# Collision Probability Assessment for Speed Control

Alain Lambert*, Dominique Gruyer**, Guillaume Saint Pierre**, Alexandre Ndjeng Ndjeng**

*IEF, UMR CNRS 8622
Université Paris Sud-XI
Bat. 220, Centre d'Orsay
91405 Orsay cedex - France

**LIVIC
INRETS/LCPC
Bat 824, 14 route de la minière
78000 Versailles Satory - France

*Abstract*— **In order to navigate safely, it is important to detect and to react to a potentially dangerous situation. Such a situation can be underlined by a judicious use of the locations and the uncertainties of both the navigating vehicle and the obstacles. We propose to build an estimation of the collision probability from the environment perception with its probabilistic modelling. Then this probability is used for updating a braking order applied to our vehicle either to avoid or to mitigate a collision. The probability of collision is computed from a product of integrals of a product of Gaussians. The integrals take into account the uncertain configurations and the volume of both the vehicle and the obstacles.**

## I. Introduction

The anticipation of a collision is necessary for a safe navigation. The prediction of collisions could be used for obstacle avoidance, speed control, speed monitoring or path planning. We chose to deal with speed control in order to follow a path in the best condition by increasing the speed if the situation is safe and by lowering it if the situation is risky. The assessment of driving situation (which rely on the study of future possible collisions) has been computed in various ways during the last years.

[1] defines a security area modeled by a circle (centered on the robot position) whose radius is proportional to the speed. A collision judgement is based on an intersecting test between this circle and high-confidence position error ellipses. Controlling the speed and steering of the mobile robot along a preplanned path is done by using the collision judgement. [2] uses an interaction component (deformable virtual zone) of the robot with the environment which leads to avoidance-oriented control laws. Furthermore an emergency area around the robot causes an emergency stop if it is broken by an obstacle. [3] performs an on-line speed monitoring by computing a time to collision. Longer times to collision lead the higher speed. In order to detect a collision, the authors grow a mobile robot with its uncertainty ellipse and they do a collision test between the resulting shape and the obstacles. The process is repeated all along the path in order to compute a time to collision. Uncertainty ellipses have also been used in [4] for safe path planning. The safety is realized thanks to a collision test between a robot enlarged with its uncertainty ellipsoid and the obstacles. [5] computes a distance to undecided regions (unknown region) or to nearby obstacles. Next they use this distance information to compute the speed.

We think that only using a measured distance to collision [5][6] is not sufficient as the real distance could be quite different and lead to unexpected collisions. Using a security area [1][2] around the vehicle is a good idea only if this security area represents the uncertainty on the vehicle location. Nevertheless such an area (an ellipse [3][4]) is a discrete and binary representation of a continuous probability of presence. Most authors uses an ellipse which represents the probability of presence of the vehicle at 90%. Unfortunately by defining a threshold they loose information for higher level algorithms.

That's why in this paper we propose to use the entire available information (the pdf of the vehicle and the obstacles) for defining the probability of collision. Such an approach has been followed in [7] for a punctual robot and a geometrical obstacle without considering the uncertainty in orientation. We are going to overcome those restrictions (punctual and no uncertainty in orientation) in order to compute a realistic collision probability for real world application. To the best of our knowledge, it is the first time that the 3D uncertainty and the volume of the objects are used when calculating the probability of collision. Consequently, it is also the first time that such a probability of collision obtained from the pdf of the objects is used for providing a speed control.

In the next section we introduce the necessary models. In section 3 we propose an analytical formula for computing the probability of collision between two configurations. The general case (both vehicle and obstacle with any shape and added uncertainties) is studied in section 4. In section 5 we consider the probability of collision between multiple objects. Section 6 introduces the risk notion and uses this notion to defining a safe speed and next a safe acceleration (or deceleration). Finally we provide simulation results.

## II. Problem statement

### A. Vehicle and obstacle models

The vehicle configuration is denoted $\mathbf{x}_v = (x_v, y_v, \theta_v)^T$ where $(x_v, y_v)$ are the coordinates of a characteristic point which is located midway between the two rear wheels of our vehicle and $\theta_v$ is its orientation. All variables are defined with respect to the global frame. The evolution of the vehicle state could be written as follows:

$$\begin{aligned}\hat{\mathbf{x}}_{\mathbf{v}}(t+\Delta t) &= f\left(\mathbf{x}_{\mathbf{v}}\left(t\right), \Delta s, \Delta\theta\right) \\ &= \begin{pmatrix} x_v(t)+\Delta s\cdot\cos(\theta_v(t)+\Delta\theta_v/2) \\ y_v(t)+\Delta s\cdot\sin(\theta_v(t)+\Delta\theta_v/2) \\ \theta_v(t)+\Delta\theta_v \end{pmatrix}\end{aligned} \tag{1}$$

where $\Delta s$ and $\Delta\theta$ are the incremental longitudinal and rotational motion which can be computed from odometers or from an inertial central.

The obstacles configuration is denoted $\mathbf{x}_o = (x_o, y_o, \theta_o)^T$. The shape of both vehicle and obstacles are denoted $\mathcal{V}_v$ and $\mathcal{V}_o$. The geometric center of $\mathcal{V}_o$ is $\mathbf{x}_o$. The shape and uncertain configuration of the obstacles could be given by a map, a wireless communication system or estimated by embedded sensors. If a communication system is available, the moving obstacles are assumed to send their data; the infrastructure sensors can also detect the objects and send similar information. This paper does not deal with such system and only focus on collision assessment and speed control.

### B. Uncertainty modeling

The pdf (probability density function) of a configuration $\mathbf{x} = (x, y, \theta)^{\mathrm{T}}$ having a $\mathbf{\Sigma}$ covariance matrix and an $\hat{\mathbf{x}}$ mean is:

$$p(\mathbf{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^3\sqrt{\det\mathbf{\Sigma}}} e^{-\frac{1}{2}\left((\mathbf{x}-\hat{\mathbf{x}})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\hat{\mathbf{x}})\right)} \tag{2}$$

Such a matrix could be the result of a filter process like the Extend Kalman Filter or could be directly defined by:

$$\mathbf{\Sigma} = E\left(\left(\mathbf{x}-\widehat{\mathbf{x}}\right)\left(\mathbf{x}-\widehat{\mathbf{x}}\right)^T\right) \tag{3}$$

The pdf of a $v$ vehicle and an $o$ obstacle are denoted $p_v$ and $p_o$ with their associated $\mathbf{\Sigma}_v$ and $\mathbf{\Sigma}_o$ matrices.

Finally, $v$ and $o$ are defined by: $v = (\mathbf{x}_{\mathbf{v}}, \mathbf{\Sigma}_v, \mathcal{V}_v)$ and $o = (\mathbf{x}_o, \mathbf{\Sigma}_o, \mathcal{V}_o)$.

### C. Prediction using proprioceptive sensor

The prediction equation which use proprioceptive (odometric) measurement is given by the vehicle model (Eq. (1)). We need to compute the new uncertainty matrix $\mathbf{\Sigma}_k$ with the help of $\mathbf{\Sigma}_{k-1}$ and $\mathbf{Q}_{k-1}$. Assuming that $\widehat{\mathbf{x}}_{v,k-1}$, $\Delta s$ and $\Delta\theta$ are not correlated, and due to $f$ non-linearity, $\mathbf{\Sigma}_k$ calculation is achieved by a first order Taylor expansion.

$$\mathbf{\Sigma}_{k/k-1} = \mathbf{F}_{k-1}\mathbf{\Sigma}_{k-1/k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{4}$$

with

$$\begin{aligned}\mathbf{F}_{k-1} &= \left(\frac{\partial f}{\partial \mathbf{x}}\right)_{\mathbf{x}=\widehat{\mathbf{x}}_{k-1}} \\ &= \begin{bmatrix} 1 & 0 & -\Delta s\cdot\sin\left(\widehat{\theta}_{k-1}+\Delta\theta/2\right) \\ 0 & 1 & +\Delta s\cdot\cos\left(\widehat{\theta}_{k-1}+\Delta\theta/2\right) \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{5}$$

Using Eq. (4) we can compute the covariance matrix of the vehicle along a path at each time instant $k$.

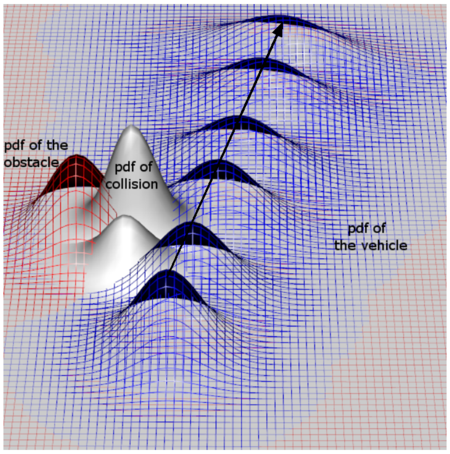


Fig. 1. pdf of the collision between an obstacle and a vehicle moving in straight line

## III. Probability of collision between 2 uncertain configurations

The probability of collision between a $v$ and an $o$ uncertain configurations (assuming that $\mathcal{V}_v = \mathcal{V}_o = \emptyset$) is defined by :

$$\mathbb{P}_{coll}(v,o) = \iiint_{\mathbb{R}^3} p_v(x,y,\theta)\cdot p_o(x,y,\theta)\cdot dxdyd\theta \tag{6}$$

The integral (6) can be analytically computed. Let's assume that $p_v(x,y,\theta)$ is the density of a $\mathcal{N}_d\left(\widehat{\mathbf{x}}_v, \mathbf{\Sigma}_v\right)$ distribution, and $p_o(x,y,\theta)$ the density of a $\mathcal{N}_d\left(\widehat{\mathbf{x}}_o, \mathbf{\Sigma}_o\right)$ distribution. Let us denote $\mathbf{x} = (x,y,\theta)^T$, $\mathbf{A} = \left(\mathbf{x}-\widehat{\mathbf{x}}_v\right)^T\mathbf{\Sigma}_v^{-1}\left(\mathbf{x}-\widehat{\mathbf{x}}_v\right) + \left(\mathbf{x}-\widehat{\mathbf{x}}_o\right)^T\mathbf{\Sigma}_o^{-1}\left(\mathbf{x}-\widehat{\mathbf{x}}_o\right)$ and $\mathbf{B} = \left[\mathbf{x}-\mathbf{m}\right]^T\mathbf{\Sigma}^{-1}\left[\mathbf{x}-\mathbf{m}\right]$ with:

$$\mathbf{m} = \mathbf{\Sigma}^{-1}\left(\mathbf{\Sigma}_v^{-1}\widehat{\mathbf{x}}_v + \mathbf{\Sigma}_o^{-1}\widehat{\mathbf{x}}_o\right) \tag{7}$$

$$\mathbf{\Sigma}^{-1} = \left(\mathbf{\Sigma}_v^{-1} + \mathbf{\Sigma}_o^{-1}\right) \tag{8}$$

We have

$$\mathbb{P}_{coll}(v,o) = \frac{\exp\left[-\frac{1}{2}\left(\mathbf{A}-\mathbf{B}\right)\right]\sqrt{\det\left(\mathbf{\Sigma}\right)}}{\sqrt{\det\left(\mathbf{\Sigma}_v\right)}\sqrt{\det\left(\mathbf{\Sigma}_v\right)}} \tag{9}$$

with

$$\begin{aligned}\mathbf{A}-\mathbf{B} &= \widehat{\mathbf{x}}_v^T\mathbf{\Sigma}_v^{-1}\widehat{\mathbf{x}}_v + \widehat{\mathbf{x}}_o^T\mathbf{\Sigma}_o^{-1}\widehat{\mathbf{x}}_o \\ &- \left(\mathbf{\Sigma}_v^{-1}\widehat{\mathbf{x}}_v + \mathbf{\Sigma}_o^{-1}\widehat{\mathbf{x}}_o\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{\Sigma}_v^{-1}\widehat{\mathbf{x}}_v + \mathbf{\Sigma}_o^{-1}\widehat{\mathbf{x}}_o\right)\end{aligned} \tag{10}$$

Equation (9) has been used to compute the probabilities of collision of Fig. 1. During this experiment corresponding to an outdoor situation, a car (so called “the vehicle”, right part of the figure) was running on its way whereas there was another static car (so called “the obstacle”) on the opposite lane (left part of the figure). The vehicle was moving from the bottom to the top of the figure in a straight line using only its proprioceptive sensors. The pdf of the vehicle is shown at 6 different time instants (every 4 meters). The pdf of the collision has been computed at each of those time instants but only 2 pdf are noticeable. The maximum height of the

**Algorithm 1** Probability of collision between $v$ and $o$

```
function ProbabilityOfCollision($v, o$)
    $\mathbb{P}_{coll}(v,o) \leftarrow 0$
    for $j \leftarrow 1$ to $N$ do
        $\mathbf{x}_v \leftarrow randc(\widehat{\mathbf{x}}_v, \mathbf{\Sigma}_v)$
        $\mathbf{x}_o \leftarrow randc(\widehat{\mathbf{x}}_o, \mathbf{\Sigma}_o)$
        if $\mathcal{V}_v(\mathbf{x}_v) \cap \mathcal{V}_o(\mathbf{x}_o) \neq \emptyset$ then
            $\mathbb{P}_{coll}(v,o) \leftarrow \mathbb{P}_{coll}(v,o) + 1$
        end if
    end for
    $\mathbb{P}_{coll}(v,o) \leftarrow \frac{\mathbb{P}_{coll}(v,o)}{N}$
    return($\mathbb{P}_{coll}(v,o)$)
end function
```

biggest pdf of the collision is tiny (0.00205) compared to the corresponding pdf's height of both the obstacle (0.16) and the vehicle (0.13). Consequently the pdf of the collision has been multiplied by 100 on Fig. 1 for a better visualization. The greatest probability of collision computed using Eq. (9) is equal to 0.007. It allows us to conclude that the situation is safe which is unrealistic considering the real situation with the volume of the cars. We should have thought about taking into account the volumes as they are big regarding the estimated distance between the vehicles. We have not investigated the interest of Eq. (9) when the volumes are not null because the following approaches (see section 4 and 5) provides results that are good enough both in term of computing time and precision.

## IV. Probability of collision between any 2 objects with Gaussian uncertainties

### A. Analytical description of the problem

The probability of collision between a $v$ and an $o$ object is the probability that $v$ and $o$ share a same part of the space. Consequently, given an $\hat{\mathbf{x}}_v$ configuration (with a $p_v$ associated pdf and a $\mathcal{V}_v$ volume) and an $\hat{\mathbf{x}}_o$ configuration (with a $p_o$ associated pdf and a $\mathcal{V}_o$ volume) the probability of collision is given by:

$$\mathbb{P}_{coll}(v,o) = \int_D p_v(x_v, y_v, \theta_v) \cdot p_o(x_o, y_o, \theta_o) \cdot dx_v dy_v d\theta_v dx_o dy_o d\theta_o \quad (11)$$

with

$$D = \{\ (x_v, y_v, \theta_v, x_o, y_o, \theta_o) \in \mathbb{R}^6 \;\backslash\; \mathcal{V}_v(x_v, y_v, \theta_v) \cap \mathcal{V}_o(x_o, y_o, \theta_o) \neq \emptyset \} \quad (12)$$

If $v$ and $o$ are punctual objects then Eq. (11) turns to Eq. (6). If $v$ and/or $o$ have an infinite volume then $v$ and $o$ always collide and Eq. (11) equals one.

### B. Monte Carlo solution

As we have no analytical solution to Eq. (11), we propose to use a MC (Monte Carlo) method.

First, we need to rewrite Eq. (11) as :

$$\mathbb{P}_{coll}(v,o) = \int_{\mathbb{R}^6} \Upsilon \cdot p_v(x_v, y_v, \theta_v) \cdot p_o(x_o, y_o, \theta_o) \cdot dx_v dy_v d\theta_v dx_o dy_o d\theta_o \quad (13)$$

Where $\Upsilon = \Upsilon(\mathcal{V}_v(x_v, y_v, \theta_v), \mathcal{V}_o(x_o, y_o, \theta_o))$ is a collision test between the volume $\mathcal{V}_v$ of the vehicle and the volume $\mathcal{V}_o$ of the obstacle:

$$\Upsilon(\mathcal{V}_v(x_v, y_v, \theta_v), \mathcal{V}_o(x_o, y_o, \theta_o)) = \begin{cases} 1 & \text{if } \mathcal{V}_v(x_v, y_v, \theta_v) \cap \mathcal{V}_o(x_o, y_o, \theta_o) \neq \emptyset \\ 0 & \text{if } \mathcal{V}_v(x_v, y_v, \theta_v) \cap \mathcal{V}_o(x_o, y_o, \theta_o) = \emptyset \end{cases} \quad (14)$$

Denoting $z = (x_v, y_v, \theta_v, x_o, y_o, \theta_o)$ (and $Z$ the associated random variable), Eq. (13) can be rewritten as:

$$\mathbb{P}_{coll}(v,o) = \int_{\mathbb{R}^6} \Upsilon(z) f(z) dz, \quad (15)$$

with $f(z) = p_v(x_v, y_v, \theta_v) \cdot p_o(x_o, y_o, \theta_o)$ and $\Upsilon(z) = \Upsilon(\mathcal{V}_v(x_v, y_v, \theta_v), \mathcal{V}_o(x_o, y_o, \theta_o))$.

Secondly we know [8] that for evaluating the integral

$$\int_{\mathbb{R}^6} \Upsilon\left(z\right) f\left(z\right) dz = \mathbb{E}_f\left[\Upsilon(Z)\right] \quad (16)$$

we can use a sample $(z_1, ..., z_m)$ generated from the density $f$. Therefore, Eq. (16) can be approximated by the empirical average

$$\bar{\Upsilon}_m = \frac{1}{m} \sum_{j=1}^{m} \Upsilon(z_j), \quad (17)$$

since $\bar{\Upsilon}_m$ converges almost surely to $\mathbb{E}_f\left[\Upsilon(Z)\right]$ by the Strong Law of Large Numbers. Variable $z$ is Gaussian as $(x_v, y_v, \theta_v)$ and $(x_o, y_o, \theta_o)$ are both Gaussian and independent. Therefore, the sample $(z_1, ..., z_n)$ can be obtained by generating separately $(x_{v_j}, y_{v_j}, \theta_{v_j}) \sim p_v(x_v, y_v, \theta_v)$ and $(x_{o_j}, y_{o_j}, \theta_{o_j}) \sim p_o(x_o, y_o, \theta_o)$.

Thanks to Eq. (17) we can rewrite Eq. (13) as:

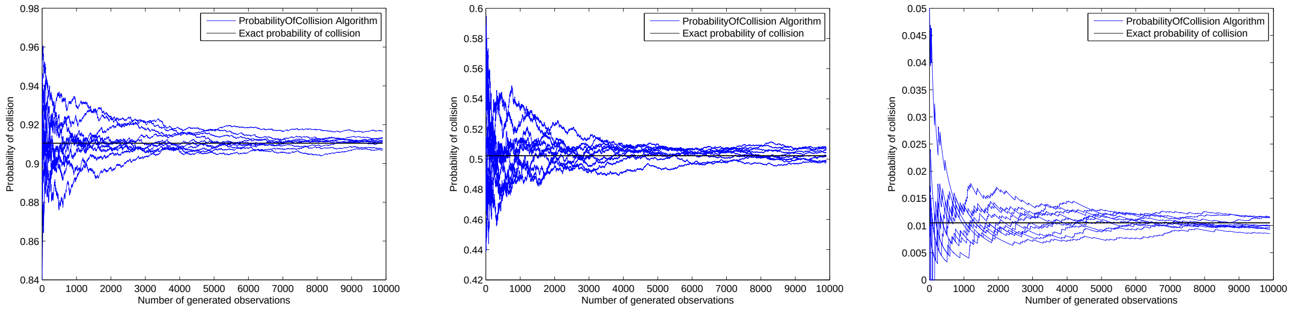
$$\mathbb{P}_{coll}(v,o) = \frac{1}{m} \sum_{j=1}^{m} \Upsilon(\mathcal{V}_v(x_{v_j}, y_{v_j}, \theta_{v_j}), \mathcal{V}_o(x_{o_j}, y_{o_j}, \theta_{o_j})). \quad (18)$$

Drawing samples $(x_v, y_v, \theta_v)$ and $(x_o, y_o, \theta_o)$ is done by an existing randc() function (many excellent generators exist to do such a job).

The Eq. (18) leads to algorithm 1 with a linear complexity in $O(N)$ where $N = m$ is a number that determines the accuracy of the integral computation.

### C. Experimental results

Results provided by algorithm 1 for different values of probability of collision have been shown on Fig. 2. The algorithm has approximated 3 different values of probability of collision: a high (Fig. 2.a), a medium (Fig. 2.b) and a low value (Fig. 2.c). Both vehicle and obstacles were represented as polygonal lines for the geometrical collision test inside the probabilistic collision test.



(a) High probability

(b) Medium probability

(c) Low probability

Fig. 2. Result of the proposed algorithm for an high, a medium and a low probability of collision

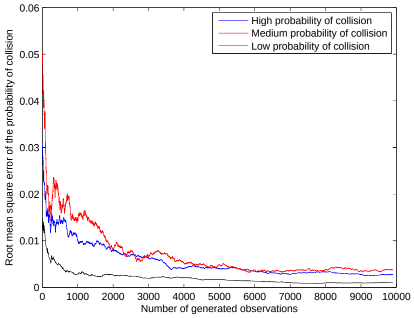


Fig. 3. Comparison of errors for an high, a medium and a low probability of collision

For each value the algorithm has been ran 10 times (although one run is sufficient to obtain an estimate of the probability) with until $10^4$ samples for each run ($N = 10^4$ in algorithm 1). Consequently the subfigures of Fig. 2 have 10 curves (plus the line of the exact value) and we can analyze various results of the algorithm on three typical situations. For each figure the "exact probability" corresponds to the mean after $10^6$ samples. On each subfigure the algorithm defines a corridor around the exact value.

The root mean square error (RMSE) is computed on Fig. 3 from the 3 subfigures of Fig. 2. The RMSE value is acceptable for the high and the medium probability values (less than 0.02 for 0.91 and 0.5 mean values after 1000 generated observations). Nevertheless the RMSE is big regarding the low probability value (less than 0.005 for 0.01 mean value after 1000 generated observations).

The proposed algorithm computes the probability of collision with $10^4$ samples (one run) in about 0.01 second on a Pentium IV processor (2 Ghz). Approximating the probability of collision takes 1 millisecond if we consider that only $10^3$ samples are necessary.

# V. Probability of collision between an object and other objects with Gaussian uncertainties

## A. Analytical description of the problem

The probability that $v$ collides with at least one obstacle can be calculated through the probability that $v$ does not collide with any obstacles :

$$\mathbb{P}_{coll}(v, o_1...o_n) = 1 - \overline{\mathbb{P}_{coll}(v, o_1)} \cdot \ldots \cdot \overline{\mathbb{P}_{coll}(v, o_n)} \quad (19)$$

The probability that $v$ do *not* collide with $o_i$ obstacle is

$$\overline{\mathbb{P}_{coll}(v, o_i)} = \int_{\mathbb{R}^6} \overline{\Upsilon} \cdot p_v(x_v, y_v, \theta_v) \cdot p_{o_i}(x_{o_i}, y_{o_i}, \theta_{o_i}) \, dx_v dy_v d\theta_v dx_o dy_o d\theta_o \quad (20)$$

with $\overline{\Upsilon} = \overline{\Upsilon(\mathcal{V}_v(x_v, y_v, \theta_v), \mathcal{V}_{o_i}(x_{o_i}, y_{o_i}, \theta_{o_i}))}$.

## B. Monte Carlo solution

Using Eq. (19) and (20) leads to algorithm 2 which is explained beneath.

- Lines 2-4: The probability $\overline{\mathbb{P}_{coll}(v, o_i)}$ that the $v$ vehicle does not collide with the $o_i$ obstacle is initialized to 0 for each of the $n$ obstacles.
- Lines 5 and 13: This first loop activates the computation of $\overline{\mathbb{P}_{coll}(v, o_i)}$ for each of the $n$ obstacles.
- Lines 6 and 12: This second loop computes $\overline{\mathbb{P}_{coll}(v, o_i)}$ using a new pair of samples at each iteration. The bigger the number $N$ of samples the better the accuracy.
- Lines 7 and 8: A $\mathbf{x}_v$ (respectively $\mathbf{x}_o$) sample is drawn following the pdf of the vehicle (respectively the $i$ obstacle).
- Lines 9-11: If the volume of the vehicle on $\mathbf{x}_v$ does not intersect the volume of the $i$ obstacle on $\mathbf{x}_o$ then the variable $\overline{\mathbb{P}_{coll}(v, o_i)}$ rises in increment of 1. The division of $\overline{\mathbb{P}_{coll}(v, o_i)}$ by the number of samples corresponds to the probability that the $v$ vehicle and the $o_i$ obstacle do not collide.
- Line 14 : The probability $\overline{\mathbb{P}_{coll}(v, o_{1..n})}$ that the vehicle does not collide with the obstacles is initialized to 1.
- Lines 15-17: $\overline{\mathbb{P}_{coll}(v, o_{1..n})}$ is updated according to $\overline{\mathbb{P}_{coll}(v, o_i)}$ (a variable which is proportional to the probability of collision with each of the obstacles).
- Line 18: The probability $\mathbb{P}_{coll}(v, o_1...o_n)$ that the vehicle collides with the obstacles is computed. $\overline{\mathbb{P}_{coll}(v, o_{1..n})}$ is divided $n$ times by $N$ where $N$ is the number of samples and $n$ is the number of obstacles.
- Line 19 returns the result.

**Algorithm 2** Probability of collision between $v$ and $o_1, ..., o_n$

```
function ProbabilityOfCollisions($v, o_1, ..., o_n$)
    for $i \leftarrow 1$ to $n$ do
        $\overline{\mathbb{P}_{coll}(v, o_i)} \leftarrow 0$
    end for
    for $i \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $N$ do
            $\mathbf{x}_v \leftarrow randc(\widehat{\mathbf{x}}_v, \mathbf{\Sigma}_v)$
            $\mathbf{x}_o \leftarrow randc(\widehat{\mathbf{x}}_{o_i}, \mathbf{\Sigma}_{o_i})$
            if $\mathcal{V}_v(\mathbf{x}_v) \cap \mathcal{V}_{o_i}(\mathbf{x}_o) = \emptyset$ then
                $\overline{\mathbb{P}_{coll}(v, o_i)} \leftarrow \overline{\mathbb{P}_{coll}(v, o_i)} + 1$
            end if
        end for
    end for
    $\overline{\mathbb{P}_{coll}(v, o_{1..n})} \leftarrow 1$
    for $i \leftarrow 1$ to $n$ do
        $\overline{\mathbb{P}_{coll}(v, o_{1..n})} \leftarrow \overline{\mathbb{P}_{coll}(v, o_{1..n})} \cdot \overline{\mathbb{P}_{coll}(v, o_i)}$
    end for
    $\mathbb{P}_{coll}(v, o_1...o_n) \leftarrow 1 - \frac{\overline{\mathbb{P}_{coll}(v, o_{1..n})}}{N^n}$
    return($\mathbb{P}_{coll}(v, o_1...o_n)$)
end function
```

The complexity of this algorithm is O($nN$) which is $n$ times the complexity of algorithm 1. This is verified by experimental results (with $N = 10^3$) where the computing time is $n$ milliseconds.

## VI. Defining a speed profile

We assume that a planned path $\mathcal{P}$ has already be defined. We want that the $v$ vehicle follows $\mathcal{P}$ at a speed which corresponds to an accepted risk of collision.

### A. Introduction

The more a vehicle is localized, the more we can know if a collision will occur or not. A vehicle alternates prediction steps and correction steps during its motion. The prediction step (link to a displacement) enlarges the configuration uncertainty whereas the correction step reduces it. Consequently, the best localization is achieved by doing numerous correction steps between small displacements. As the correction step is done at a constant rate, the only way to issue repeated correction step on a given path in comparison with small displacement is to reduce the speed. Consequently reducing the speed enable the best localization which reduce the probability of collision. Furthermore, reducing the speed reduces the crash consequences (cost of collision). A crash at small speed could have no consequences whereas a crash at high speed generally lead to the destruction of the car. But as we want that the vehicle arrives at destination as fast as possible, we should deal with a trade-off between low probability of collision and fast speed. This trade-off corresponds to an accepted risk of collision as described by Eq. (22) (section VI-B).

A human being follows the same speed reduction strategy then the one that we expect for an intelligent vehicle. For instance, he drastically reduces the speed of its car in a narrow corridor (which reduces the cost of collision which will be too higher for higher speeds). The speed could be very slow which allows the driver to quickly alternate left and right localization while steering the car (a better localization reduces the probability of collision). Despite everything, if a collision occurs at low speed it will generate small damages.

In this paper we are going to define a speed profile on a given time window. In order to compute the required speed for a given situation, we should use the localization uncertainties. As we want to compute future speed we should compute future uncertainties (and associated probability of collision). The future uncertainties are computed thanks to the prediction part (Eq. (4)) of the localization algorithm along $\mathcal{P}$.

In the next section we are going to introduce the risk of collision so as to compute an adequate speed along $\mathcal{P}$.

### B. From probability of collision to speed

A risk function is classically defined as a product of a probability by a cost. Consequently the risk of collision is given by:

$$risk_{coll}(v) = p_{coll}(v, o_1...o_n) \cdot cost_{coll}(v) \tag{21}$$

The cost of collision (also called severity of crashes) depends on the square of the vehicle's speed. Consequently, Eq. (21) could be rewritten as :

$$risk_{coll}(v) = p_{coll}(v, o_1...o_n) \cdot speed_v^2 \tag{22}$$

We will tolerate a maximum risk of collision $risk_{coll}^{max}$ as human does when driving. Consequently, we tolerate a maximum speed:

$$speed_v = \sqrt{\frac{risk_{coll}^{max}(v)}{p_{coll}(v, o_1...o_n)}} \tag{23}$$

If $speed_v$ is defined beyond the speed limit of the way then it is lowered to the speed limit.

We define a minimum value for $p_{coll}(v, o_1...o_n)$ which corresponds to the accuracy on the computation of $p_{coll}(v, o_1...o_n)$ (see section IV-C). Practically we have computed an upper bound (0.01) on the accuracy of the algorithm 2 and consider a maximum speed (100km/h) for the lowest probability of collision (0.01). Consequently, our maximum risk of collision is set to 1. Furthermore in order to stop our vehicle in the presence of an high risk of collision we consider that a small computed speed (inferior to 2km/h) is a null speed.

### C. From speed to acceleration

We want to define the acceleration of the vehicle at time $k$ in order to achieve a speed $s_{k+1}$ at time $k+1$. By considering a constant acceleration $a$ between $k$ and $k+1$ and by integrating it, we obtain:

$$s_{k+1} = a_k(t_{k+1} - t_k) + s_k \tag{24}$$

$$c_{k+1} = \frac{a_k(t_{k+1} - t_k)^2}{2} + s_k(t_{k+1} - t_k) + c_k \tag{25}$$

where $s_k$ and $c_k$ are respectively the speed and the curvilinear x-coordinate. Equation (24) can be rewritten as :

$$t_{k+1} - t_k = \frac{s_{k+1} - s_k}{a_k} \tag{26}$$

Next, we can replace $t_{k+1} - t_k$ in Eq. (25) thanks to Eq. (26):

$$a_k = \frac{s_{k+1}^2 - s_k^2}{2(c_{k+1} - c_k)} \tag{27}$$

Unfortunately, acceleration and deceleration are bounded value : $a_k \in [a_{min}, a_{max}]$. Practically, if the needed acceleration is bigger than $a_{max}$ then the vehicle will not achieve the higher required speed. This has no consequence on the path safety; it only increase the traveling time.

If the needed acceleration is lower than $a_{min}$ then the vehicle will not achieve the required security speed. Consequently, using Eq. (27), we propose to define a new speed $s_k^{max}$ that will replace the previous speed $s_k$:

$$s_k^{max} = \sqrt{s_{k+1}^2 - 2a_{min}(c_{k+1} - c_k)} \tag{28}$$

Next $a_{k-1}$ could be computed and so on until the current time.

### D. Experimental results

We have implemented the previous algorithm using C++ language inside the SiVIC simulator [9]. The SiVIC software architecture allows very easily to model virtual road environment including vehicles, infrastructure and sensors.

In order to prototype, to test and to evaluate our approach, we have used 2 different types of scenarios. These scenarios are presented in figures 4 and 5. In the two cases, the vehicle drives in straight line and we consider small uncertainties on the obstacles configurations. Whatever is the vehicle uncertainties, it drives at the full speed (100km/h) when it is far from the obstacles. When it becomes closer to the obstacles, it gradually reduces its speed. The negative acceleration depends on the vehicle uncertainty: the smaller is the vehicle uncertainty the bigger is the negative acceleration (if there is a near obstacle). In the case of figure 4, the vehicle passes trough the tunnel at a reduced speed (14km/h) and then gradually accelerates until the full speed. The brick wall scenario (figure 5) provides similar results. The vehicle goes out of the road (in straight line) to the center of the front wall of the house. As the uncertainty on the vehicle is big, the speed slowly decrease to 2km/h. Next the vehicle stops.

## VII. Conclusion

We have defined the probability of collision for a vehicle in a cluttered environment as a product of integrals of a product of Gaussians. The probability of collision takes into account the uncertainties and the volume of both vehicle and obstacles. Once the probability of collision is computed we can use it to compute a safe speed for the vehicle. By repeating this process over a given time windows and by retro-propagate it we can compute an acceleration profile.

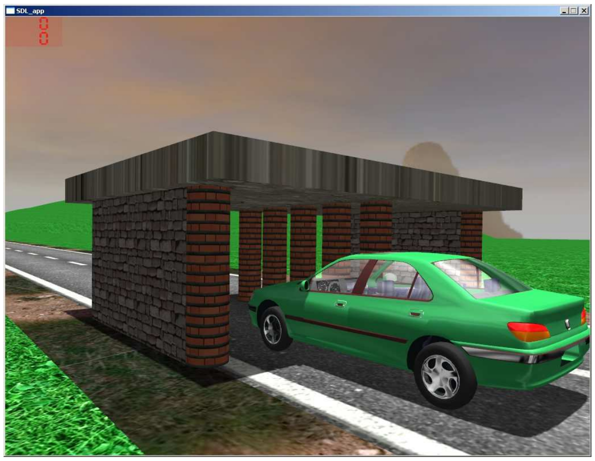


Fig. 4. The tunnel scenario

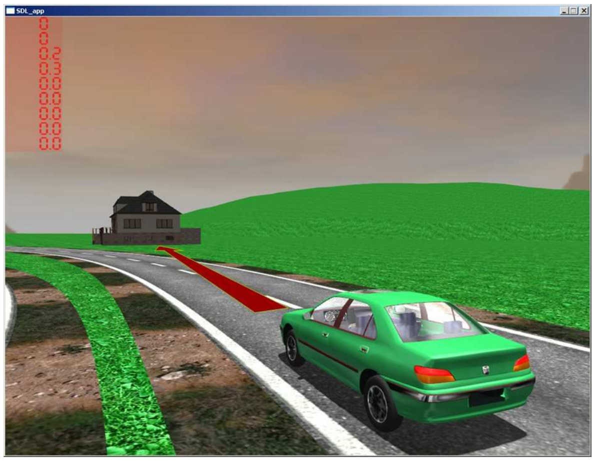


Fig. 5. The brick wall scenario

## References

[1] H. Hu, M. Brady, and P. Probert, “Navigation and control of a mobile robot among moving obstacles,” in *Proc. of the 30th IEEE International Conference on Decision and Control (CDC’91)*, Brighton, England, 11-13 Dec 1991, pp. 698–703.

[2] R. Zapata, P. Lepinay, and P. Thompson, “Reactive behaviors of fast mobile robots,” *Journal of Robotic Systems*, vol. 11, pp. 13–20, 1994.

[3] L. Codewener and D. Meizel, “On line speed monitoring of mobile robots tasks,” *Engineering applications of artificial intelligence*, vol. 7(2), pp. 152–160, 1994.

[4] A. Lambert and N. LeFort-Piat, “Safe task planning integrating uncertainties and local maps federation,” *International Journal of Robotics Research*, vol. 19(6), pp. 597–611, 2000.

[5] J. Miura, Y. Negishi, and Y. Shirai, “Adaptive robot speed control by considering map and motion uncertainty,” *Robotics and Autonomous Systems*, vol. 54(2), pp. 110–117, 2006.

[6] K. M. Krishna, R. Alami, and T. Simeon, “Safe proactive plans and their execution,” *Robotics and Autonomous Systems*, vol. 54, pp. 244–255, 2006.

[7] P. Burlina, D. DeMenthon, and L. Davis, “Navigation with uncertainty: reaching a goal in a high collision risk region,” in *Proc. of the IEEE International Conference on Robotics and Automation*, Nice, France, 12-14 May 1992, pp. 2440–2445 vol.3.

[8] C.P. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Verlag, 2004.

[9] D. Gruyer, C. Royere, and all, “Sivic and rtmaps, interconnected platforms for the conception and the evaluation of driving assistance systems,” in *Proc. of the Intelligent Transport Systems World Congress*, London, United Kingdom, 08-12 October 2006, pp. 1–8.