chosen at each time step.

A summary of the internal communication through the system, as well as inputs and outputs, are described in Table I and a system overview is shown in Fig. 1. All the subsystems are listed below:

- **Inputs** originate from the environment, object tracking and possibly a trajectory planner. The trajectory planner input is optional, but required if collision probability function is to be used.
- **Intention Estimation** receives the observed object states x_t^{obj} through a sensor, as an input as well as the map data ξ . It then checks how similar the current state of the object is to reference paths (e.g. turn left or go straight) in order to assign probability values ψ to each of these. The output is a vector of probability values $\bar{\psi}$ that indicates how probable all reference paths are depending on the current state of the vehicle.
- **Long-term Trajectory Predictions** gets the object state x_t^{obj} , map data ξ and the probability vector $\bar{\psi}$ derived by intention estimation as inputs. Its goal is to calculate the predicted trajectories $\bar{\Omega}_{obj}$ for the observed object given the intention trajectories with highest probability values.
- **Uncertainty Estimation** obtains the predicted trajectories $\bar{\Omega}_{obj}$ from the path prediction along with the object state x_t^{obj} and intention path probability vector $\bar{\psi}$. A Gaussian uncertainty is estimated along the predicted

$\bar{\Omega}$ and the output is the set of distributions $\bar{G}(t_{vir})$.

- **Probability Fields** is a set of Gaussian distributions $\bar{G}$. They are combined with the probability vector $\bar{\psi}$ related to each intention path in order to produce a probability field. To make a single probability field for one set of estimations, the following is done

$$\hat{P}(t_{vir}) = \bar{\psi} \cdot \bar{G}(t_{vir}) \quad (1)$$

where $\hat{P}$ holds the estimation of the target objects future position. The produced field can be an output to a trajectory planner, which in turn could use this to predict safer trajectories, or as input to the collision probability algorithm.

- **Collision Probability** uses the probability field $\hat{P}$ and the ego trajectory $\bar{\Omega}_{ego}$ to calculate potential collision points by drawing samples from the probability field and comparing these with the ego states along the trajectory.
- **Outputs** are the probability field and the collision probability. Firstly, a trajectory planner may use the probability field to make safe trajectories and feeding back the predicted trajectory to evaluate it, thus closing the loop. The other output is the probability of collision given a certain ego trajectory.

IV. SUBSYSTEMS

A. Intention Estimation

The intention estimation is generated by comparing the current state of the observed vehicle x_t to idealized paths placed in the middle of the road, these are from now on called intention paths and can be seen in Fig. IV-A. Such paths have been used in by other authors, e.g. [15]. The paths contains position heading and ideal speed and are described by using cubic hermite splines. To calculate how likely a intention path is the following is used

$$\psi_i = e^{-\sqrt{\sum_{i=0}^M |x_t^{obs} - x_i^{intention}|^2}} \quad (2)$$

where $x_i^{intention}$ is the closest state on the path to the observed vehicle and M is the number of intention paths. The intention estimations are then normalized according to

$$\bar{\psi}(i) = \frac{\psi_i}{\sum_{i=0}^M \psi_i} \quad (3)$$

$$\sum_{i=0}^M \bar{\psi}(i) = 1$$

where the intention with the highest value is the most likely.

B. Long-term Trajectory Predictions

The chosen approach for making long-term trajectory predictions in the system is based of the studies performed by [9], [10]. The method utilizes map information coupled with speed gradients to obtain reference trajectories describing the centreline of all different manoeuvre in an intersection, such as going straight through or turning. The state of the observed vehicle is compared to the state of all reference trajectories at the point closest to the car.

Fig. 4. T-intersection with intention paths.

Fusing two trajectories can be done by comparing states at the same time instances and combining them as done in [10] and [9]. The method proposed by [9] is designed for lane changing purposes and does this by utilize a weighting function to control the impact of each trajectory at every time instance t . The final path can therefore be calculated as

$$\Omega_{res} = w\Omega_{model} + (w-1)\Omega_{intention} \quad (4)$$

where w is the weight factor obtained by the weighting function, as shown in Fig. 5.

The weight function used in (4) for fusing together the two trajectories has been chosen as

$$w(t_{vir}) = \begin{cases} \frac{1 + \cos(\frac{\pi t_{vir}}{t_{break}})}{2}, & t_{vir} \leq t_{break} \\ 0, & t_{vir} > t_{break} \end{cases} \quad (5)$$

where t_{break} indicates the future time state where the reference path should be seen as the absolute truth. This function was inspired from the weight function introduced by [10] and was chosen for its simplicity.

C. Uncertainty Estimation

Given a state x_t and a trajectory Ω the applied method is described by Lambert [13] where he applies a KF

$$G_t = F_{t-1}G_{t-1}F_{t-1}^T + Q_{t-1} \quad (6)$$

Fig. 5. The resulting trajectory Ω_{res} is obtained by fusing the model trajectory Ω_{model} and the intention trajectory $\Omega_{intention}$.

by using the model of a nonlinear vehicle movement function to describe covariance matrix in a long term prediction along a trajectory. The vehicle model

$$\begin{aligned} \bar{x}(t + \Delta t) &= f(\bar{x}(t), \Delta s, \Delta \theta) \\ &= \begin{pmatrix} x(t) + \Delta s \cdot \cos(\theta + \Delta \theta / 2) \\ y(t) + \Delta s \cdot \sin(\theta + \Delta \theta / 2) \\ \theta(t) + \Delta \theta \end{pmatrix} \end{aligned} \quad (7)$$

is used and linearised with a *first order taylor expansion*

$$\begin{aligned} F_{t-1} &= \left(\frac{\partial g}{\partial x_{t-1}} \right) \\ &= \begin{bmatrix} 1 & 0 & -\Delta s \cdot \sin(\theta_{t-1} + \Delta \theta / 2) \\ 0 & 1 & +\Delta s \cdot \cos(\theta_{t-1} + \Delta \theta / 2) \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (8)$$

where Δs describes the longitudinal movement and $\Delta \theta$ describes the rotation between each time step. (6) is then applied to generate the Gaussian distribution G along a given trajectory.

D. Collision Probability

The probability of two vehicles colliding can be described/evaluated using Gaussian normal distributions. One needs to consider both distributions of the two objects and check for collision

$$P_{coll} = \int p_o(x_o, y_o, \theta_o) \cdot p_e(x_e, y_e, \theta_e) \cdot dx_o dy_o d\theta_o dx_e dy_e d\theta_e \quad (9)$$

where p_o and p_e is the Probability Density Function (PDF) at a specific point for the observed and the ego object. However, one problem that occurs is that this solution does not take object volume into account and therefore the likeliness of collision is reduced compared to the actual collision probability. A solution to this problem was suggested by [13], where a Monte Carlo (MC) based method to solving this integral is applied. This is done by generating random samples from the two Gaussian distributions and boundary checked if they occupy the same space or not, according to

$$P_{coll} = \int \varphi p_o(x_o, y_o, \theta_o) \cdot p_e(x_e, y_e, \theta_e) \cdot dx_o dy_o d\theta_o dx_e dy_e d\theta_e \quad (10)$$

where

$$\begin{aligned} \varphi &= \varphi(v_o(x_{o,i}, y_{o,i}, \theta_{o,i}), v_e(x_{e,i}, y_{e,i}, \theta_{e,i})) \\ &= \begin{cases} 1 & \text{if } v_o(x_o, y_o, \theta_o) \cap v_e(x_e, y_e, \theta_e) \neq \emptyset \\ 0 & \text{if } v_o(x_o, y_o, \theta_o) \cap v_e(x_e, y_e, \theta_e) = \emptyset \end{cases} \end{aligned} \quad (11)$$

which tests if the two objects occupy the same space, returns 1 if they do and 0 otherwise. $v_o(x_{o,i}, y_{o,i}, \theta_{o,i})$ is the volume occupied by the observed vehicle and $v_e(x_{e,i}, y_{e,i}, \theta_{e,i})$ is the ego vehicles volume. As described (10) will then converge down to the true probability due to the *Strong Law of Large Numbers*. Therefore the probability can be estimated using an empirical average

$$\bar{\psi} = \frac{1}{N} \sum_{i=0}^N \varphi(x_{o,i}, y_{o,i}, \theta_{o,i}, x_{e,i}, y_{e,i}, \theta_{e,i}) \quad (12)$$

Fig. 6. Illustration test case 1. The green ego vehicle move along the green trajectory and the path of the observed orange vehicle may vary within the orange shaded area and its velocity may vary inside the interval shown to the right.

where N is the number of samples used in the estimation. This is used at each time step t_{real} and predicted time t_{vir} for the whole predicted horizon.

E. Planning Controller

At each time instant a new controller input is chosen and the criteria for choosing what input to use is the lowest collision probability along the whole prediction horizon. Let I, J be indexing sets such that

$$\begin{aligned} I &= \{i \in N | 1 \leq i \leq u_{max}\} \\ J &= \{j \in \mathbb{Z} | 0 \leq j \leq t_{horizon}\} \end{aligned} \quad (13)$$

let M be the set of maximum row values of the tensor $u_i(t_{vir}^j)$

$$M := \{m_i | i \in I; m_i = \max(u_i(t_{vir}^j)) \quad \forall j \in J\} \quad (14)$$

The minimum element in this set is the control input that yields the lowest probability of collision. Therefore we seek

$$x \in I \text{ such that } m_x = \min(M) \quad (15)$$

thus, u_x is the chosen control value at each time step. Fig. 3 shows the system output during the span of the simulation time, using the controller thus avoiding a collision. Using (13) to select a controller input at each time, and avoiding collision. A priority order is implemented to handle the situation where several control inputs are equal in regards of collision probability.

V. CASE SETUP

The system is evaluated by calculating the frequency of collisions for two different simulated cases, as shown in Fig. 6 and Fig. 7. The VPC is initialized with a priority list, assigning a priority number to each possible control input. The priority list favors braking motions over accelerating motions and solves the possible scenario of two controllers having equally low collision probabilities.

Two variables are varied for each case: *Object Tracking Noise*, which is zero mean normal distributed σ and affects the perceived position of the observed vehicle, and the *Number of samples* which are used for the collision detection calculations. Moreover, for each noise/sample combination, the path and velocity profile of the observed vehicle is

Fig. 7. Illustration test case 2. The green ego vehicle move along the green trajectory and the path of the observed orange vehicle may vary within the orange shaded area and its velocity may vary inside the interval shown to the right.

randomized 13 times within the boundaries depicted in Fig. 6 and Fig. 7.

In the first case, as shown in Fig. 6, the observed vehicle has velocity profiles that may vary within 50 – 90km/h. Its path is fairly close to ideal, i.e., a path which is not diverging significantly from the center of its lane. The second case however, as shown in Fig. 7, allows a much more aggressive velocity profile in contrast, where a sudden braking motion is performed in the middle of the intersection. Furthermore, its path may vary within a wider range that allow the vehicle to enter the opposing lane. The second case are thus to be considered as a more erratic case which shall be more difficult for a collision estimation system to handle.

VI. RESULTS

The results of the simulations for each case can be seen in Fig. 8 and Fig. 9. The collision frequency of the 13 independent runs that represents one data point in the graphs is shown as a color in a color scale ranging from blue to red. Both cases shows that the collision frequency decrease when the number of samples increases, since more samples describes the distribution more accurately.

Case 1 is shown in Fig.8. The collisions occurred when the number of samples used for the collision estimation drops below ~ 150 . Noise is handled without any problems, although two light blue dots indicates that some arbitrary collisions may occur when the noise is increased. One thing to note is that the noise was normal distributed, and this is the 1σ value.

Case 2 is shown in Fig. 9. Here, collisions also occurs when the number of samples drops below ~ 150 . However, the collision frequency increase when increasing the sensor noise levels, thus indicating that this is a harder case for the system to handle. The non-monotonic behavior is a result of the high sensor noise injected in the system.

VII. CONCLUSIONS

The suggested system that makes future time predictions in order to make decisions is similar to how most humans approach traffic situations in real life. It is shown that with this approach, collisions can robustly be avoided in terms of object tracking noise if enough samples are used to estimate the collision probability. One thing to note is that the type of

Fig. 8. The results from Case 1 simulations showing the collision frequency when the number of samples and sensor noise are varied.

Fig. 9. The results from Case 2 simulations showing the collision frequency when the number of samples and sensor noise are varied.

controller uses the collision estimation to avoid a collision by picking the controller with lowest probability of collision at each time step. This is an aggressive approach, since it does not take for example, distance to other vehicles into consideration and implementation that would instead use the probability field to calculate a trajectory around the collision could be a more defensive and safer approach.

The system functionality and the utility gained from this kind of predictions are very interesting to give the autonomous vehicles a higher level of situational awareness. The fact that this system is not designed for a specific traffic situation, or even a intersection, makes it a very useful core functionality of a autonomous vehicle the only thing it needs is a environment map together with the intention paths which are placed in the middle of the road. Since it is indicated by the simulations that the with this type of implementation the noise level needs $\sigma \leq 0.75$ using purely an object tracking system with GPS and communication might be problematic, therefore a sensor fusion approach might be better.

Possible directions for future work includes implementation of the system on an real-time embedded platform, further research on the subsystems introduced in the paper as well as research on interactions between multiple vehicles.

REFERENCES

- [1] ASIRT. Asirt- association for safe international road travel. <http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics>, 2016. [Online; accessed 15-May-2016].
- [2] Visionzeroinitiative. Vision zero initiative. <http://www.visionzeroinitiative.com/>, 2016. [Online; accessed 15-May-2016].
- [3] Volvo. Volvo , no fatal accidents by 2020. <https://www.media.volvocars.com/global/engb/media/pressreleases/152776/360-view-technology-key-to-volvo-cars-goal-of-no-fatal-accidents-by-2020>, 2016. [Online; accessed 15-May-2016].
- [4] Youssef A Bazzi, Mohannad Murad, Nizar Al-Holou, and Rawa Adla. Automotive collision avoidance methodologies sensor-based and its-based. In *2013 ACS International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2013.
- [5] John B Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.
- [6] Matthias Schreier, Volker Willert, and Jurgen Adamy. An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–16, 2016.
- [7] Adam Houenou, Philippe Bonnifait, and Veronique Cherfaoui. Risk Assessment for Collision Avoidance Systems. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Qingdao, China, 2014.
- [8] Stefan Klingelschmitt, Florian Damerow, and Julian Eggert. Managing the complexity of inner-city scenes: An efficient situation hypotheses selection scheme. In *Intelligent Vehicles Symposium (IV)*, 2015 IEEE, pages 1232–1239. IEEE, 2015.
- [9] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, and Wen Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. In *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on, pages 4363–4369. IEEE, 2013.
- [10] Panagiotis Lytrivis, George Thomaidis, and Angelos Amditis. Co-operative path prediction in vehicular environments. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 803–808. IEEE, 2008.
- [11] Adrian Broadhurst, Simon Baker, and Takeo Kanade. Monte Carlo road safety reasoning. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2005(412):319–324, 2005.
- [12] a. Eidehall and L. Petersson. Statistical Threat Assessment for General Road Scenes Using Monte Carlo Sampling. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):137–147, 2008.
- [13] Alain Lambert, Dominique Grayer, Guillaume Saint Pierre, and Alexandre Ndjeng Ndjeng. Collision probability assessment for speed control. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2008.
- [14] Christian Laugier, Igor E. Paromtchik, Mathias Perrollaz, Mao Yong, John David Yoder, Christopher Tay, Kamel Mekhnacha, and Amaury Negre. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. *IEEE Intelligent Transportation Systems Magazine*, 3(4):4–19, 2011.
- [15] Alejandro Ivan Morales Medina, Nathan van de Wouw, and Henk Nijmeijer. Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles. In *Intelligent Transportation Systems (ITSC)*, 2015 IEEE 18th International Conference on, pages 1696–1701. IEEE, 2015.