Analytic Collision Risk Calculation for Autonomous Vehicle Navigation

Andreas Philipp[1] Daniel Goehring[1]

***Abstract*— Collision checking and avoidance is an import part of the perception and planning system for autonomous driving. We present a new analytic approach to calculate the probability of a future collision and extend another already known solution to be suitable for ground vehicle navigation. Our new concept of the collision octagon facilitates in both cases the derivation of an analytic solution. Both approaches are compared to each other using simulated and real world scenarios. By comparing the results of the analytic solutions to the corresponding Monte Carlo simulations, their accuracy and real-time capability is demonstrated. The suitability of the analytic solutions for real world autonomous systems is further proven by integrating them into the trajectory prediction and planning system of the self-driving car of the Freie Universität Berlin.**

I. INTRODUCTION AND RELATED WORK

Estimating the risk of a future collision is essentially for autonomous driving [1] as well as for driver assistance systems [2]. In both cases, systems in the car have to perceive the environment, identify and track the relevant objects and to avoid collisions with them. Collision risk includes strictly speaking not only the probability of collision, but also the expected costs. In this paper we will concentrate on the probability estimation.

Collisions occur, when the ego vehicle and an obstacle are at the same position at some time in the future. In classical tracking applications, as air traffic control, objects are usually considered as points or circles. This is reasonable, since the regular distances between objects are very large compared to their extend due to the high velocities. In ground traffic situations, especially in urban environments, the distances between objects are much shorter and they have to be handled more realistically as extended objects. A common approach is to model traffic participants as oriented rectangles. Two rectangles are than in collision state if they overlap at least partially.

The relevant objects in the environment may be static or dynamic. Avoiding collisions with static objects is straight forward, despite the case of very noisy sensors. Dynamic objects are much harder to handle, since their movement in the future has to be predicted. Usually, the future states of the objects are predicted iteratively in steps of 0.1 or 0.2 seconds. With each step, the uncertainty of the state increases, depending on the selected motion model and the corresponding process noise. Since it is impossible to predict the point in time with the highest collision risk, the risk must be computed for every time step.

[1]DCLMR, Computer Science Institute, Freie Universität Berlin, Germany {andreas.philipp | daniel.goehring}@fu-berlin.de

For driver assistance systems, not only the obstacle trajectory is uncertain, but also the intention of the driver of the ego vehicle. The task is simplified in the case of autonomous driving, since the future trajectory of the ego vehicle is given by the planner of the same autonomous system and can be considered as deterministic. On the other hand, ADAS systems usually predict only a time span of 3 - 5 seconds, which is considered long enough for a driver warning or emergency brake. Autonomous cars have to predict longer time spans (5 - 10 seconds), since they are responsible not only for the safe, but also for a comfortable ride.

In many systems, instead of the collision risk, the time to collision (TTC) is estimated deterministically. Some similar measures as time to react (TTR) or time to brake (TTB) are also used [3] [2]. None of these measures consider the variances of the calculated time, nor the probability of colliding at all.

There are two different approaches for the probabilistic collision risk calculation:

- Collision state probability (CSP): probability of spatial overlap of two objects at a certain point in time [4] [5] [2]. All existing approaches use MC simulation to calculate the probability.

- Collision event probability (CEP) density: probability density of a collision event at a certain point in time [6] [7]. While [7] also uses MC simulation, [6] presents a analytic solution, but does not consider extended obstacles.

The CEP, which is the probability of a collision during a period of time, is calculated by integrating the CEP density over time.

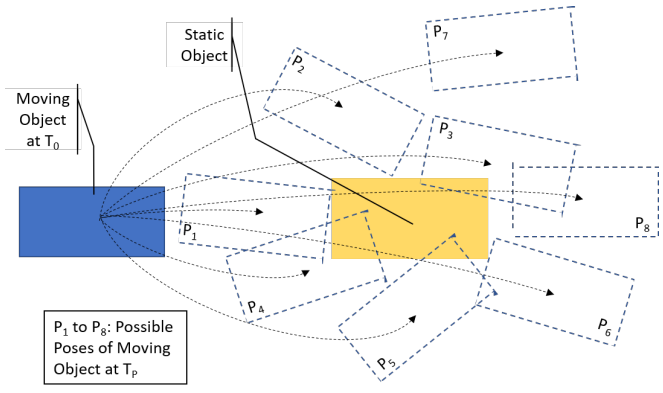

Fig. 1. Samples of predicted poses of moving object.

Fig.1 shows examples of collision states and events. In this case, a moving object approaches a static object. The state of the moving object at time T_0 is an uncertain estimate

provided by some tracking module. The prediction of future poses is further disturbed by process noise. The figure shows 8 samples of possible poses at prediction time T_p. Poses 1 and 2 represent collision events: The boundary of the moving object is just penetrating the boundary of the static object. Pose 6 is not collision event, since it is not a boundary crossing from outside to inside. Poses 1-6 represent collision states: the object rectangles overlap at least partially. Poses 7 and 8 are neither collision events or states.

Section II presents the general approach for CSP and CEP calculation. In Section III, possible solutions using Monte Carlo Simulations are shown. Analytic methods to compute CSP and CEP in real-time are presented in Section IV. Evaluation results are shown in Section V and conclusions and future work are summarized in section VI.

II. GENERAL SOLUTION

In the general case, the future 2D poses of the two objects are uncertain (see Fig. 2).

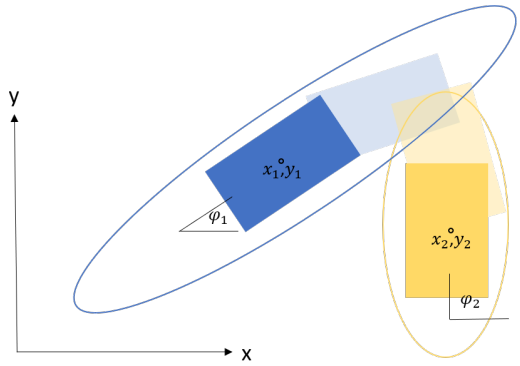

Fig. 2. Overlap uncertainty of 2 oriented rectangles at point in time.

To compute the collision risk, one has to solve the integral over their common state distribution (1). It is assumed, that the states of the two objects are independent of each other.

$$\int_{x_1}\int_{y_1}\int_{\varphi_1}\int_{x_2}\int_{y_2}\int_{\varphi_2} I_C(x_1, y_1, \varphi_1, x_2, y_2, \varphi_2) \\ p\,(x_1, y_1, \varphi_1)\;\; p\,(x_2, y_2, \varphi_2)\;\; dx_1 dy_1 d\varphi_1 dx_2 dy_2 d\varphi_2 \quad (1)$$

The indicator function (2) yields 1, if the two rectangles overlap at least partially. This indicator function is the reason, why the integral in the general case cannot be solved analytically.

$$I_C(x_1, y_1, \varphi_1, x_2, y_2, \varphi_2) = 1 \\ if\;\; S(x_1, y_1, \varphi_1) \cap S(x_2, y_2, \varphi_2) \neq 0 \quad (2)$$

Since the predicted states of an object at different points in time are not independent of each other, the collision risk cannot be accumulated over time to get the total collision risk over a time span [8]. Instead, the common distribution of the two objects over all time steps of the prediction horizon would have to be evaluated, which is of course intractable.

III. SOLUTION USING MONTE CARLO SIMULATION

CSP and CEP calculation can be achieved by Monte Carlo Simulation. These simulations usually not fulfill the time requirements of real driving solutions, but they are useful to provide ground truth for evaluation of better performing analytic solutions.

For the CSP calculation, the trajectories of both objects are predicted using some suitable process model. Both, the state and the covariances have to be forwarded. At each time step, a number of samples are taken from the state distributions of the both objects and checked, whether the rectangles overlap. The proportion of overlapping cases of all samples is the CSP at that point in time.

Calculation of the CEP requires drawing a number of complete trajectory samples. At simulation start, samples from the initial distribution are drawn to initialize each trajectory. At each time step, these trajectories are predicted forward using a suitable process model, and afterwards samples from the process noise distributions are drawn and added to the state. At each time step, the part of newly collided trajectories represent the CEP density. These trajectories are removed before applying the next prediction step. The proportion of collided trajectories over a time span is the CEP for that time span.

As the number of MC samples increases, the results converges to the true probability value due the strong law of large numbers [1]. The number of samples required for realistic results depends mainly on the number of probabilistic state variables. But it has also to be observed, that small risk values require more samples than higher ones [4].

IV. ANALYTIC SOLUTION

To achieve an analytic solution for autonomous driving, we propose two simplifications:

- the trajectory of the ego vehicle is considered deterministic. This follows from the fact, that the driver intention of the ego vehicle is given be the path planner. Additionally, the state of the ego vehicle is usually measured much more exactly and the uncertainty of prediction is negligible compared to the other traffic participants.

- the predicted trajectory of the obstacle normally follows the drive spline of the lane and therefore the variance of the orientation of the obstacle is very low. Moreover, the orientation of a non-holonomic system is not an independent random variable, but almost entirely determined by the velocity vector. On this account it is considered to be deterministic.

A. Collision octagon

Both types of risk calculation can be simplified by transforming the problem of collision between two oriented rectangles into the collision between a point and a collision octagon. The collision octagon is the result of the convolution of the two oriented rectangles (see Fig. 3).

The rectangle of the ego vehicle is replaced by the collision octagon, which depends on the centroid position of the ego vehicle and the orientation angle, the length and width of

both objects. The collision octagon is the trace of the obstacle centroid when the obstacle rectangle is moved around the border of the ego rectangle. The corners of the collision octagon can be computed by simple vector arithmetic.

The CSP at a certain point in time is than the probability, that the centroid of the obstacle is somewhere inside the octagon.

The CEP density is the probability density, that the obstacle centroid crosses one of the eight edges of the octagon from outside to inside.

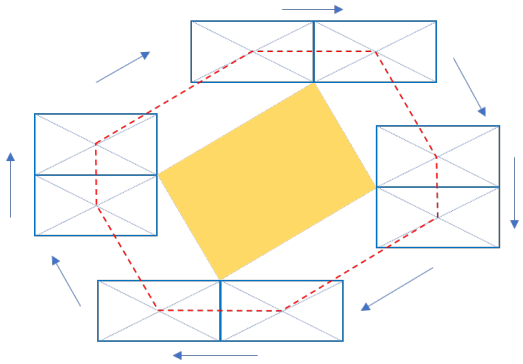

Fig. 3. Obstacle rectangle moving around the ego vehicle. In red the resulting collision octagon.

B. CSP Calculation

The obstacle is placed at the center of the coordinate system and its length is aligned with the x axis. Therefore the expected x and y - values are zero. In our system, we predict the lateral and longitudinal motion of the obstacle independently of each other, which results in a correlation coefficient of zero between x and y position. We assume a Gaussian distribution of the uncertain position of the obstacle centroid. The integral (1) simplifies to (3):

$$P(C,T_p) = \int_y \int_x I_C(x,y) \ \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} \ dxdy \tag{3}$$

This integral is the probability mass of the bivariate Gaussian integrated over the area of the octagon at time T_p. We split this into three integrals each for the upper and lower bounding edges of the octagon, the integrals for the two vertical edges are zero. Other solutions are possible. See Fig. 4 for the integration boundaries of the six integrals.

The integrals have a variable integration limit for Y, given by the straight line equation of the octagon edges (4).

$$P(C_i,T_p) = \int_{y=0}^{y=m_i x+b_i} \int_{x=x_{l,i}}^{x=x_{u,i}} \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} dxdy \tag{4}$$

Due to the variable boundaries of the outer integral, only the first part of the integration can be solved analytically.

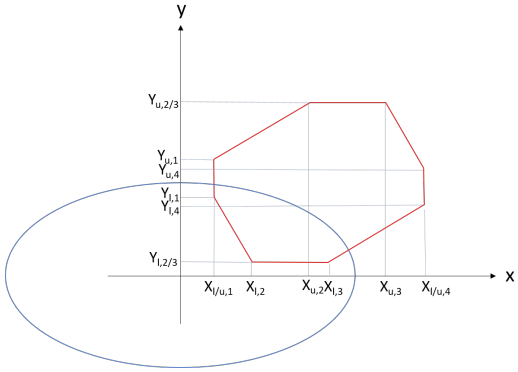

Fig. 4. Upper and lower integration boundaries. Collision octagon is shown in red, 1-Sigma ellipsis of Gaussian state distribution of obstacle in blue.

$$P(C_i,T_p) = \frac{1}{\sqrt{8\pi}\sigma_x} \int_{x=x_{l,i}}^{x=x_{u,i}} erf(\frac{m_i x+b_i}{\sqrt{2}\sigma_x}) e^{-\frac{1}{2}\frac{x^2}{\sigma_x^2}} \ dx \tag{5}$$

The remaining integral (5) over a smooth continuous function with constant integration limits can be evaluated very efficiently using numerical integration. The sum of the six integrals gives the CSP at time T_p (6).

$$P(C,T_p) = \sum_{i=0}^{i<3} P(C_i,T_p) - \sum_{i=3}^{i<6} P(C_i,T_p) \tag{6}$$

C. CEP Calculation

This calculation method has recently been presented in [6], where it is used to calculate the CEP density between a rectangle representing the ego vehicle and a point obstacle. We present here a short summary of the approach and extend it to the CEP between two rectangles using the collision octagon.

The approach is based on the probability rate for boundary crossings of stochastic vector processes [9] . The centroid of ego vehicle is placed in the origin of the coordinate system. Each edge of the octagon is treated as a boundary and the probability rate of the obstacle centroid penetrating this boundary is computed (see Fig. 5).

The integral (7) is derived in detail in [6]. The result is the probability density of crossing one edge of the collision octagon at time T_p.

$$\frac{dP(C_i,T_p)}{dt} = -p_{T_p}(x_0) \int_{\dot{x}\leq 0} \int_{y\in I_y} \dot{x} \ p_{T_p}(\dot{x},y|x_0) \ d\dot{x}dy \tag{7}$$

The integral depends on the probability of the obstacle being at the edge of the octagon $p(x_0)$ at time T_p, the relative x-velocity between the obstacles and the ego vehicle $\dot{x}$, and the common distribution of this velocity and the y-position of the obstacle conditioned on the x-position of the boundary $p(\dot{x},y|x_0)$. Integration limits are all negative velocities (directed to the inside of the octagon) and the y-limits of the octagon edge.

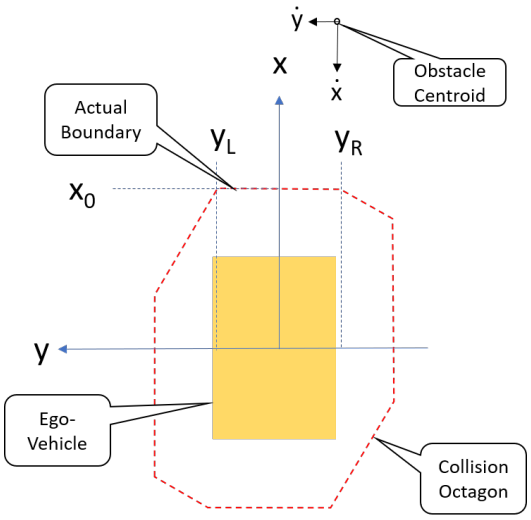

Fig. 5. Boundary crossing probability for one edge of the collision octagon.

This integral can be computed approximately by usage of a Taylor expansion for the off-diagonal element of the inverse covariance matrix. Evaluated to zeros order, this yields (8)

$$\frac{dP(C_i, Tp)}{dt} = -\mathcal{N}(x_0; \mu_x, \sigma_x) \\ (\mu_{\dot{x}|x_0}\Phi(\frac{-\mu_{\dot{x}|x_0}}{\tilde{\sigma}_{\dot{x}|x_0}}) - \tilde{\sigma}^2_{\dot{x}|x_0}\mathcal{N}(0; \mu_{\dot{x}|x_0}, \tilde{\sigma}_{\dot{x}|x_0})) \\ (\Phi(\frac{y_R - \mu_{y|x_0}}{\tilde{\sigma}_{y|x_0}}) - \Phi(\frac{y_L - \mu_{y|x_0}}{\tilde{\sigma}_{y|x_0}})) \quad (8)$$

$\mu_{\dot{x}|x_0}$ and $\mu_{y|x_0}$ are the expected relative x-velocity and y-position of the obstacle conditioned on the x-position of the boundary. $\tilde{\sigma}_{\dot{x}|x_0}$ and $\tilde{\sigma}_{y|x_0}$ are the corresponding standard deviations. $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.

This calculation may be refined by higher order terms of the Taylor expansion.

To get the total CEP density, we have to add the rates of all eight edges of the collision octagon (9). For this purpose, the coordinate system has to be rotated, so that the respective edges of the octagon become orthogonal to the x-axis.

$$\frac{P(C, T_p)}{dt} = \sum_{i=0}^{i<8} \frac{dP(C_i, T_p)}{dt} \quad (9)$$

The accumulation of the event rates over a time span T_1 to T_2 amounts to the CEP over that time span (10)

$$P(C, T_1, T_2) = \sum_{t=T_1}^{t<T_2} \Delta t \frac{dP(C, t)}{dt} \quad (10)$$

V. Evaluation

Evaluation of the collision risk algorithms is done in two parts. In the first part, we use a hypothetical scenario. We compare the two analytic algorithms with the corresponding MC implementations and show, that only a analytic solution is real-time capable. Furthermore, we contrast the CSP with the CEP and the reasons for their deviations.

In the second part, we present a real world example recorded with our MadeInGermany, where the algorithms are embedded into a complete tracking and prediction pipeline and the collision risks are computed for the planned trajectory of the ego vehicle.

A. Simulated Scenarios

We consider the following scenario: The position of the ego vehicle is in the coordinate origin, while the obstacle approaches the ego vehicle from diagonally in front. The prediction is done using a simple constant acceleration model.

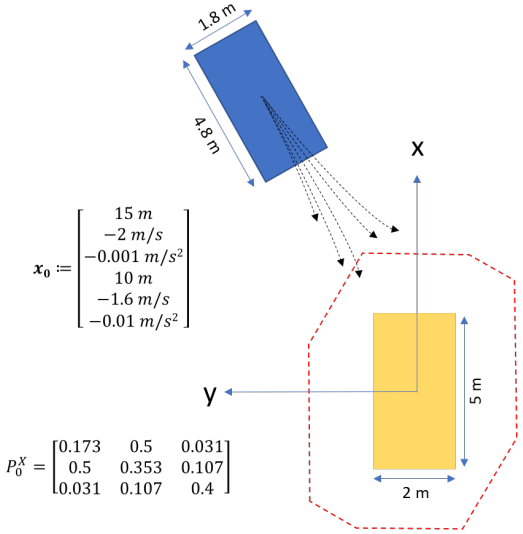

Fig. 6. Simulated collision scenario: initial obstacle state and random trajectories. Assumed process noise $q_x = q_y = 0.1\frac{m^2}{s^5}$. Initial state covariance matrix is shown for X dimension, Y variances are identically.

Fig. 7 shows the CEP density calculated by the analytic solution and the MC simulations with 1,000, 10,000, 100,000 and 1,000,000 runs. It has its peek at 3.5 seconds with a density of $\approx$ 0.01/seconds. While the Monte Carlo simulations with 1,000 and 10,000 samples show significant deviations from the analytic solution, the MCS results for 100,000 and 1,000,000 samples converge strongly. The MC simulation with 1,000,000 samples can be taken as ground truth and the good coincidence with the analytic results shows, that the Taylor expansion of the CEO calculation is sufficiently accurate.

Fig. 8 shows the CEP, which is the accumulated event rate, over time. After 6 seconds, it has almost reached its saturation level of $\approx$ 22 %. This relative low risk results from the high variances in all directions, which makes it probable, that the obstacle passes the ego vehicle on the left or right side. Again, MCS results for 100,000 and 1,000,000 runs are very close.

Fig. 9 shows the CSP over time. It reaches its peak at $\approx$ 4.4 seconds and than decreases again. The reason for this

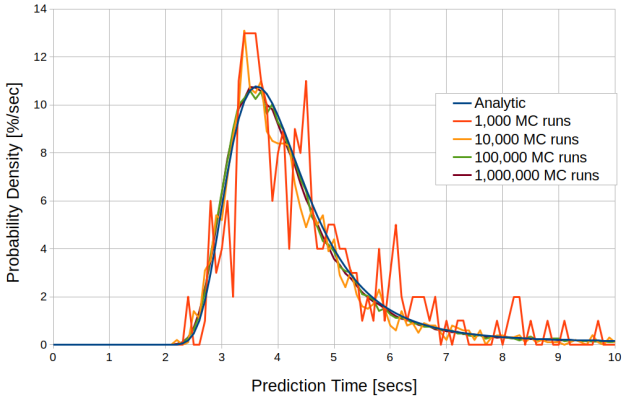

Fig. 7. CEP density over time.

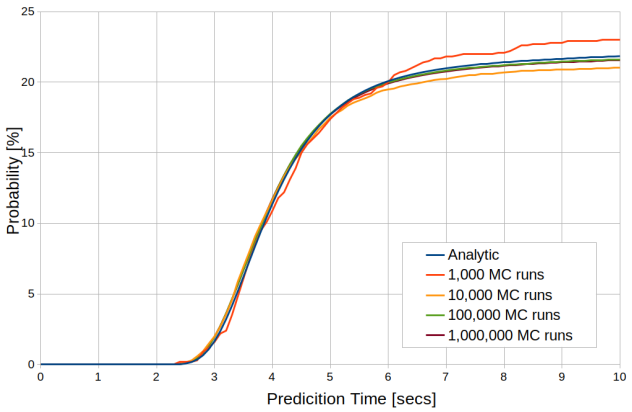

Fig. 8. CEP over time.

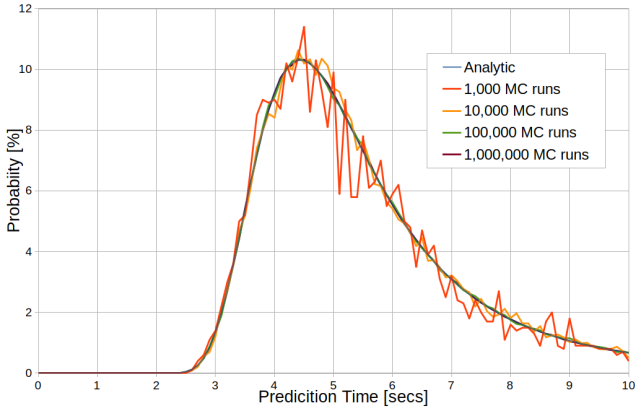

Fig. 9. CSP over time.

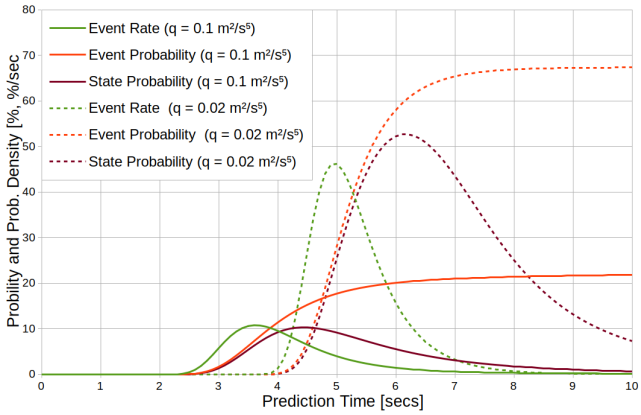

Fig. 10. CSP, CEP and CEP density over time for the same scenario with different process noise rates.

decrease of the probability is that states of the obstacle behind the ego vehicle (at $x \ll 0$) are not collision states, even though the obstacle must have past through the ego vehicle before. Again, MCS results for 100,000 and 1,000,000 runs are satisfactory. The good coincidence between the results of 1,000,000 MC runs and the analytic solution proves the sufficient accuracy of the numerical integration part in the CSP calculation.

Fig. 10 (solid lines) shows the analytic state probability, event probability and event rate in one diagram. The state probability reaches only about the half of the event probability, before it decreases again. This is due to the high variances in the initial state and the added process noise. The dotted lines show the results for the same scenario, but with reduced initial variances and process noise. In this case, the resulting collision probability is much higher and the deviation between the maximal CSP and CEP is strongly reduced.

B. Real World Scenario

For real world scenarios, the collision risk calculation was integrated with other modules of the self-driving car solution MadeInGermany [10]. The system is based on the ROS operating system [11] and uses as principal tracking device a Velodyne HDL-64 LIDAR. The perception pipeline implements various types of Kalman Filters [12] and combines them using a IMM filter [13] for obstacle tracking. Based on the tracking results, obstacle maneuvers are detected using a Bayesian network [8] [14] and trajectories are predicted based on Gaussian processes [8] [15]. The predicted trajectories of the obstacles are compared to the planned trajectory of the MadeInGermany [16] [17], which can be considered as deterministic thanks to the controller architecture of the car [18].

Fig. 11 shows a real world scenario recorded during a test drive with the MadeInGermany self-driving car in front of the Freie Universität in the Thielallee in Berlin-Dahlem. The planner of ego-vehicle (no.1) prepares a left turn, while from the right side approaches an obstacle (no. 38), which has priority. The Bayesian network sees a follow road maneuver as most probable for the obstacle. The Gaussian trajectory prediction module for the follow road maneuver assumes, that vehicle n.32 tries to keep the middle of its lane with low variance and will continue in this case with nearly constant velocity.

Fig. 12 shows the same scenario as seen by the front mounted fish eye camera of the MadeInGermany.

Fig. 13 show the resulting CEP density, CEP and CSP. The event rate has a sharp peak of 250 % per second at about

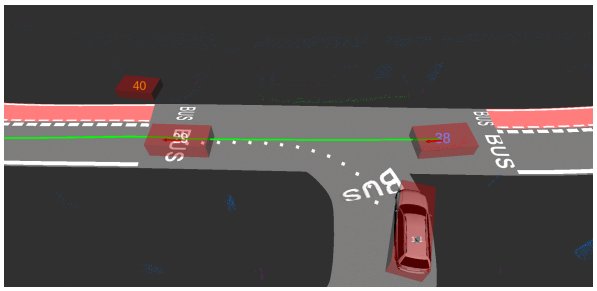

Fig. 11. Real world collision scenario: planned trajectory of ego-vehicle (white dotted line) and predicted trajectory of obstacle (green line).

Fig. 12. Front camera fish eye view of collision scenario.

1.7 seconds. The CEP increases very fast to more than 90%, as does the CSP.

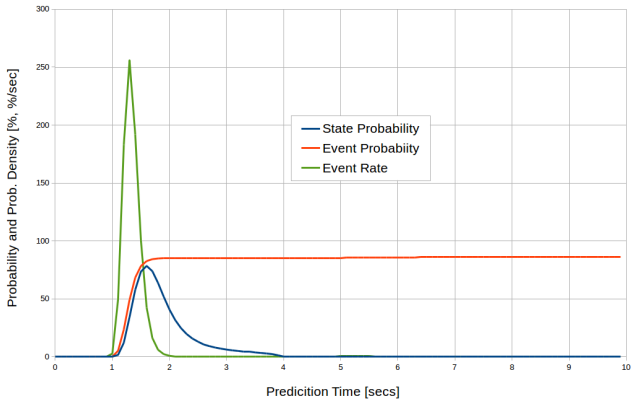

Fig. 13. CSP, CEP and CEP density over time (real world scenario).

C. Timing Evaluation

Using the simulated scenario, we have measured the following performance values using a Intel i7-8750H (see Table I). All figures are for the calculation of a 10 seconds predicted trajectory (100 positions). The Monte Carlo simulations include 10,000 runs.

Evaluation results (mSecs)			
	MCS	Analytic	Factor
CSP	95,6 ms	0,9 ms	106
CEP	266,3 ms	0,3 ms	887

TABLE I
RESULT OF TIMING EVALUATION.

VI. CONCLUSIONS AND FUTURE WORK

We have presented analytic solutions for computation of the CSP and CEP. Both solutions have shown to provide numerically accurate results by comparing them to large scale Monte Carlo simulations. Their performance is about 100 - 800 times better than the MC simulations. The analytic calculation for oriented rectangles is supported by the concept of the collision octagon.

Comparing both approaches, we come to the following results:

- CSP calculation needs only position data and variance as input, while CEP needs position, velocity and the corresponding covariance matrix
- CSP may be calculated for a single trajectory point, while CEP is only meaningful, when evaluation whole trajectories
- CEP calculation results in a probability density function and a cumulative distribution function. CSP calculates probability masses at discrete points in time, but these form no valid probability mass function, since they are not independent of each other
- If variances are low, the maximal CSP probability is close to the cumulated event probability
- CEP calculation is about 3 times faster than CSP

Both solutions have successfully been integrated into the self-driving car system MadeInGermany of the Freie Universität Berlin. The differences between the two probability approaches are low when applied to a real world driving situation due to the lower variances.

This work has shown, that it is feasible to calculate the risks emerging from the various possible maneuvers of all traffic participants in the scene in real-time. This provides not only valuable information for the planner of the autonomous vehicle, but in the future may also be fed back into the perception pipeline to provide some kind of risk based attention control. Upcoming devices, such as solid state LIDARs provide means to control the scanning resolution and scanner frequency of specific regions interest. This can be leveraged in combination with the collision probabilities of the relevant obstacles to implement attention control for enhanced perception efficiency.

Further future work on this topic will try to evaluate, whether the analysis of the temporal and spacial distribution of the CEP density can be used as additional input to planning modules. Especially the temporal gradient of the distribution could be interpreted as a recommendation to decelerate or accelerate the ego vehicle to reduce the collision risk.

REFERENCES

[1] S. Annell, A. Gratner, and L. Svensson, "Probabilistic collision estimation system for autonomous vehicles," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 473–478.

[2] A. Houénou, P. Bonnifait, and V. Cherfaoui, "Risk assessment for collision avoidance systems," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 386–391.

[3] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[4] A. Lambert, D. Gruyer, and G. Saint Pierre, "A fast monte carlo algorithm for collision probability estimation," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*. IEEE, 2008, pp. 406–411.

[5] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.

[6] R. Altendorfer and C. Wilkmann, "What is the collision probability and how to compute it," *arXiv preprint arXiv:1711.07060*, 2017.

[7] P.-J. Nordlund and F. Gustafsson, "Probabilistic conflict detection for piecewise straight paths," 2008.

[8] M. Schreier, "Bayesian environment representation, prediction, and criticality assessment for driver assistance systems," Ph.D. dissertation, Technischen Universitaet Darmstadt, 2015.

[9] Y. K. Belyaev, "On the number of exits across the boundary of a region by a vector stochastic process," *Theory of Probability & Its Applications*, vol. 13, no. 2, pp. 320–324, 1968.

[10] Wikipedia. (2011) Madeingermany. [Online]. Available: https://de.wikipedia.org/wiki/MadeInGermany

[11] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[14] M. J. Druzdzel, "Smile: Structural modeling, inference, and learning engine and genie: a development environment for graphical decision-theoretic models," in *Aaai/Iaai*, 1999, pp. 902–903.

[15] T. Christopher, "Analysis of dynamic scenes: application to driving assistance," Ph.D. dissertation, Institut National Polytechnique de Grenoble-INPG, 2009.

[16] F. Ulbrich, D. Goehring, T. Langner, Z. Boroujeni, and R. Rojas, "Stable timed elastic bands with loose ends," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 186–192.

[17] Z. Boroujeni, D. Goehring, F. Ulbrich, D. Neumann, and R. Rojas, "Flexible unit a-star trajectory planning for autonomous vehicles on structured road maps," in *Vehicular Electronics and Safety (ICVES), 2017 IEEE International Conference on*. IEEE, 2017, pp. 7–12.

[18] D. Göhring, "Controller architecture for the autonomous cars: Madeingermany and e-instein," *Technical report, Freie Universitat Berlin*, 2012.