Monte Carlo Sampling Based Imminent Collision Detection Algorithm

Peng Chang†
Princeton Vision
Potomac, USA
pchang@princetonvision.com

Christoph Mertz
Robotics Institute,
Carnegie Mellon University
Pittsburgh, USA

***Abstract* — Imminent collision detection is an important problem for automotive safety, and is critical for driving assistance systems and fully autonomous vehicles. Imminent collision detection systems require very low false alarm rate, due to the potential outcome of the detection result. Most current approaches are based on Kalman filter or its variants and tend to have degraded accuracy when the underlying noise models deviate from the Gaussian assumption. In this paper, we present a Monte Carlo sampling based Imminent Collision Detection algorithm (MCICD) to achieve improved accuracy by faithfully modeling the noise distributions of the sensor measurements. We further demonstrate a Monte Carlo sampling framework to perform the FPR/FNR analysis for any given collision detection system, as the criterion to evaluate the performance of different collision detection approaches. Experiments with synthetic data and a laser scanner based prototype system have been conducted to validate our approach.**

***Index Terms*—collision avoidance, Monte Carlo method, noise distribution, sampling method**

I. INTRODUCTION

IMMINENT collision detection (ICD) is an integral component of automotive active safety systems. Imminent collisions are collisions which will occur within less than a human driver's reaction time. Once they are detected, imminent collisions can potentially be mitigated by active safety measures. Therefore there has been enormous interest for a reliable imminent collision detection system in the automotive industry. Nonetheless, due to the potential liability associated with false alarms, the requirement for the accuracy of any potential ICD product is very stringent. In this paper, we present a novel methodology to improve the ICD accuracy by faithfully modeling the sensor measurement noise, and incorporating human behavior models if needed. We also demonstrate that the Monte Carlo sampling framework can be applied to compute the FPR/FNR of any given ICD prototype system, as the criterion to compare ICD systems with different sensor suites and approaches.

Most collision detection approaches in the literature assume Gaussian noise in the sensor measurements, and use Kalman filter or its variants to integrate the sensor measurements through time, and perform the collision detection based on the filter output [1], [3], [11], [21], [23]. In practice, the sensor measurement noise often deviates from the Gaussian assumption. It is also not uncommon to find that certain noise distributions in a prototype collision detection system are scenario-dependent.

In the literature, most recent work on collision detection systems have been focused on target detection, tracking and motion measurement via specific sensor suites [4], [5], [8], [18], [22], [25], and not much attention has been given on the measurement noise modeling.

In this paper, we propose to faithfully model the uncertainty of the sensor measurements, and then compute the collision probability according to the realistic uncertainty models as a means to achieve better false alarm rate. We further propose a Monte Carlo sampling framework to compute the overall FPR/FNR for any given ICD prototype, as a criterion to compare different ICD approaches, and guide the selection and development of ICD sensor suites.

A. Related Work

Various collision detection prototype systems have been proposed with different sensor suites, such as the monocular camera, stereo vision, radar, laser scanner and their combinations [3], [5], [12], [16], [17],[23][12]. These existing approaches tend to focus on how to extract the target information from the sensor data, and there has been a lack of study on the noise characteristics of the measurements and how the noise characteristics affect the FPR/FNR of the final system, which is carefully studied in this paper. Experimental results have been obtained with a laser scanner based prototype system.

Crash probability analysis has been studied in the literature [11], [14], and Monte Carlo method has been applied to driver safety analysis [15]. We believe our approach is the first to (1) faithfully model the realistic uncertainties in sensor measurements, (2) incorporate the realistic uncertainty models into the proposed Monte Carlo sampling framework, and (3) demonstrate the effectiveness on a real prototype ICD system.

There is a line of research in the robotics community on Inevitable Collision States (ICS), which is aimed to determine safe paths for robot motion planning. In 2004 ICS was firstly defined and formalized in [7], and later probabilistic approaches were proposed to calculate the risk of collision based on ICS as in [1], [2], [20], with the uncertainty in measurements all assumed to be Gaussian. In [10] an algorithm was proposed to detect ICS by exhaustively checking for all possible maneuvers, and in [6] a Monte Carlo simulation based approach was proposed to find the ICS by searching a subsample set of all the possible maneuvers. Our ICD detection

This work was supported in part by DARPA STTR contract number W91CRB 10C0103.

† Peng Chang is also with Ke Li Corporation.

algorithm differs from ICS in that the purpose of ICD algorithm is to faithfully report the collision probability based on the current observed system states and input. The ICD collision probability can be used to trigger or pre-arm active safety devices, or be used as input to the automatic maneuver control module if it exists to generate maneuvers to mitigate the collision danger if any.

II. System Overview

A. Hardware Setup

MCICD algorithm is independent of particular sensor suite selection. In this paper, the primary sensor for detecting and tracking vehicles is a SICK™ LMS laser scanner. In addition to the distance measurements from the laser scanner we also need the state (velocity, turn rate, etc.) of the host vehicle. The speed of the host vehicle is read from the OBDII and the rotation and acceleration are measured with a 6 DOF MEMS IMU. A GPS device gives the absolute position and a Flea2™ captures color video. Absolute position and video is not needed for detection and tracking, it is used for visualization and analysis. All this data is passed over RS232, RS422, or firewire to a JREX-plus-690™ computer as in Fig. 1.

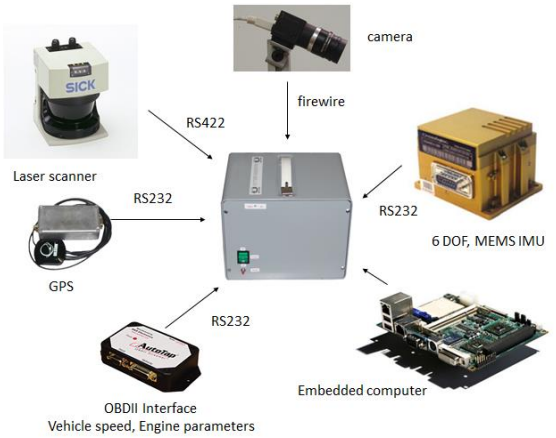


Fig. 1 Hardware components connected to the computer

The laser scanner is mounted on the front of and the GPS antenna on top of the host vehicle. The camera is behind the windshield looking forward and the rest of the hardware is packed into an enclosure.

III. Main Algorithm

A. Geometric Representation

Previously point based models have been used in most radar based systems, partly because it is difficult to reason about the size of the threat object from the automotive radar return. A point based model is adequate for certain domains such as the aerial navigation safety or applications such as collision warning systems, where the size of the threat is secondary to the safety distance required between vehicles. However the point model is *not* sufficient in imminent collision detection, where the size of the buffer zone is comparable to the size of the vehicles. In our work, the shape of the vehicle is modeled as a rectangle, although other shape models can be applied too.

Fig. 2 shows a 2D rectangular model used in our collision detection algorithm, which is a close approximation to the shape of the ground projection of most automobiles. All the results and algorithms developed with the rectangular representation can be extended to other geometric models without inherent difficulty.

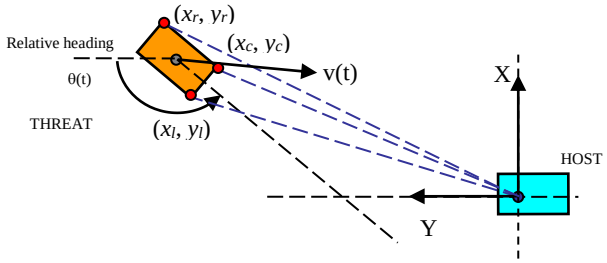


Fig. 2 Rectangle based model for collision geometry

The geometry of the host vehicle is usually known, such as the width and length of the vehicle, and the location of the sensor suite, which is used to transform the sensor measurements into a common coordinate system. The geometry of the threat vehicle is estimated up to the accuracy of the sensor data and processing. In Fig. 2, the geometry of the threat vehicle is represented by the left-most, right-most and the closest point, all visible from the laser scanner, as indicated as = (x_l , y_l), (x_r , y_r) and (x_c , y_c).

B. Dynamics Model

A vehicle dynamics model is utilized to predict the trajectory of the threat and host vehicle [13]. The accuracy of the dynamics model and the motion parameter estimate plays a critical role in the accuracy of the collision detection system. The chosen dynamics model in the proposed approach includes the following parameters: the velocity, acceleration, heading angle and yaw rate, indicated as $\{v(t), a(t), \theta(t), \dot{\theta}(t)\}$as in Fig. 2. It covers or closely approximates most vehicular motion, while still remains observable from the host sensor suite.

C. State Definition

The combination of the rectangular model and the chosen dynamics model suggests that the estimated state of the threat vehicle includes the following variables as in Equation (1),

$$\boldsymbol{X}(t)=\{ x_l, y_l, x_r, y_r, x_c, y_c, v, a, \theta , \dot{\theta} \} \quad (1)$$

where $\{x_l, y_l, x_r, y_r\}$ are the left-most and right-most boundaries of the threat vehicle, and $\{x_c , y_c\}$ the closest point on the threat vehicle to the host, all observable from the host sensor suite, as illustrated in Fig. 2, in addition to the speed, acceleration, heading angle, and yaw rate. Furthermore we assume the host vehicle motion is perfectly known from the vehicle ECU (electronic control unit).

D. Data Preprocessing

The laser scanner gives an ordered set of points that represent a cross section of the scene in front of the host vehicle. We also have the state of the host vehicle (velocity etc.). The continuous stream of this data is used to detect and track moving objects (DATMO). A detailed description of this system can be found

in [16], and here we will give a short overview.

The flowchart of the DATMO algorithm is shown in Fig. 3. The raw scan line from the laser scanner is transformed into the world coordinate system. There the points of the scan line are clustered with a region growing algorithm into segments. Lines and corners are fit to the segments and the resulting edges and corners are the feature points. A segment that overlaps with an object from the current object list is associated with it. The feature points of the segment are used to update the Kalman filter of the associated object. If a segment cannot be associated, a new object is created. Objects that have not been updated for a while will be deleted. DATMO is also taking care of special cases like occlusions, partial occlusions, splits, merges, etc. The updated list of objects and their kinematics is then passed to other software modules.

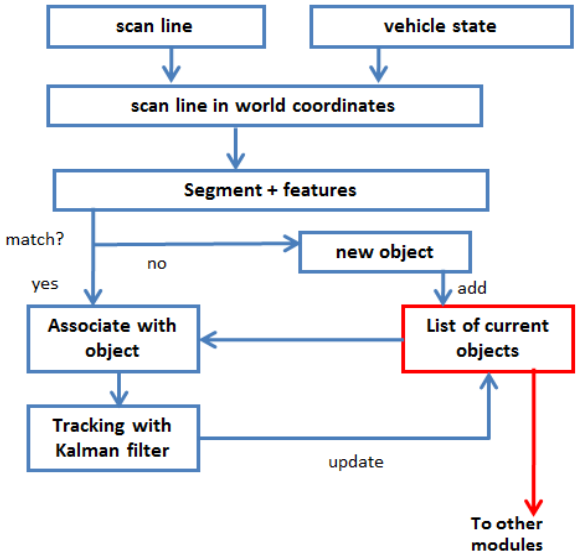


Fig. 3 Flow chart of preprocessing algorithm

The specific state vector that is calculated for this project by DATMO is shown in Equation (1). Speed, acceleration, heading angle, and yaw rate are calculated from the output of the Kalman filter. The left-most and right-most boundaries of the threat vehicle are directly calculated from the point cloud of the segment. The Kalman filter also gives the covariance associated with the dynamic variables. In Section E we will discuss how these play a role in the noise modeling.

E. Noise Modeling

We built the realistic noise models of the sensor measurements from both real data and simulated data.

1) Methodology

To investigate the noise model of a system one needs to collect large amount of measured data and compare it to ground truth data. Collecting large amount of instrumented data in real world consumes time and effort. A practical method is to use simulations. We developed a simulator that provides raw sensor data for a given input scenario. The raw data is then analyzed and the resulting measured state vector is compared with the ground truth, e.g. the input. Fig. **4** shows a sample scenario and raw laser scanner data from a simulation.

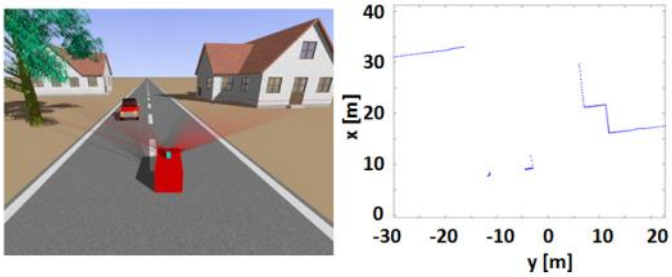


Fig. 4 Left: Snapshot of a simulation scenario. The host vehicle is marked as a red block. Right: The scan line measured by the simulated laser scanner.

We simulated a total of 2180 runs. The variations in the state vector of the host and threat vehicles are listed in Table 1. In the first set of simulations the turn rates are zero and in the second the accelerations are zero.

In the following we show the error models of the sensor measurements obtained from the combination of both real data and simulated data.

Table 1 Variations in the host and threat state sensor in the simulations.

		first set		second set	
		min	max	Min	max
Threat	x [m]	-3	3	-80	80
	y [m]	40	40	12	83
	v [m/s]	6.7	26.7	6.7	26.7
	a [m/s^2]	-2.45	1.47	0	0
	θ [rad]	$-\pi/2$	$-\pi/2$	0	π
	$\dot{\theta}$ [rad/s]	0	0	-2.2	2.2
Host	v [m/s]	6.7	26.7	6.7	26.7
	a [m/s^2]	-2.45	1.47	0	0
	$\dot{\theta}$ [rad/s]	0	0	-1.8	1.8

2) Position error

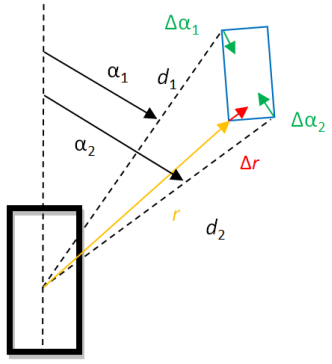


Fig. 5 Definition of radial and tangential errors.

According to the spec of the laser scanner and with the configuration we use for experiments, the position accuracy of the laser scanner for a single point is Δr=0.01m radically and Δt=0.0087r tangentially. The position error models we are interested in are the left-most, right-most and the closest point of the threat vehicle. It is observed that the errors are mostly caused by missed points and the laser beam hitting the target at different spots, e.g. the grill instead of the front bumper, which makes the distributions asymmetric in general.

In the next sections we discuss the radial error and the tangential error for the three points, namely (Δr, Δt), (Δd_1, Δt_1) and (Δd_2, Δt_2), where the tangential errors $\Delta t_1 = \Delta\alpha_1\, d_1$ and $\Delta t_2 = \Delta\alpha_2\, d_2$ (see Fig. 5 for the definitions.)

a) Position error of the closest point

The error distribution of the measured tangential distance to the closest point (Δr) is Gaussian. Fig. **6** shows that the distribution of the radial distance has two peaks and is skewed towards longer measured distances, with a parametric form of $0.082\boldsymbol{N}(x|0,0.034)+0.018\boldsymbol{N}(x|0.27,0.019)$ and $\boldsymbol{N}$ is the normal distribution. A possible explanation for this shape is that most times we see the front bumper of the target, but sometimes we see the grill (second, smaller peak). We never see anything in front of the front bumper and therefore the distribution is skewed.

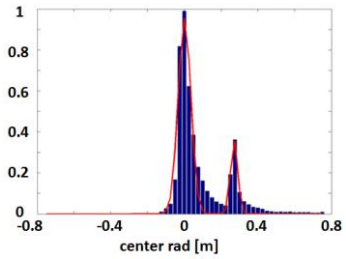


Fig. 6 Error distribution of the radial distance (Δr) to the closest point.

b) Position error of the leftmost/rightmost point

Fig. **7**(left) shows the error distributions of the measured radical distance to the rightmost point. The distribution of the radical distance can be modeled as $0.101\boldsymbol{N}(x|0.259, 0.04)+0.067\boldsymbol{N}(x|-3.281,0.051)+0.05C(-1,0.1)$. This distribution has a mode located at -3.281 meter, which is caused by confusion of the rightmost point with the right-front point when the angle LOC is small. It happens more often when the threat vehicle is far away from the host vehicle.

Fig. **7**(right) shows the distributions of the measured errors of the tangential distances to the leftmost and rightmost points. The left tangential distance Δt_1 can be modeled as $\boldsymbol{N}(x|0.11,0.06)$. The distribution is off-centered and has a tail to one side. As we mentioned above, the distribution is skewed because there are almost never any spurious returns outside the target, but there can be missed points, and α_1 is biased towards the center of the target vehicle. Similarly, the distribution of the right tangential distance Δt_2 is also skewed, as shown in the figure.

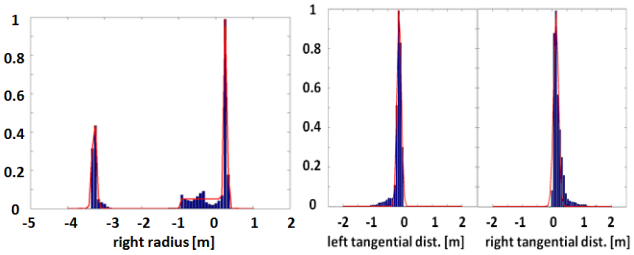


Fig. 7 Error distributions of the radical distance (left) and the tangential distances (right).

3) Speed error

Fig. **8** (top-left) shows the distribution of the measured speed difference after the initialization phase has passed. A Gaussian is fitted to it and one can see that Gaussian $\boldsymbol{N}(0,0.27)$ describes the distribution quite well. Fig. **8** (top-right) shows the relationship of the measured error to the error that is estimated by the Kalman filter of DATMO. One can see that they track each other fairly well, and the error from the Kalman filter is therefore a good estimate.

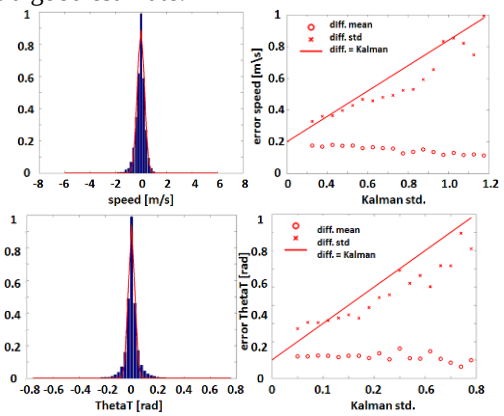


Fig. 8 (Top) Error distribution of speed measurement. (Bottom) Error distribution of heading measurement.

4) Orientation (or heading) error

As with the speed, the Gaussian $\boldsymbol{N}(0,0.023)$ describes the distribution fairly well and the error from the Kalman filter is a good estimate, as shown in Fig. 8 (bottom).

5) Acceleration error

We found that the measured error was only a little better than the standard deviation distribution of the acceleration itself, as shown in Fig. 9. This means that estimating the acceleration to be zero gives a similar error than measuring it.

It is no surprise that it is difficult to measure the acceleration since it is the time derivative of the velocity. It might be possible to measure the acceleration more reliably in cases where the time horizon is longer than the average of 2 seconds we have in our situations.

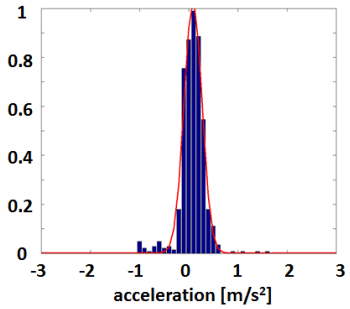


Fig. 9 Acceleration

6) Yaw rate error

As with the acceleration, we found that the measured error was only a little better than the standard distribution of the change in orientation (turn rate) itself. This means that estimating the change in orientation to be zero gives a similar error than measuring it.

F. Collision Detection

Given the estimated state X at time t = 0, the collision detection algorithm *predicts* if a collision will occur within the next T seconds and outputs a binary decision $\delta(X, T)$, where T is

referred as the *look-ahead time*.

To compute $\delta(X, T)$, we first calculate the following time varying binary function $F(X, t)$ for $0< t <\boldsymbol{T}$:

$F(X, t)=1$,if there is a collision, otherwise, $F(X, t)=0$

Then, $\delta(\boldsymbol{X},T)= \delta(\int_{t=(0,T)} F(X, t))$

A collision at any time instance t is defined as the threat object and the host vehicle intersect. With the rectangular representation of the vehicles, the intersection is detected by checking whether the two rectangles overlap, given the predicted state $X(t)$, as follows.

1. Propagate $X(0)$ into $X(t)$, for any given $0< t <\boldsymbol{T}$,
2. Perform the required geometric reasoning to check the overlap given $X(t)$.
3. Return 1 if there is an overlap; otherwise return 0.

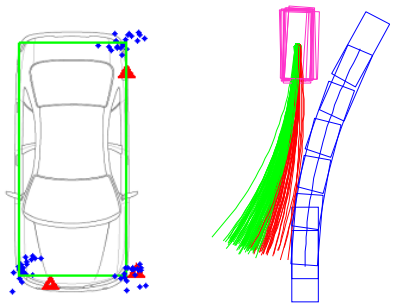
G. Main Algorithm

1) Sample based representation for noise distribution $\boldsymbol{N_X}$ and state X

Sampling technique is a general method to faithfully represent any distribution, even in high dimensional space [9]. It has proven to be a very effective technique to solve challenging problems involving unknown and non-Gaussian noise distributions. For any given distribution $f(\boldsymbol{X})$, where X is a random vector variable, and the exact form of $f(\boldsymbol{X})$ is unknown, the distribution can be effectively represented by a set of samples properly drawn from $f(\boldsymbol{X})$, noted as $\{\boldsymbol{X}^S, S=1\ldots K\}$, where K is the sample size. An important property of sample based representation is that the accuracy of the estimate is independent of the dimensionality of the space sampled from (P176 of [9]). This property guarantees that the number of independent samples need not grow exponentially with the dimension of the state. In our case, the random state vector X has a dimensionality of 8, and the sampling technique is probably the only viable method to represent this distribution faithfully without making strong assumptions.

2) Sampled representation from the noise models

Realistic sensor measurement noise model is obtained for the laser based sensor suite, and sample based representation is obtained for each measurement, through a sampling process. We use a sample size of 200 for all the sampled distribution.



(a) (b)

Fig. 10 Sampled representation of the location error (Note, the scales of the axis are different, for better viewing)

a) Location error

To obtain a sampled representation of the location error, we first translate the error in the polar coordinates into the usual Cartesian, then perform the sampling from the given distribution. An example of the sampled uncertainty is given in Fig. 10.

3) Monte-Carlo ICD algorithm

The MCICD algorithm, with predictable FAR/FNR is described in the following bounding box, where P_T is a decision threshold that is within [0, 1]. Note that $P(\delta(\boldsymbol{X},T)=1)$ and $P(\delta(\boldsymbol{X},T)=0)$ can be estimated by counting the corresponding samples.

MCICD Algorithm

1. Given the measured threat vehicle state $\boldsymbol{X}$, generate the sampled distribution $\boldsymbol{X}^i$
2. For i=1…N, evaluate $\delta(\boldsymbol{X}^i,T)$ as described in Section *F*
3. Estimate $P(\delta(\boldsymbol{X},T)=1)$ and $P(\delta(\boldsymbol{X},T)=0)$ by counting the corresponding samples.
4. Set $\delta(\boldsymbol{X},T)=1$, if $P(\delta(\boldsymbol{X},T)=1)>P_T$. Otherwise, set $\delta(\boldsymbol{X},T)=0$

4) Uncertainty Propagation through Time

In Section 3), the Monte-Carlo ICD algorithm is described based on a sampling based representation of the vehicle state measurements at a given time instance. In practice, the ICD system will continuously make measurements of the vehicle states, and the ICD algorithm be performed at each time instance. A particle filtering process is needed to propagate the measurements and the uncertainty through time, to achieve the improved accuracy [19].

Let $\boldsymbol{X_t}$ be the estimated vehicle state at time t, $p(\mathbf{X_t}|\overrightarrow{\mathbf{Z_t}})$ the probabilistic distribution of $\boldsymbol{X_t}$ based on observations $\overrightarrow{\mathbf{Z_t}}$ up to time t. Given the system dynamics $p(\boldsymbol{X_{t+1}}|\boldsymbol{X_t})$ and the observation $\boldsymbol{Z}_{t+1}$, we need to calculate $p(\mathbf{X_{t+1}}|\overrightarrow{\mathbf{Z_{t+1}}})$, the state distribution at time $t+1$. Note that according to our ICD algorithm, the probabilistic distribution of the state uncertainty is represented with sample set.

The following particle filtering algorithm propagates the sample set $\{X_t^i, i = 1 \ldots N\} \propto p(\mathbf{X_t}|\overrightarrow{\mathbf{Z_t}})$ to generate $\{X_{t+1}^i, i = 1 \ldots N\} \propto p(\mathbf{X_{t+1}}|\overrightarrow{\mathbf{Z_{t+1}}})$

Step 1 of the algorithm is basically to extrapolate the current vehicle into the next time frame, according to the dynamics model we use. In practice, we deploy the Runge-Kutta 4th order integration to perform the extrapolation. Since we only need to predict the state $\boldsymbol{X_{t+1}}$ at $t+1$, the computation in the prediction step is mild.

Step 2 of the algorithm is simply to sample the same noise distribution at time $t+1$.

Step 3 of the algorithm mixes the two sample sets of the predicted vehicle state and the measured vehicle state, and draws the new sample set from the mixture.

In the particle filtering process described above, the sample size N and M are chosen to reflect the respective confidence of the predicted state and the measured state at time $t+1$. By

default *M* and *N* can be set equal.

Particle Filtering Based Uncertainty Propagation

1 For i=1…N, for each sample X_t^i, generate the prediction set $\{X_{t+1}^{i,1}, i = 1 \dots N\}$ according to the vehicle dynamics and unicycle model.

2 At time *t+1*, obtain the direct state measurement set $\{X_{t+1}^{i,2}, i = 1 \dots M\}$, according to $p(\boldsymbol{X}_{t+1} \mid \boldsymbol{Z}_{t+1})$

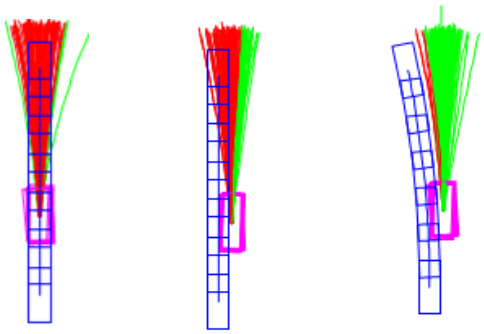
3 Random sample from the joint set $\{X_{t+1}^{i,1}, i = 1 \dots N, X_{t+1}^{i,2}, i = 1 \dots M\}$, to generate the sample set $\{X_{t+1}^{i}, i = 1 \dots N\} \propto p(\mathbf{X}_{t+1} | \overrightarrow{\mathbf{Z}_{t+1}})$.

IV. Experimental Results

A. Simulation Results

Given the algorithm, the corresponding FAR/FNR can be computed by testing the algorithm on the given set of scenarios with ground truth. In this section, we first show the results on several representative scenarios, followed by the FAR/FNR results on a simulated testing set.

1) Representative Scenarios

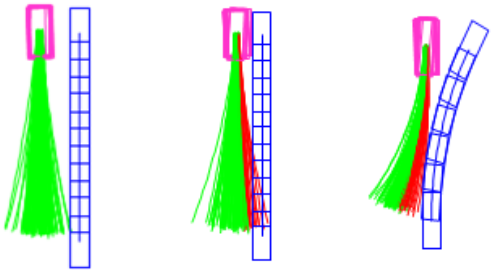


(a) (b) (c)

Fig. 11 Rear-end collisions

Fig. **11** shows several examples of rear-end collisions and near misses. The blue rectangles represent the host vehicle trajectory, which is predicted from the known host motion parameters. The cyan rectangles represent the samples of the estimated initial position of the threat vehicle, according to the sensor measurements and error distributions. The red and green curves are the predicted path samples of the threat vehicle according to the sensor measurements and noise models. A predicted path sample is colored as red if a collision event is detected; otherwise the path sample is colored as green.

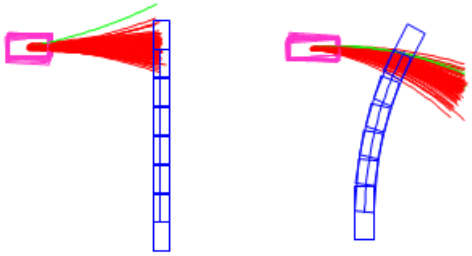
Fig. **11**(a) shows a collision scenario caused by the front car decelerating, and Fig. 11(b) shows a similar collision scenario with a lateral offset between the host and threat vehicles. Fig. **11**(c) shows a near miss scenario, where the front car is decelerating, and the host vehicle makes a left turn. In all these scenarios, collisions and misses are correctly detected.



(a) (b) (c)

Fig. 12 Traffic from the opposite lane

The traffic at the opposite lane is another common traffic pattern, and a major source of false alarms for any potential ICD system. Fig. **12**(a) shows the host and threat vehicles are travelling with normal lateral separation (1.95 meters), and Fig. 12(b) shows the host and threat vehicles are travelling with less than usual lateral separation (0.61 meters). The MCICD algorithm correctly estimates the increased collision probability with reduced lateral separation. Fig. **12**(c) shows both cars are making right turns and MCICD algorithm correctly labels it as a near miss scenario.



(a) (b)

Fig. 13 T-bone collisions

T-bone collision is yet another important collision scenario, due to the potential severity of the vehicle damage and occupant injury. In Fig. 13(a) both cars are travelling at constant speed, and in Fig. 13(b) both cars are turning and accelerating. In both cases MCICD algorithm correctly detects the T-bone type collision.

2) FAR/FNR Analysis

It is desirable to have a single FAR/FNR encapsulating the overall system performance. Since the algorithm performance may differ across scenarios, the outcome of the FAR/FNR is thus dependent on the selected testing scenarios.

In this report, we extract scenarios based on the real world crashes [24], and then extrapolate the parameters to generate a testing set including both crash and non-crash scenarios.

Specifically we categorize the testing scenarios into 6 categories shown in Table 2, according to (1) the angle between the heading directions of the host and threat vehicles, and (2) whether the vehicle motion is constant speed or any motion involving acceleration/deceleration or turning. For example, 0DegConst includes testing scenarios where both vehicles driving at the same direction with constant speed, and 90DegNonConst includes scenarios of two vehicles at 90

degree intersections driving with acceleration/deceleration and/or turning motion.

Table 2 Simulation result

Testing Scenarios	FAR (total cases)	FNR (total cases)
0DegConst	0.24% (4595)	0% (2698)
0DegNonConst	0.12% (19406)	0.031% (9700)
90DegConst	0.41% (2182)	0.17% (1778)
90DegNonConst	0.23% (30884)	0.21% (24788)
180DegConst	0.24% (4595)	0% (2698)
180DegNonConst	0.11% (32236)	0.17% (12358)
Overall	0.17% (93898)	0.15% (54020)

B. Results on Real Data

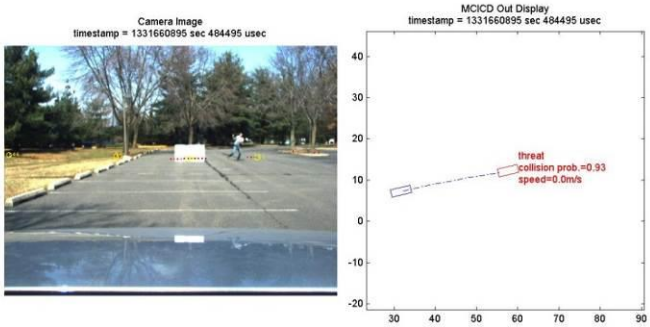
We mount the laser sensor and the MCICD system on a Ford Escape SUV as in Fig. 14. Live testing has been performed in a parking lot with staged scenarios, as well as in rural road driving situations.

Fig. 14 Interior and exterior of the MCICD system mounted on a Ford Escape.

1) Staged scenarios

One goal of this work is to show that the MCICD system can differentiate true collisions from near-misses. Staged collision scenarios are set up with cardboard objects, and near-miss scenarios are set up with a stationary car.

Fig. **15** shows an example run of the staged collision scenario. The plot on the left of the top row of Fig. **15** shows the overall trajectory of the host vehicle running over the stationary threat vehicle (cardboard object). The threat vehicle is colored in red. The images on the left show the scene captured at different time frames, with the laser return overlaid as red dots. The plot on the right shows the MCICD detection output, with the detected collision probability and threat vehicle speed. An online video is located at http://youtu.be/clVlzlg4llI.



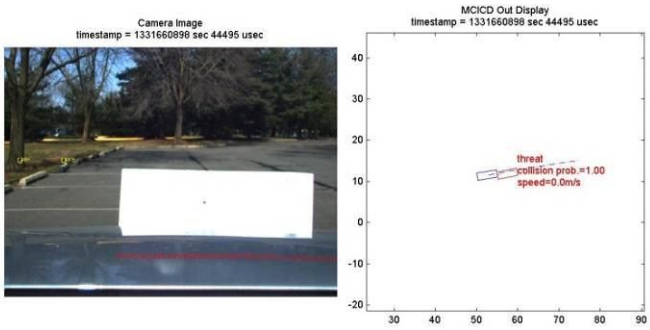


Fig. 15 Overall trajectory and snapshots of a staged collision scenario.

Fig. **16** shows an example run of the staged near-miss scenario. The plot on the right of the top row of Fig. **15** shows the overall trajectory of the host vehicle passing by the stationary vehicle with a narrow margin. In Fig. **16** the threat vehicle is colored in red. The images on the left are the snapshots of the scene, and the plots on the right show the detected collision probability. As can be seen from the figure, the MCICD algorithm correctly detects the scenario as non-collision. An online video is located at http://youtu.be/pp0va5XYZds.

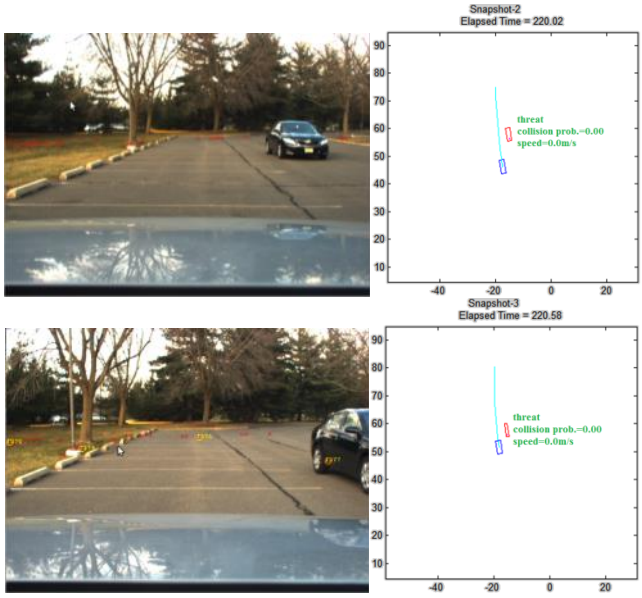


Fig. 16 Overall trajectory and snapshots of a staged near-miss scenario.

2) Road data

We also performed live road tests with the MCICD system in rural traffic.

Fig. **17** are two snapshots from a driving scenario on a curved road. The collision probability is detected as 0.10 when the threat vehicle is posed at a certain angle toward the host vehicle during the turn, and the collision probability is back to 0 when the turn is complete.

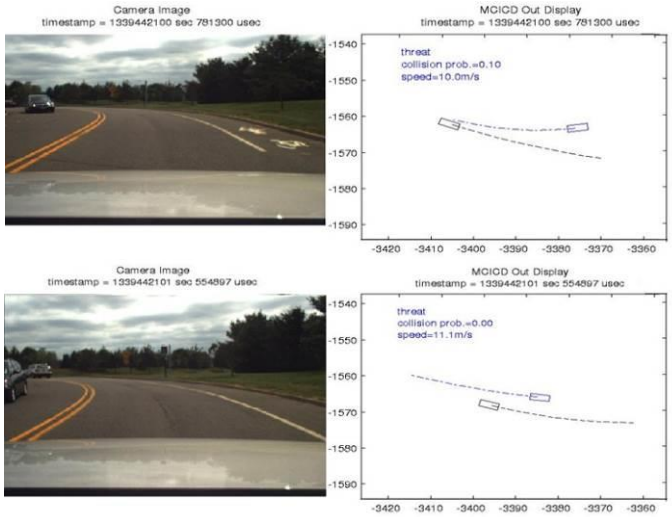


Fig. 17 Snapshots of a sequence at a curved road.

V. CONCLUSION

In this paper, we advocate the importance of realistic sensor noise modeling. A Monte-Carlo based MCICD algorithm is presented, which can incorporate the realistic sensor noise models and generate better FAR/FNR results for imminent collision detection. The MCICD algorithm is especially useful when the real noise models deviate from the commonly assumed Gaussian model, which we find is the case in practice. We validate the MCICD algorithm through both simulated data and real traffic data.

REFERENCES

[1] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 1492–1498.

[2] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: a probabilistic perspective," in Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, pp. 4022–4027

[3] P. Chang, T. Camus and R. Mandelbaum, "Stereo-based vision system for automotive imminent collision detection," in *IEEE Intelligent Vehicles Symposium*, 2004.

[4] Y. Chang, C. Yu, and L. Lee. "Research of Vehicle-Mounted Detection and Collision Avoidance System Based on Dedicated Short Range Communications." IEEE International Symposium on Computer, Consumer and Control (IS3C), 2016.

[5] E. Dagan, O. Mano, G.P. Stein and A. Shashua, "Forward collision warning with a single camera," in *IEEE Intelligent Vehicles Symposium*, 2004.

[6] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using monte carlo sampling," Intelligent Transportation Systems, IEEE Transactions on, vol. 9, no. 1, pp. 137–147, 2008.

[7] T. Fraichard and H. Asama, "Inevitable collision states a step towards safer robots?" Advanced Robotics, vol. 18, no. 10, pp. 1001–1024, 2004.

[8] J. Jie, A Khajepour and W. Melek, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints." IEEE Transactions on Vehicular Technology, 66.2 (2017): 952-964.

[9] Michael I. Jordan, *Learning in Graphical Models*. The MIT Press, 1998.

[10] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," Ieee Transactions on Intelligent Transportation Systems, vol. 10, no. 4, pp. 678–687, 2009

[11] T. Kim and H. Jeong, "Crash probability and error rates for head-on collisions based on stochastic analyses," *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 4, 2010.

[12] R. Labayrade, C. Royere and D. Aubert, "A collision mitigation system using laser scanner and stereovision fusion and its assessment," Proc. IEEE Intell. Veh. Symp., 2005, pp. 441-446

[13] H. B. Pacejka, *Tyre and Vehicle Dynamics*. London, U.K.: Butterworth Heinemann, 2002

[14] K. Lee and H. Peng, "Evaluation of automotive forward collision warning and collision avoidance algorithms," Vehicle System Dynamics, Vol. 43, No. 10, October 2005, pp. 735–751.

[15] C. Mertz, "A 2d collision warning framework based on a Monte Carlo approach," in *Proc. ITS America 14th meeting and expo*, Apr. 2004.

[16] C. Mertz, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, A. Steinfeld, A. Suppe, C. Thorpe and C. Wang, "Collision warning and sensor data processing in urban areas," in *Proc. 5th Int. Conf. ITS Telecommunications*, 2005, pp. 73-78.

[17] S. Pietzsch, O. Aycard, J. Burlet, T. D. Vu, T. Hackbarth, N. Appenrodt,J. Dickmann and B. Radig, "Results of a precrash application based on laserscanner and short wave radars," *IEEE Intelligent Vehicles Symposium*, 2008.

[18] G. Schildbach, M. Soppert, and F. Borrelli. "A collision avoidance system at intersections using Robust Model Predictive Control." IEEE Intelligent Vehicles Symposium (IV), 2016.

[19] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, September 2005.

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

[21] P. Tu and J. Kiang, "Estimation on location, velocity, and acceleration with high precision for collision avoidance", *IEEE Trans. Intelligent Transportation Systems*, vol. 11, no. 2, 2010.

[22] Wang, Pangwei, et al. A Novel Cooperative Collision Avoidance Model for Connected Vehicles. TRB Conference 2017.

[23] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *IEEE Trans. Intelligent Transportation Systems*, vol. 10, no. 4, 2009.

[24] Traffic Safety Facts 2012, published by NHTSA of the Department of Transportation of United States.

[25] Zolock, John, et al. The Use of Stationary Object Radar Sensor Data from Advanced Driver Assistance Systems (ADAS) in Accident Reconstruction. No. 2016-01-1465. SAE Technical Paper, 2016.