Transportation Research Part C 51 (2015) 66–82

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Extending Time to Collision for probabilistic reasoning in general traffic scenarios

James R. Ward [a,*], Gabriel Agamennoni [b], Stewart Worrall [a], Asher Bender [a], Eduardo Nebot [a]

[a] Intelligent Vehicles and Safety Systems Group, The Australian Centre for Field Robotics, The University of Sydney, NSW 2006, Australia
[b] Institute of Robotics and Intelligent Systems, ETH Zurich, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 1 July 2014
Received in revised form 2 October 2014
Accepted 6 November 2014
Available online 8 December 2014

Keywords:
Collision avoidance
Vehicle safety communications
Probabilistic model
Intelligent Transportation Systems
Vehicle to vehicle communication

ABSTRACT

Vehicle-to-vehicle communication systems allow vehicles to share state information with one another to improve safety and efficiency of transportation networks. One of the key applications of such a system is in the prediction and avoidance of collisions between vehicles. If a method to do this is to succeed it must be robust to measurement uncertainty and to loss of communication links. The method should also be general enough that it does not rely on constraints on vehicle motion for the accuracy of its predictions. It should work for all interactions between vehicles and not just a select subset. This paper presents a method to calculate Time to Collision for unconstrained vehicle motion. This metric is gated using a novel technique based on relative vehicle motion that we call “looming”. Finally, these ideas are integrated into a probabilistic framework that accounts for uncertainty in vehicle state and loss of vehicle-to-vehicle communication. Together this work represents a new way of considering vehicle collision estimation. These algorithms are validated on data collected from real world vehicle trials.

1. Introduction

One of the great advances in the development of automotive technologies is the increasing research into vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. These technologies facilitate the exchange of information between vehicles which has the potential to lead to more sophisticated and coordinated vehicle operations (Stiller et al., 2007; Ge and Orosz, 2014; Guler et al., 2014; Sepulcre et al., 2013). A number of trials of the communications technology have been conducted around the world (Stubing et al., 2010; Alexander et al., 2011). These trials tend to focus on lower level issues around reliability of connectivity, bandwidth or system reliability. Although the trials generally use the transmitted information to evaluate whether the communication systems provide sufficient warning to avert conflict (Hafner et al., 2013; Worrall et al., 2012), they do not tend to delve deeply into how the transmitted information can be best exploited by the receiving vehicle.

It must also be acknowledged that any state information exchanged between vehicles is subject to the uncertainties. The navigation information of all interacting vehicles, that is position, velocity and attitude will be known with different level of uncertainty. We expect smart vehicles with high accurate position to be able to interact in the most safe and efficient manner

[*] Corresponding author.
E-mail addresses: j.ward@acfr.usyd.edu.au (J.R. Ward), gabriel.agamennoni@mavt.ethz.ch (G. Agamennoni), s.worrall@acfr.usyd.edu.au (S. Worrall), a.bender@acfr.usyd.edu.au (A. Bender), nebot@acfr.usyd.edu.au (E. Nebot).

http://dx.doi.org/10.1016/j.trc.2014.11.002
0968-090X/© 2014 Elsevier Ltd. All rights reserved.

with other vehicles with basic navigation capabilities. This makes essential the development of algorithms that can be able to deal with this uncertainty to optimise the operation of a fleet of vehicles retrofitted with different quality sensing.

This paper presents three major contributions to the problem of vehicle collision prediction. Firstly, a method for generalising Time to Collision calculations to unconstrained vehicle motion is developed in Section 3. Secondly, a technique for gating predictions based on the relative motion of the vehicles (which we call *looming*) is presented in Section 4.2. Finally, in Section 6 these approaches are modified to account for uncertainties in vehicle states that are inherent in any real world deployment of V2V systems.

2. Background

Time to Collision (TTC) is a proximal safety indicator (Archer, 2005; Svensson, 1998) defined by Hayward (1972) as:

Time To Collision. The time required for two vehicles to collide if they continue at their present speed and on the same path.

We adopt the same definition in this paper, and we also calculate a second measure of TTC with the assumption of constant acceleration rather than constant speed.

Existing approaches to implementing TTC on real vehicles are most often limited to calculations relative to a vehicle directly in front of the ego vehicle in the same traffic lane. They often rely on camera systems or radar for the online calculation of TTC. The *MobileEye* system (Dagan et al., 2004) is an example of such an implementation. A technique for alerting drivers to potential collision was developed by Huang et al. (2009) however this system has not been demonstrated other than in simulation. The technique also does not account for uncertainty in state information which cannot be neglected if a system is to be of use in real world applications.

In this work we assume that the following states are observable and estimated by a filtering system (e.g. EKF/UKF):

- Position, $\mathbf{p}$ (m)
- Speed, v (m/s)
- Heading, ψ (°)
- Yaw rate, ω (°/s)

We model the movement of vehicles using an Unscented Kalman Filter (UKF) (Julier and Uhlmann, 1997) and thus we also have the uncertainty in the measurements of these states available as well. The UKF uses speed and yaw rate to perform predict steps, and the position information to perform update steps. It is assumed that vehicles are regularly broadcasting this state and uncertainty information to other vehicles in the vicinity. Messages transmitted by vehicles employing the SAE J2735 DSRC Message Set include the required information.

Although each vehicle may use different algorithms for state estimation, if the estimated state and uncertainty are broadcast they can be used as the inputs to a UKF. Thus, the ego vehicle maintains a UKF for each vehicle in its vicinity in order to estimate their states. Speed and yaw rate are used as inputs to the UKF, and position and heading are maintained as states. If communications between vehicles is lost, predict steps are continued with the most recent speed and yaw rate. When communications are reestablished, the UKF is reinitialised with the most recent state transmitted by the intruder vehicle.

3. Generalised Time-To-Collision

In this section we introduce an extension of the well-known Time-To-Collision (TTC) safety indicator to the more general case of two-dimensional movement. With this indicator, and an additional one defined in the next section, we will derive a method for predicting conflicting trajectories based on the relative movement between two vehicles.

Generalising the TTC to the planar case is surprisingly challenging. In the vehicle following scenario the motion is one-dimensional, meaning that all physical quantities are scalars and can be fully characterised in terms of distance and relative speed. In the plane, however, the vehicles' states are vector quantities, and combining these vectors to form a scalar measure of proximity is non-trivial.

3.1. Time-To-Collision in the plane

Studies using the Time To Collision metric often treat the movement of the vehicles as a 1D problem. That is, the calculation is performed on a one dimensional manifold, such as the position along the road that the vehicles are driving along (Dabbour and Easa, 2014). For situations other than vehicle-following, there exists no safety indicator analogous to the TTC. For instance, applying the TTC at an intersection is ill-defined since there are infinitely many possible paths that vehicles can follow as they approach. In this paper we propose an indicator that generalises TTC to the planar case, thus extending its applicability to a much larger spectrum of traffic scenarios. Furthermore, our indicator reduces to the TTC in the vehicle-following scenario.

Vehicle movement is not always constrained to a 1D manifold. For example, tt cannot always be assumed that the vehicle will follow the lanes of the road. Indeed, this may be a situation in which calculating an accurate TTC is very important as we wish to alert the driver to threats caused by this non-standard driving. The vehicles may also be operating in an area where they are less constrained such as in a car park or in an industrial operation. We present the following method to calculate TTC in 2D.

3.2. Generalisation

Consider two vehicles with vector positions, velocities and accelerations of $\mathbf{p_i}$, $\mathbf{p_j}$, $\mathbf{v_i}$, $\mathbf{v_j}$ and $\mathbf{a_i}$, $\mathbf{a_j}$, respectively. The dimensions of the vehicles cannot be neglected so we define the position vectors $\mathbf{p_i}, \mathbf{p_j}$ to be the position of the point on the vehicle closest to the other vehicle at the given time step. The position of this point may change over time as the vehicles move relative to one another, however this behaviour does not adversely affect the algorithm. This is because any change in the position of this point is smoothly continuous from time step to time step. There are no abrupt movements of this point therefore the results of the algorithm are also smooth over time. The definitions are illustrated in Fig. 1.

The separation between the vehicles, d_{ij} is given implicitly by the following equation:

$$d_{ij}^2 = \|\mathbf{p}_i - \mathbf{p}_j\|^2 = (\mathbf{p}_i - \mathbf{p}_j)^\top (\mathbf{p}_i - \mathbf{p}_j)$$

Differentiating both sides of this equation yields the rate of change of the separation, $\dot{d}_{ij}$. (The closure rate is $-\dot{d}_{ij}$).

$$d_{ij}\dot{d}_{ij} = (\mathbf{p}_i - \mathbf{p}_j)^\top (\mathbf{v}_i - \mathbf{v}_j)$$

Differentiating again yields the second derivative of the separation, $\ddot{d}_{ij}$,

$$\dot{d}_{ij}^2 + d_{ij}\ddot{d}_{ij} = (\mathbf{v}_i - \mathbf{v}_j)^\top (\mathbf{v}_i - \mathbf{v}_j) + (\mathbf{p}_i - \mathbf{p}_j)^\top (\mathbf{a}_i - \mathbf{a}_j) \tag{1}$$

If we assume that the vehicles move at an almost constant velocity, so that $\mathbf{a}_i$ and $\mathbf{a}_j$ are sufficiently small, then the right term (inner product) on the right-hand side of (1) is much smaller than the left term and can be safely neglected.

To summarise, the separation between vehicles i and j and its first and second derivatives can be computed as follows:

$$\begin{aligned}
d_{ij} &= \sqrt{(\mathbf{p}_i - \mathbf{p}_j)^\top \ (\mathbf{p}_i - \mathbf{p}_j)} \\
\dot{d}_{ij} &= \frac{1}{d_{ij}} (\mathbf{p}_i - \mathbf{p}_j)^\top \ (\mathbf{v}_i - \mathbf{v}_j) \\
\ddot{d}_{ij} &= \frac{1}{d_{ij}} \left((\mathbf{v}_i - \mathbf{v}_j)^\top \ (\mathbf{v}_i - \mathbf{v}_j) - \dot{d}_{ij}^2 \right)
\end{aligned}$$

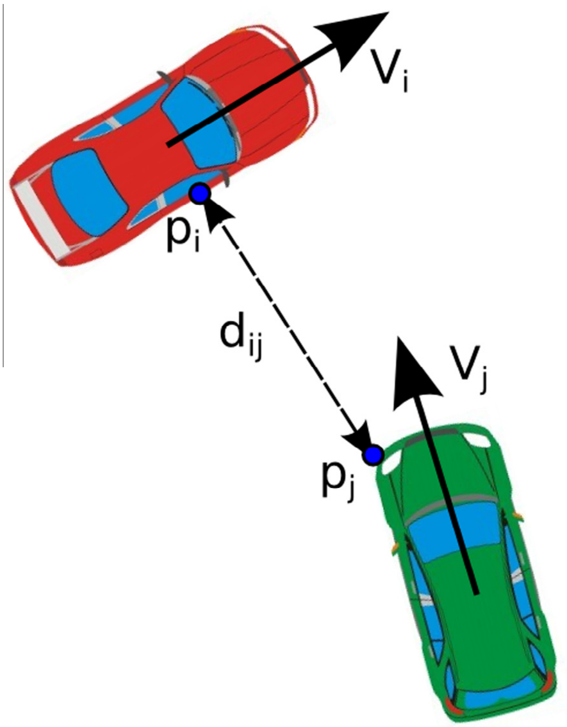


Fig. 1. Definition of vehicle state parameters. The position of each vehicle is the point on each vehicle closest to the other.

Time to Collision values can be calculated in two ways. We define *first order TTC*, T_1, as the TTC value calculated when change of closure rate is omitted. T_1 assumes constant closure rate between the vehicles. The *second order TTC*, T_2, accounts for changes in closure rate. In this paper when referring to Time to Collision, we are referring to T_2.

In both cases we solve the standard equations of motion to find the time when the displacement between the vehicles is 0.

In the first order case:

$$\begin{aligned} &d_{ij} + \dot{d}_{ij} T_1 = 0 \\ &T_1 = -\frac{d_{ij}}{\dot{d}_{ij}} \end{aligned} \tag{2}$$

The second order case:

$$d_{ij} + \dot{d}_{ij} T_2 + \frac{1}{2} \ddot{d}_{ij} T_2^2 = 0 \tag{3}$$

Eq. (3) can be solved using the quadratic formula. The discriminant, Δ, is therefore:

$$\Delta = \dot{d}_{ij}^2 - 2\ddot{d}_{ij} d_{ij}$$

In the second order case, if the acceleration term, $\ddot{d}_{ij}$, is zero it reverts to the first order case and $T_2 = T_1$. When Δ is negative there are no real roots and we define T_2 to be the time of closest approach. In the case where Δ is zero or positive we will have two real roots. When Δ is zero the two roots will be the same. In the case when the roots are both positive, we take the lower value as it is the first time that the vehicles will collide. If one root is positive and the other negative, we take the positive value as it represents a collision in the future, which is what we are interested in predicting. For the case where both roots are negative, we take the root with the lowest absolute value (i.e. closest to zero) as it represents the most recent interaction. Thus, the definition of T_2 is as follows:

$$T_2 = \begin{cases} T_1 & \text{if } \ddot{d}_{ij} = 0 \\ \frac{\dot{d}_{ij}}{\ddot{d}_{ij}} & \text{if } \Delta < 0 \\ \min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) & \text{if } \min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) \geqslant 0 \\ \max\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) & \text{if } \min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) < 0 \end{cases} \tag{4}$$

It will be noted in (2) and (4) that T_1 is undefined when $\dot{d}_{ij} = 0$, and T_2 is undefined when $\dot{d}_{ij} = 0$ and $\ddot{d}_{ij} = 0$. This is the degenerate case where the vehicles are stationary. There is no risk of collision in this case, thus we set the value to negative infinity. Such a value indicates that there is no predicted collision in the future no matter how large the time horizon and that no useful information can be gleaned about the predicted time since the last interaction.

4. Augmenting TTC

In this section we introduce a novel proximal safety indicator based on looming. This, together with the extension of the TTC in the previous section, will form the basis for predicting traffic conflicts from relative vehicle motion.

Aircraft pilots are often taught that other aircraft on a collision course will remain in the same relative position in the windscreen. More generally, if an object is approaching (i.e. the TTC is decreasing) and it lies on a collision course, then its relative bearing will not change. This is illustrated in Fig. 2.

If the relative bearing of the object is increasing (i.e. moving clockwise) then it will pass from left to right in front of the observer. A further generalisation allows this to be applied to an entire object, as the above technically only applies to the point of the vehicle on a collision course with the observation point.

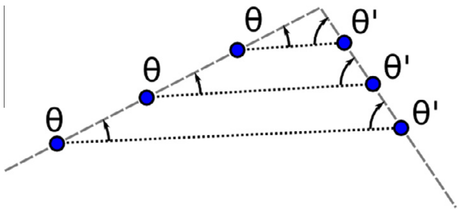


Fig. 2. Relative bearing of the other object remains constant when on a collision course.

4.1. Limitations of planar TTC

The calculations described in Section 3.1 are useful for predicting future collisions when vehicles are, in fact, on a collision course. However, if the vehicles are not on a collision course the TTC values are incorrect insofar as they predict a collision at some time in the future when this will not actually occur. Consider, for example, two vehicles driving in opposite directions down a straight road. They are not on a collision course but if the system is set to alert the driver when the TTC falls below, say, 2 s, T_2 and T_1 will both breach this value during the approach. A system that performed like this would issue an alert every time the driver passed oncoming traffic, making the system worthless as a safety aid.

If the system were able to determine that the trajectories of the vehicles intersected or not, this information could be used to gate any alarms raised by low TTC values. In this section we present such a method.

If the vehicles are assumed to be points moving in 2D space then it is trivial to project their velocity vectors forward and check for intersection. A further check must be made to find whether the vehicles reach the point of intersection at the same time, or whether one passes in front of the other. This assumption is not acceptable when performing calculations for vehicles where their dimensions are non-negligible when compared to the distance between them. The point at which a collision occurs may not correspond to the point at which the velocity vectors intersect. Calculating whether the 2D representations of the vehicles collide or not is complicated and time consuming.

4.2. Looming

We introduce the concept of *looming* to make this process simpler. Consider the relative bearings of the points of the angle subtended by the approaching object. As shown in Fig. 3 we denote these angles as α and β, respectively, with $\alpha > \beta$.

Then, the intermediate value theorem guarantees that

$$\dot{\beta} \leqslant 0 \wedge \dot{\alpha} \geqslant 0 \Rightarrow \exists \theta : \beta \leqslant \theta \leqslant \alpha \wedge \dot{\theta} = 0 \tag{5}$$

If the leftmost point of the object is moving anticlockwise relative to the observer (i.e. $\dot{\alpha} > 0$) and the rightmost point is moving clockwise ($\dot{\beta} < 0$), then the object is filling more and more of the observer's field of vision. In other words, the object is looming, as shown in Fig. 3.

Consider the movement of a point relative to another as shown in Fig. 4.

The *loom rate* of a point is its angular velocity from the point of view of the ego point, $\mathbf{p}_i$. This ego point is known as the *loom point*. The calculations of loom rates is performed as follows. Firstly, the linear velocity, $\bar{\mathbf{v}}_i$, of the loom point is calculated using the vehicle velocity, $\mathbf{v}_i$, the displacement of the loom point from the vehicle centre of rotation, $(\mathbf{p}_i - \mathbf{p}_c)$, and the vector yaw rate, $\boldsymbol{\omega}_i$:

$$\bar{\mathbf{v}}_i = \mathbf{v}_i + (\mathbf{p}_i - \mathbf{p}_c) \times \boldsymbol{\omega}_i$$

The linear velocity of the loom point is the vector sum of the vehicle velocity and the linear velocity due to the yaw of the vehicle about its centre.

To calculate the loom rate, $\dot{\theta}$:

$$\dot{\theta} = \frac{(\mathbf{p}_j - \mathbf{p}_i) \times \bar{\mathbf{v}}_i + (\mathbf{p}_j - \mathbf{p}_i) \times \mathbf{v}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|^2} \tag{6}$$

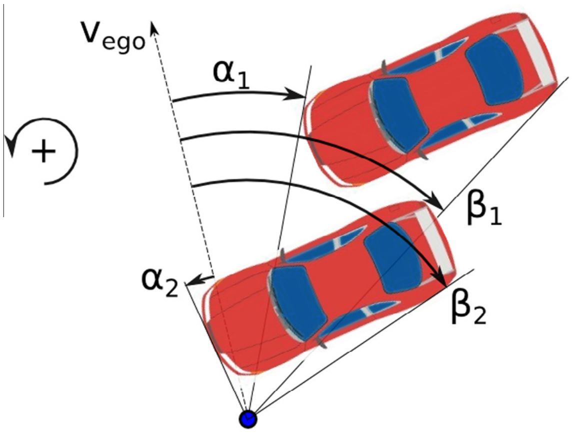


Fig. 3. Behaviour of relative bearings of leftmost and rightmost points of intruder vehicle.

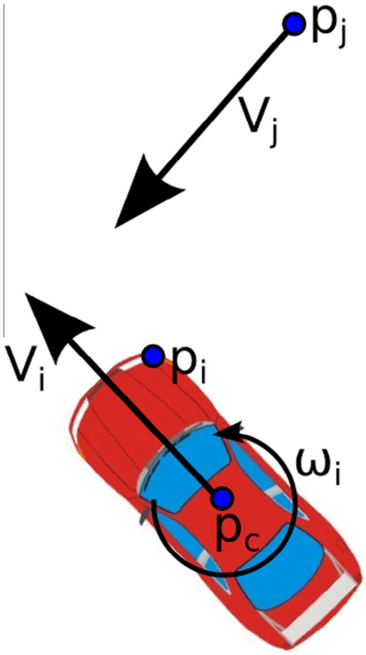


Fig. 4. Calculation of loom rates for closest intruder point $\mathbf{p}_j$.

Eq. (6) is the sum of angular velocity due to the movement of $\mathbf{p}_j$ around the loom point $\mathbf{p}_i$ and the angular velocity of the loom point about $\mathbf{p}_j$. As the equation uses the linear velocity of the loom point, $\bar{\mathbf{v}}_i$, rather than the linear velocity of the ego vehicle, the equation accounts for the yaw rate of the ego vehicle. Eq. (6) does not include the yaw rate of the intruder vehicle as its effect on the loom rate is negligible compared to the effect of the intruder's linear velocity, $\mathbf{v_j}$. Since the yaw rate of the intruder is not considered, the geometry of the intruder vehicle does not need to be considered like it is with the ego vehicle. This is because the linear velocity of the point $\mathbf{p}_j$ is the same as the linear velocity of the centre of the intruder vehicle.

This looming calculation can indicate whether an object is on a collision course with the point at which it is calculated. In the case of a vehicle it is possible for the other vehicle to be on a collision course with certain points on the ego vehicle but not on a collision course with others. The calculation must be performed at a number of points on the ego vehicle. We call these points *loom test points* or simply *loom points*. If the loom test points are chosen so that the minimum distance between points is less than the minimum dimension of the intruder vehicle it is not possible for a collision between vehicles to occur without the intruder vehicle being on a collision course with at least one of the loom test points.

Fig. 5 shows an example of two loom points placed too far apart to detect the impending collision of the green car. Both loom points have loom rates with the same sign which implies that the object will pass those loom points. This is true, however the green car will collide with the red car between the loom points. Clearly this is a situation that needs to be avoided by placing loom points sufficiently close together.

In practice one wants to bias the loom points towards the front of the vehicle as predicting the likelihood of collision with this part of the vehicle is more useful to the driver than predicting collision likelihoods for the very rear of the vehicle. In forward driving there is little that the driver can do to avoid a collision at the rear of the vehicle. Braking will typically increase the likelihood of this type of collision rather than reduce it. A typical selection of loom test points is shown in Fig. 6.

By only considering the TTC when Eq. (5) holds for at least one loom test point, a useful collision prediction system can be created. We call this approach *loom gated Time to Collision*.

Consider the vehicles travelling in opposite directions on a curved road in Fig. 7. This figure shows synthetic data with no uncertainties in vehicle states. At any given time we are interested in predicting the likelihood of collision in the subsequent 2 s. Collision likelihoods are shown when calculated using T_2 below the 2 s threshold, and when this is gated by the looming tests. As this is synthetic data with no uncertainty, the collision likelihood for a 2 s window is 100% when T_2 falls below 2 s. It is clear that the loom gating is able to eliminate the false collision probabilities that are calculated in the first case as the vehicles pass.

5. Predictive model

Given the theoretical basis for the indicators described in the previous section, the next step is to create a model that can predict collisions within the time horizon of interest.

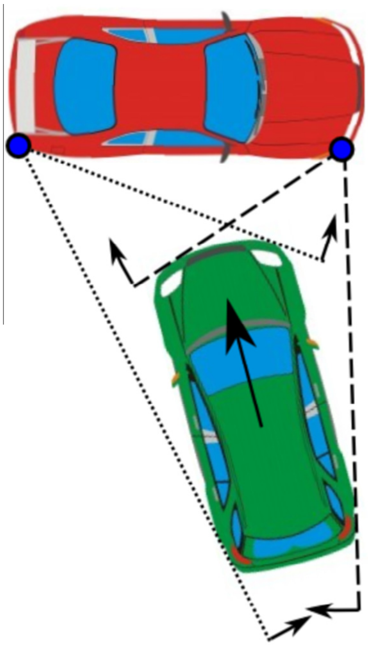


Fig. 5. Sparse loom point placement can lead to potential collisions going undetected.

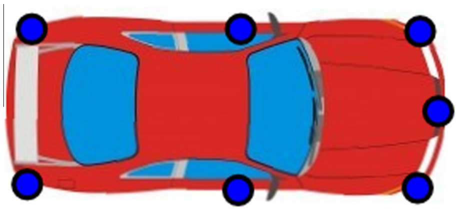
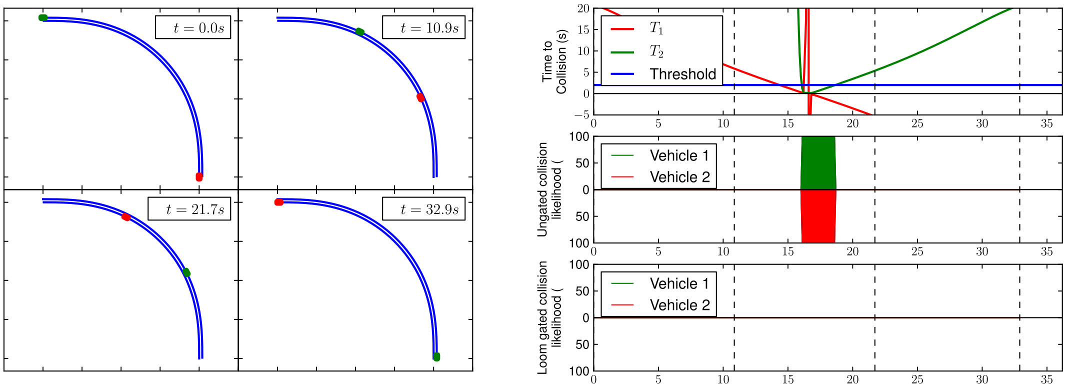


Fig. 6. Typical placement of loom test points.



(a) Trajectories.

(b) Time to collision and collision likelihoods.

Fig. 7. Constant speed passing on curved road. Without loom gating the algorithm predicts a high collision likelihood even though the vehicles pass one another safely.

In order to evaluate the predictive ability of the loom gated TTC method, various metrics were calculated when applied to empirical data. This section describes the data and the performance of the algorithm.

5.1. Experimental trajectories

A vehicle with GPS and dead reckoning was driven around the local streets surrounding the Australian Centre for Field Robotics. State information including position, speed, heading and yaw rate was logged at 10 Hz. Williams et al. (2012) shows that difference of position measurements can be as low as 1 m. Our values are lower (typically around 0.1 m) due to the way the data were collected. This it typical of the positional accuracy that will need to be demonstrated in order to make autonomous driving feasible. Fig. 8 shows the raw position data collected during this drive.

In order to create interactions for the algorithm, the state data were time shifted to create overlapping trajectories. By controlling the time shift, the trajectories could be made to pass one another clearly, pass closely or collide. It is clearly infeasible to generate trajectory data for actual collisions so time shifting trajectories was used to overcome this experimental limitation.

The full set of data used in the production of this paper is available from our research group's website at http://its.acfr.usyd.edu.au/datasets.

5.1.1. Scenarios

After collection, the data were split into segments. Interaction points were chosen at traffic lights and roundabouts. Segments were created when the vehicle drove within a 35 m radius of these intersection points and the segment concluded when the vehicle passed outside this radius. This process resulted in 18 segments around 7 intersection points. Fig. 9 shows the results of this segmentation process.

A typical sequence of data is shown in detail in Fig. 10. It can be seen that the trajectory is not smooth and has jagged movements during GPS updates of the UKF. This is important to note, as it demonstrates that the algorithm is resilient to noisy, non-smooth data as would be expected from vehicles exchanging state information over V2V communication systems.

5.2. Training data

Pairs of trajectory segments were selected at random and random time offsets were applied to their start times. The trajectory of the two vehicles was played back and the interactions of the vehicles examined. To be included in the final training set the trajectories had to satisfy the following criteria:

- At least 6 s of playback available.
- At least 3 s of playback before a collision occurs.
- Minimum separation of 30 m at beginning of playback.

If the trajectories met the above criteria they were then classified on the basis of their interactions in one of three ways. Trajectories labelled "clear" were ones where the minimum separation remained above 10 m. Trajectories labelled "close" involved minimum separation below 10 m but no collision. "Collision" trajectories involved a collision between vehicles. This was repeated until there were 100 trajectories in each of the three categories.

Once trajectories were generated and classified they were played back to generate the training dataset. In order to give the UKFs time to settle, data from first two seconds of playback were not added to the dataset. At each time step the loom rates were calculated for each vehicle at the loom points shown in Fig. 6. The TTC, T_2, and first order TTC, T_1 were also

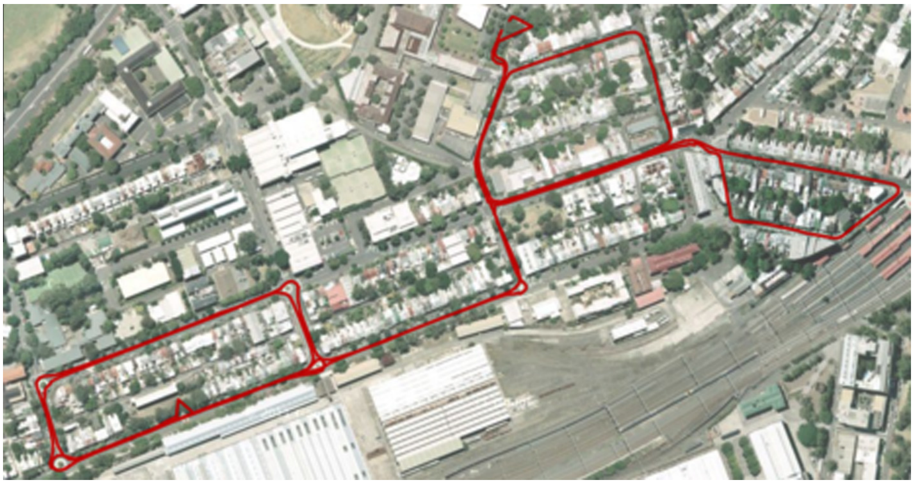


Fig. 8. Position data collected for algorithm training and testing.

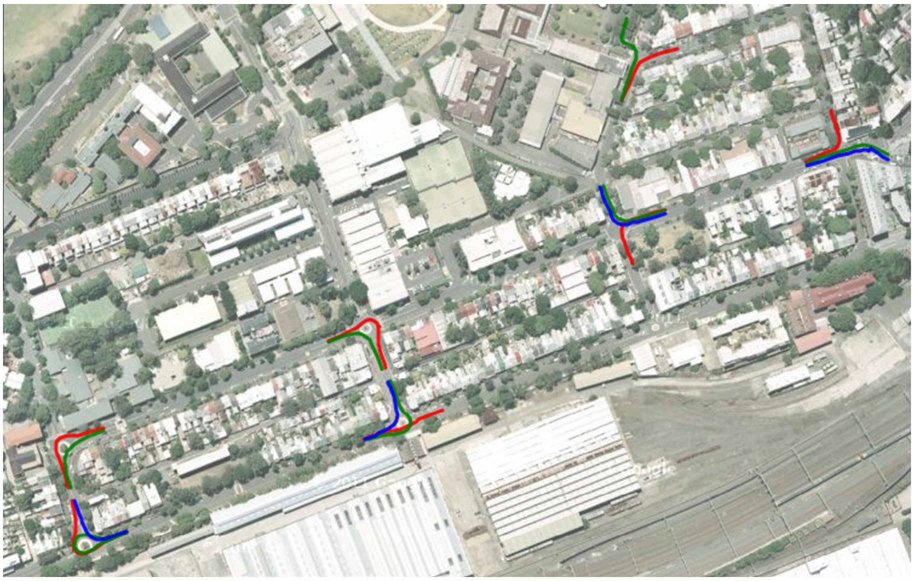


Fig. 9. Collected data split into separate tracks within 35 m of intersections.

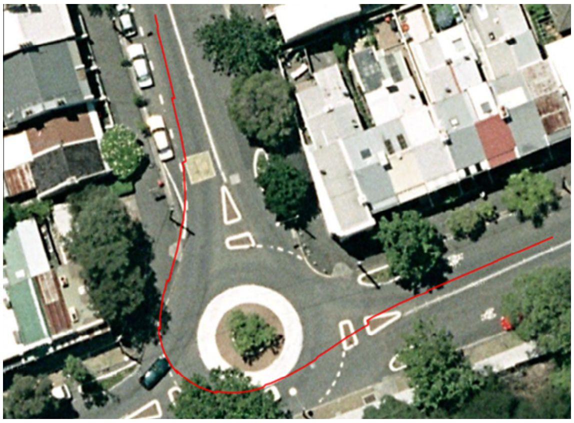


Fig. 10. Detailed view of one of the collected trajectories showing the noise in the measurements.

calculated. Finally, if a collision was going to occur at some point in the next two seconds this was also recorded. That is, the horizon of collision prediction was two seconds. At the point where the two vehicles collided the playback was stopped. This process resulted in 65,000 data points to be used for algorithm evaluation.

5.3. Performance metrics

Let *TP* and *FP* be, in that order, the number of true and false positives, and *TN* and *FN* the number of true and false negatives, respectively. These statistics are calculated on the test data after running the algorithm.

The metrics that are most important for this algorithm are precision, recall and accuracy, defined as

$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is the proportion of test points classified as leading to a collision that *actually* result in a collision. A high precision means that the classifier has high credibility, i.e. it produces few false alarms. Recall, also known as sensitivity, is the proportion of points known to result in a collision that test positive for it. A high recall means that the classifier is capable of detecting the imminence of a collision, i.e. it can provide timely warning. Accuracy is the proportion of points (both collisions and safe passages) that are correctly classified. Although high accuracy does not necessarily imply high predictive power, it is still a useful metric for visualisation.

It should be noted that selecting the algorithm on the basis of high precision will tend to make the algorithm very conservative in its predictions. As it is penalised for false positives, this selection criterion will favour algorithms that return few positive results that it is very certain about. In the most extreme case, an algorithm returning only one positive result (that was also a true positive) would have a precision of 100%. This is clearly not the desired behaviour. Alternatively, selecting an algorithm on the basis of high recall will favour an algorithm that returns positive results for values even when unsure as this will minimise false negatives. The extreme case here is that the algorithm returns a positive for all samples, leading to zero false negatives and a recall of 100%. Since precision and recall are seemingly at odds, we favour the F_1 score, defined as

$$F_1 = \frac{TP}{TP + FP/2 + FN/2}$$

The F_1 score is an increasing function of both precision and recall, and hence is more appropriate for imbalanced data, where the number of positives cases is much smaller/larger than the number of negative ones. Our data contains a large number of safe passages and only a small number of collisions, which is why we prefer this metric.

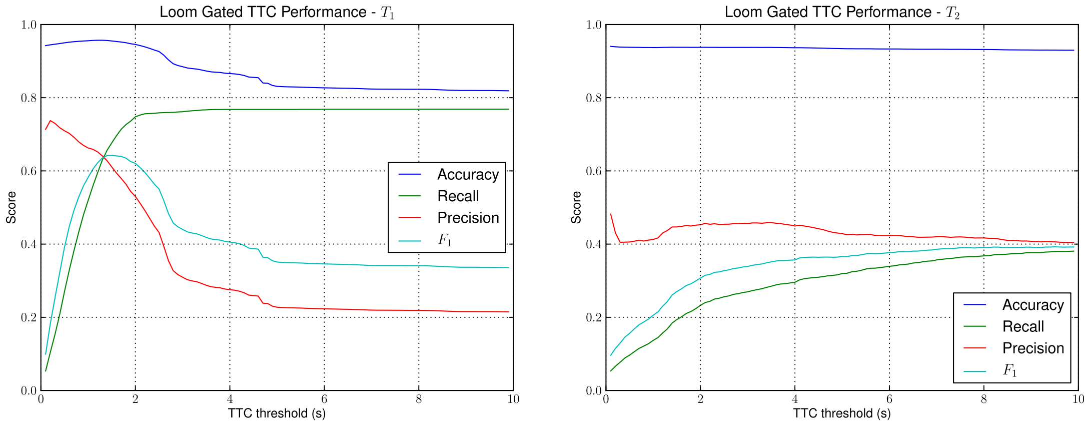
Fig. 11 show the performance of the algorithm when used with a threshold on T_1 and T_2. The threshold is varied between 0 and 10 s. Loom gating is performed as described in Section 3 – i.e. a loom pair must test positive for a collision course when considering the two loom rates at that point. In this sense, the threshold for loom rates is set at 0.

Fig. 11 shows that the method using a threshold on the first order TTC is the most effective. The F_1 score is around 0.65 when the threshold is set just below that of the prediction horizon (i.e. just below 2 s). Predicting collisions within the 2 s horizon is universally poor when using the second order TTC, with a maximum F_1 score below 0.4.

When performing collision predictions using the loom gated TTC method there is information that is discarded which could play a role in improving performance. Firstly, only one of T_1 or T_2 is used. Any correlation between the two that could be used to predict collisions is unavailable. Secondly, the loom gating provides a binary assessment of whether loom angles are moving in a particular direction or not. The magnitude of the loom rates is ignored in the algorithm, and the threshold to determine whether to gate on the basis of movement rates is set at 0. The theoretical basis for this has been laid out previously, but it is possible that the threshold could be changed and the magnitude of loom rates used to improve performance. In order to explore this possibility we provided both T_1 and T_2 plus all of the loom rates to various machine learning algorithms to produce a classifier that could perform better than a straight application of the loom gated TTC method. Upon doing so it was discovered that a Support Vector Machine performed best.

5.4. Support Vector Machines

Kernel-based Support Vector Machines (SVMs) are statistical binary classification models, i.e. they map input vectors to 0/1 outcomes. Given a set of input–output pairs, an SVM selects a subset of these pairs, called support vectors, and uses them to make decisions about previously unseen data. Theoretically well motivated and empirically shown to have good performance at generalisation, they have been successfully applied in many different fields including vehicle collision prediction



(a) First order TTC.

(b) Second order TTC.

Fig. 11. Performance metrics of loom gated TTC collision prediction versus TTC threshold. Overall, use of T_1 is better than T_2. An algorithm tuned to provide high precision would tend to set the TTC threshold very low, resulting in poor recall. Conversely, an algorithm tuned to provide high recall would perform poorly for precision. The F_1 metric provides a good compromise between the two.

Table 1
Knots for Spline Basis Functions.

Data	Knots				
TTC	$-\infty$	0 ($\times$4)	2.5	5.1	30.0	∞
Loom rates	$-\infty$	-0.157	-0.0374	0.00145	∞	

and avoidance Álvarez et al. (2010). Here, we apply them because of their flexibility (kernel SVMs are non-linear classifiers) and sparsity (the number support vectors is typically much smaller than the size of the training sample).

In order to determine collision likelihoods, we train a Support Vector Machine (SVM) on a dataset featuring TTC and loom information. The SVM was implemented by using the Scikit-learn Python package Pedregosa et al. (2011), and uses an Radial Basis Function (RBF) kernel.

5.5. Inputs

The values for T_2 and T_1 were expanded into infinite cubic Spline Basis Functions (SBFs). The knots for the SBF were set at the quartile values for the positive values of these parameters. Any values of T_2 or T_1 above 30 s was capped at 30 s as these values lie outside the horizon of prediction that we are interested in. The SBF knots are listed in Table 1.

The SBF was used so that any scaling effect of a value over the 30 s threshold for T_1 or T_2 (see Section 3.1) would not skew the SVM. The knot at $-\infty$ serves to create a step function for negative values - they are all treated the same.

An SBF was applied to loom rate values in a similar way, with the knots being chosen at the quartile values of the loom rate data. Table 1 shows these knots. It can be seen that the distribution is skewed towards negative values. This is because the vehicles used to generate the dataset were driving on the left side of the road, causing loom rates to be negative as the vehicles passed one another.

The preprocessed values were fed as inputs to the SVM. The inputs were:

- T_2 after processing with the SBF;
- T_1 after processing with the SBF;
- The 14 loom rate values corresponding to left and right loom rates at the 7 loom points, after processing with the SBF.

The signal that the SVM was being trained to predict was the flag (0 or 1) indicating whether a collision would occur within the following 2 s horizon.

5.6. Parameter estimation

The parameters of the SVM were estimated by performing a 5-fold cross-validation on the dataset. This was run for error term penalty parameter, C, values ranging between 10^2 and 10^8, and training example influence parameter, γ, values between 10^{-3} and 10^1. The F_1 performance metric found by running the cross validation are shown in Fig. 12. Fig. 13 shows various performance metrics for a vertical transect of C values with γ fixed at its optimum value of $10^{-1.0}$. We select the model that maximises F_1; from Fig. 12 the maximum of F_1 occurs for $C \simeq 1000$ and $\gamma \simeq 0.1$.

6. Collision likelihood under uncertainty

If the state information of the two vehicles was perfectly accurate then applying the SVM to the loom gated TTC values and relaying the result to the driver would be an acceptable approach. This situation does not occur in actual deployments and a method of dealing with the uncertainty of state estimates must be used. For example, consider the situation in Fig. 14. While it appears that no collision is imminent when considering the mean vehicle position from its position distribution, it is possible for the actual position to be either side of this. An error to one side would be catastrophic for the oncoming driver while an error to the other just results in increased passing clearance. We cannot say definitively which outcome will occur. Nevertheless it is important to account for the uncertainties in state estimation that are inherent in localisation problems such as this. The algorithm presented in this section is able to perform under such uncertainty.

Given that we model the behaviour of both vehicles with a UKF, we have access to the uncertainty information in the model. We perform Monte Carlo simulation and sample the state of each vehicle from the distribution provided by the UKF. In the model presented in this paper we use a sample of size 25 per vehicle[1]. The SVM returns a positive or negative result for each sample and these results are aggregated to produce a collision probability. This allows the model to provide a probability of collision rather than a binary assessment of "collision/no collision".

The loom-gated probabilistic collision prediction algorithm which accounts for uncertainty in ego and intruder state estimation is provided in Algorithm 1.

[1] Assuming that each vehicle possesses their own, independent filtering system, so that the filtered posterior distributions are independent.

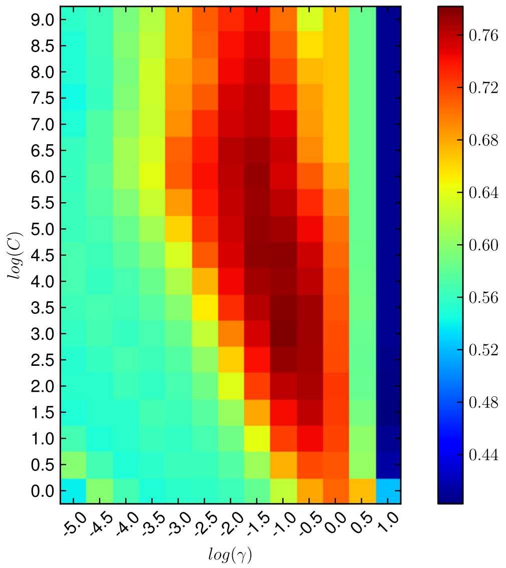


Fig. 12. SVM F_1 metric for varying error term penalty parameter, C, and influence parameter, γ. Parameters are tuned to maximise the performance metric of interest, in this case F_1. F_1 is the harmonic mean of precision and recall.

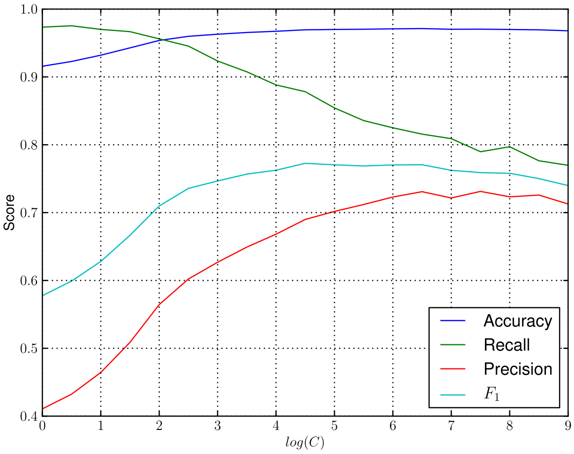


Fig. 13. SVM tuning metrics versus error term penalty parameter. γ is fixed at $10^{-1.0}$. Maximising F_1 results in a good trade-off between precision and recall.

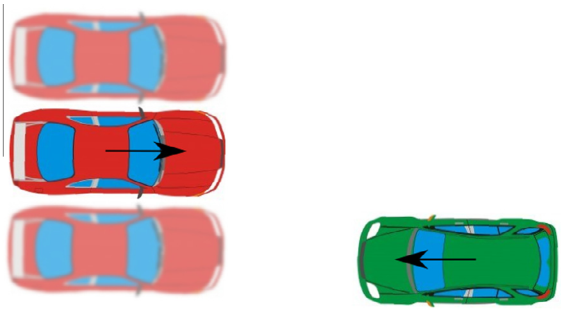


Fig. 14. The mean predicted position and samples from the position distribution. Each predicted position results in very different outcomes for the green vehicle. A useful collision prediction algorithm must be able to account for such uncertainty. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 1. Loom-gated probabilistic collision prediction algorithm

```
for samples do
    sample from UKFs: ${\bf p}, v, \psi, \omega\}_{ego}, \{{\bf p}, v, \psi, \omega\}_{intruder}$
    calculate TTC: $T_1$ and $T_2$
    calculate loom rates: $\{\dot{\alpha}_1, \ldots\}$ and $\{\dot{\beta}_1, \ldots\}$
    $inputs \leftarrow$ BasisFunction$(T_1, T_2, \{\dot{\alpha}_1, \ldots\}, \{\dot{\beta}_1, \ldots\})$
    $prediction \leftarrow$ SVM($inputs$)
    $positive\_samples \leftarrow positive\_samples + prediction$
end for
$collision\_probability \leftarrow \frac{positive\_samples}{samples}$
```

7. Experimental evaluation

In order to evaluate the usefulness of the methods described in this paper, they were run on the empirical dataset described in Section 5.1. This consisted of experimental data captured from a vehicle driving on urban roads that are reflective of those that would be encountered in real V2V operations. We used the logged data from the vehicle trial to create scenarios for testing. None of the scenarios presented in this section were used in the training of the SVM.

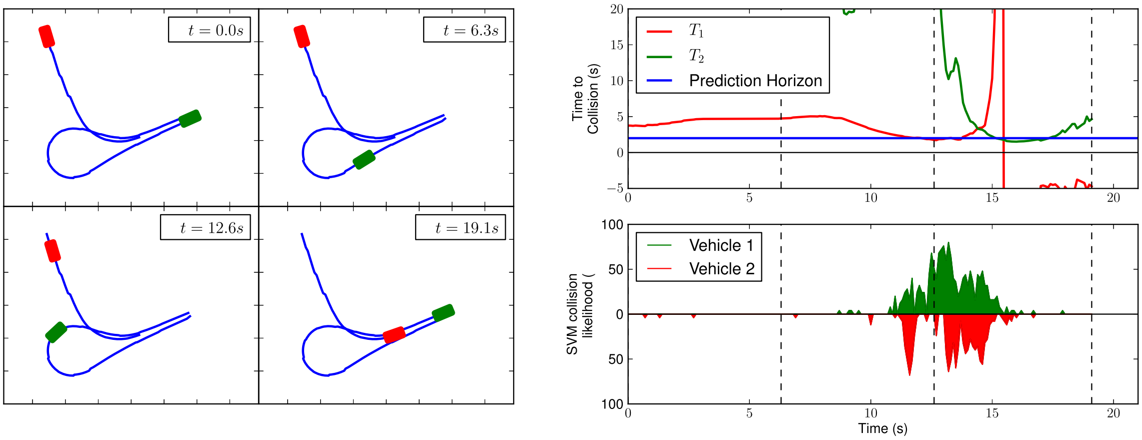
Figs. 15–17 show the same roundabout with two vehicles crossing one another's paths.

Fig. 15 shows the red vehicle entering the roundabout and executing a left turn well after the green vehicle has passed by. The vehicles are close to one another but the minimum separation never falls below 10 m. The probability of collision calculated by the SVM rises as the red vehicle approaches the roundabout where the green vehicle is leaving but the probability spends very little time above 50%. The probabilities of collision are different for each vehicle as the pose of each vehicle as seen from the other is different, and that in turn affects the magnitudes of the loom rates calculated.

The second scenario is shown in Fig. 16. In this scenario the vehicles execute the same manoeuvres but this time pass within 10 m of one another. As they reach their closest point at around $t = 12$ s the calculated collision probabilities rise sharply and remain over 50% for some time. The SVM is able to recognise that manoeuvres in close proximity have a higher likelihood of collision than those of the first scenario. As the vehicles establish a following trajectory the collision likelihood drops rapidly but remains non-zero due to the closeness of the two vehicles.

The final scenario in Fig. 17 shows the two vehicles colliding. The main points to note from this example are that the collision probability rises rapidly to above 50% where it spends a good deal of time. The probability rises approximately three seconds before the collision occurs, so it is sensitive enough to predict collisions well ahead of time.

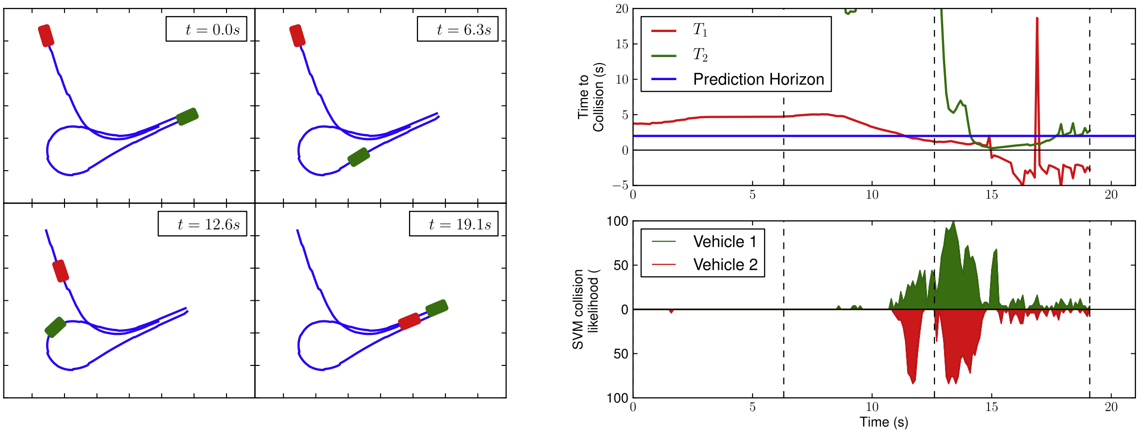
All three situations demonstrate the difference in behaviour between T_1 and T_2. As the two vehicles approach the roundabout they are both slowing down in anticipation of other traffic. This means that the closure rate is decreasing as the separation decreases. In this case the decrease in closure rate and separation are such that T_1 stays within the sub-five to zero second band. In contrast, the decrease in the separation rate creates very large values of T_2 until the vehicles are in very close



(a) Trajectories.

(b) Time to collision and collision likelihoods.

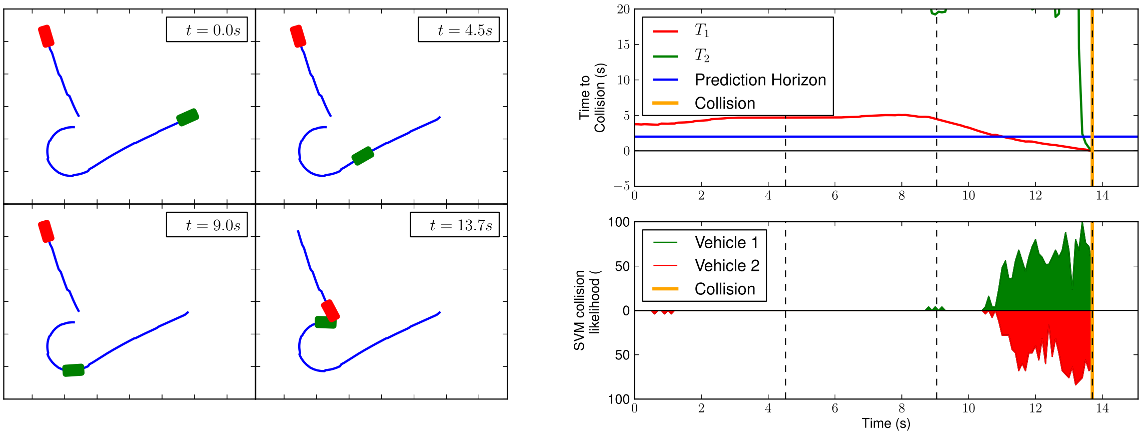
Fig. 15. Turning across paths at a roundabout. Both vehicles remain clear of one another. Collision likelihoods rise as the vehicles approach, but do not spend significant amounts of time above 50%.



(a) Trajectories.

(b) Time to collision and collision likelihoods.

Fig. 16. Turning across paths at a roundabout. Both vehicles pass closely. Collision likelihoods are higher and spend longer above the 50% probability point as the vehicles draw close.



(a) Trajectories.

(b) Time to collision and collision likelihoods.

Fig. 17. Turning across paths at a roundabout. Vehicles collide. Collision probabilities for both vehicles rise well in advance of the collision and remain elevated until the point of collision.

proximity. This is because T_2 projects the rate of change of closure rate forward in time - it assumes that the closure rate will continue to decrease linearly with time. In this particular instance, this linear extrapolation leads to predictions that the vehicles will continue to slow much more than they do, and the T_2 stays high. The difference between T_1 and T_2 values in this case is a good example of why both values are used as SVM inputs so that reliance is not placed solely on a TTC metric that does not accurately reflect the behaviour of the vehicles. There are other scenarios where the T_2 value is a better predictor of TTC than the T_1 value.

In summary, the SVM model fed with loom rate and TTC data demonstrates that:

1. the features are discriminative (they allow us to differentiate collisions from non-collisions) in all three scenarios;
2. the features generalise the TTC non-trivially and behave as we would expect them to behave from intuition (by looking at the graphs);
3. the SVM is sensitive and reacts quickly to sudden changes in vehicle velocity and acceleration;
4. the model is continuous-valued (a probability) and so it can be thresholded and/or averaged over time to create a decision rule for generating alarms.

These features make it a useful model for use in deployments on real vehicles and mean it can be used as the basis for more sophisticated processing. This will ultimately lead to a driver collision warning system that is reactive, accurate and works for all vehicle interaction scenarios.

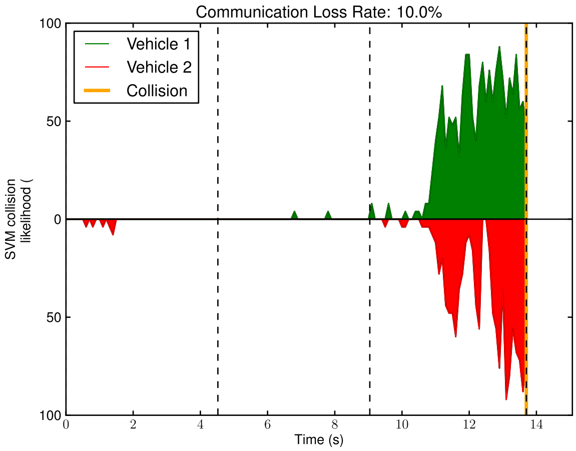


Fig. 18. Collision likelihoods. 10% V2V communication packet loss rate. The precision likelihoods become slightly choppy as the UKFs are denied updates from the other vehicle. However, the signal is still clear and well in advance of the collision.

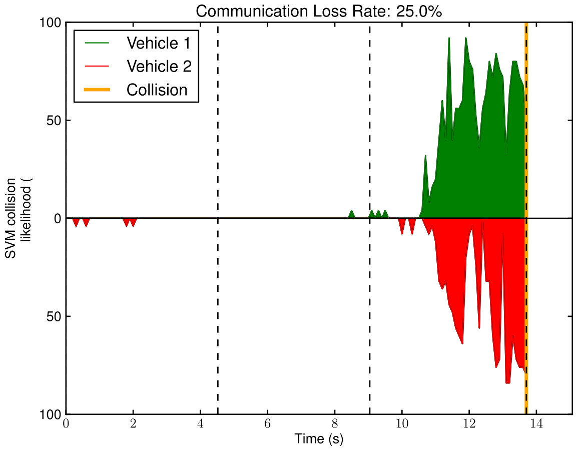


Fig. 19. Collision likelihoods. 25% V2V communication packet loss rate. Collision likelihoods do not appear much changed from the 10% loss rate. A clear signal is still visible in advance of the collision.

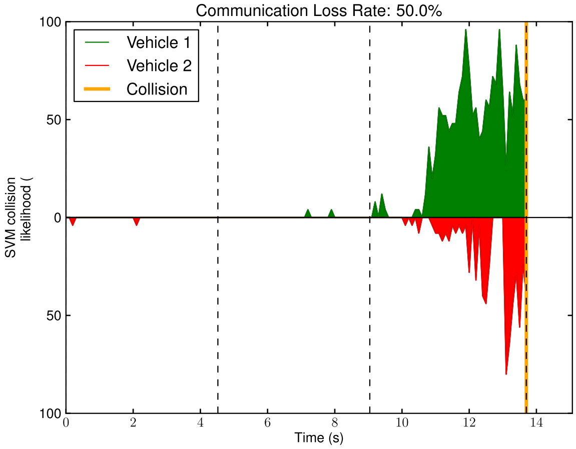


Fig. 20. Collision likelihoods. 50% V2V communication packet loss rate. Useful collision probabilities are still generated for the green vehicle, due to the relatively straight trajectory of the red vehicle. However, the constant change of heading of the green vehicle as it transits the roundabout renders collision prediction by the red vehicle almost inoperative at this level of packet loss. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

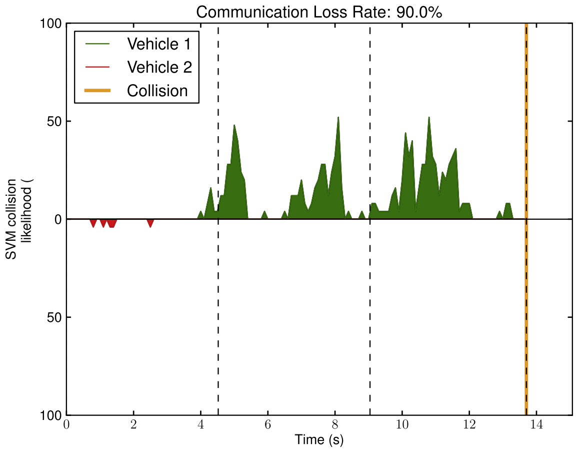


Fig. 21. Collision likelihoods. 90% V2V communication packet loss rate. Such a high rate of packet loss renders all collision prediction inoperable.

7.1. Missing communications

While current V2V systems are designed around broadcast rates of 10 Hz or higher, there is no guarantee that these broadcasts will always be received. This problem is exacerbated in urban areas with occlusions or in certain traffic conditions such as communication between vehicles on opposite sides of a large truck. This section examines the performance of the algorithm when the rate of received communications between vehicles drops.

We examine the signal produced by the SVM for the roundabout collision scenario presented in Section 7. At each time step a random number generator is used to determine whether or not to disregard the state information of the intruder vehicle, thereby introducing a failure in communication. If it is ignored, the last known speed and yaw rate information is fed to the predict step of the UKF for the intruder vehicle. The UKF for the ego vehicle continues to predict and update at 10 Hz as this information is always available - it is not passing over a communication link.

Fig. 18 shows 10% of V2V communication packets being lost. There is still a clear signal to both vehicles in advance of the collision.

Fig. 19 shows the V2V communication packets loss rate at 25%. The quality of the collision likelihood signal for the red vehicle has started to degrade. This is because the green vehicle is turning and the uncertainty of its position grows over time. The green vehicle is still able to predict the state of the red vehicle reasonably accurately as the red vehicle is maintaining a fairly constant velocity.

When the communication loss rate increases to 50% (Fig. 20) the algorithm still produces predictions of collision for both vehicles. The signal is becoming noisier as it tends to fluctuate more between minimum and maximum values. This is likely to be because the prediction becomes more accurate (i.e. higher probability) each time the UKF performs a correction step when information from the intruder vehicle is received. This happens on average at 5 Hz.

At 90% packet loss (Fig. 21, the red vehicle has essentially lost the ability to predict the trajectory of the green vehicle. No collision is predicted by the algorithm for the red vehicle, and the algorithm is not able to work effectively with this level of communication corruption. It should be noted that a packet loss rate of this magnitude results in an effective update interval of 1 s. Given that this is half of the interval before collision that we wish to generate a prediction of collision, it is not surprising that the algorithm cannot perform as expected in the ideal case of perfect communication.

The SVM approach to collision prediction is robust to communications loss between vehicles. It is only when loss rates exceed 50% that the performance of the algorithm begins to degrade. One would not expect to see such large loss rates in a V2V communication system.

8. Conclusion

This paper has presented a new method for predicting the collision likelihood between two vehicles communicating over a V2V link. We have laid out three major contributions to the problem of vehicle collision prediction. Firstly, we have presented a method for generalising Time to Collision calculations to unconstrained vehicle motion. The method does not require that the interaction be constrained in any way - it works for all general traffic scenarios. Secondly, we have demonstrated a technique for gating TTC calculations based on the relative motion of the vehicles involved – whether the intruder vehicle is "looming" or not. The method for determining looming and its theoretical basis were provided. Finally, these approaches are integrated and modified to account for uncertainties in vehicle states that are inherent in any real world deployment of V2V systems. It is robust to uncertainty in the state measurements for the vehicles, and is robust to lost communications. This makes it a comprehensive method for collision likelihood prediction and thus a very good basis for the development of more sophisticated algorithms, ultimately leading to driver aids that can cope with the realities of V2V communications and sensing uncertainty. A future line of research that we are pursuing is to develop higher level algorithms that can take the predictions of the method presented here and turn the signals into meaningful warnings to the driver in order to avert collisions. To the best of our knowledge this is the first time that collision prediction, uncertainty in position and communication loss have been integrated into a consistent, formal framework and verified on experimental data.

Acknowledgements

This work is supported by Australian Research Council Linkage Project LP120100700.

References

Alexander, P., Haley, D., Grant, A., 2011. Cooperative intelligent transport systems: 5.9-GHz field trials. IEEE Proc. 99 (7), 1213–1235.

Álvarez, S., Sotelo, M.A., Ocaña, M., Llorca, D.F., Parra, I., Bergasa, L.M., 2010. Perception advances in outdoor vehicle detection for automatic cruise control. Robotica 28, 765–779, URL http://journals.cambridge.org/article_S0263574709990464.

Archer, J., 2005. Indicators for Traffic Safety Assessment and Prediction and Their Application in Micro-simulation Modelling: A Study of Urban and Suburban Intersections (Ph.D. thesis). Karlstad University.

Dabbour, E., Easa, S., 2014. Proposed collision warning system for right-turning vehicles at two-way stop-controlled rural intersections. Transp. Res. Part C: Emerg. Technol. 42 (0), 121–131, URL http://www.sciencedirect.com/science/article/pii/S0968090X14000618.

Dagan, E., Mano, O., Stein, G.P., Shashua, A., 2004. Forward collision warning with a single camera. In: Intelligent Vehicles Symposium, 2004 IEEE. IEEE, pp. 37–42.

Ge, J.I., Orosz, G., 2014. Dynamics of connected vehicle systems with delayed acceleration feedback. Transp. Res. Part C: Emerg. Technol. 46 (0), 46–64, URL: http://www.sciencedirect.com/science/article/pii/S0968090X14001119.

Guler, S.I., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transp. Res. Part C: Emerg. Technol. 46 (0), 121–131, URL http://www.sciencedirect.com/science/article/pii/S0968090X14001211.

Hafner, M., Cunningham, D., Caminiti, L., Del Vecchio, D., 2013. Cooperative collision avoidance at intersections: Algorithms and experiments. IEEE Trans. Intell. Transp. Syst. 14 (3), 1162–1175.

Hayward, J.C., 1972. Near-miss determination through use of a scale of danger. Highway Res. Rec. (384).

Huang, C.-M., Lin, S.-Y., Yang, C.-C., Chou, C.-H., 2009. A collision pre-warning algorithm based on V2V communication. In: Proceedings of the 4th International Conference on Ubiquitous Information Technologies & Applications, 2009. ICUT'09. IEEE, pp. 1–6.

Julier, S., Uhlmann, J., 1997. A new extension of the Kalman filter to nonlinear systems. In: International Symposium on Aerospace and Defense Sensing, Simulation and Control.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Sepulcre, M., Gozalvez, J., Hernandez, J., 2013. Cooperative vehicle-to-vehicle active safety testing under challenging conditions. Transp. Res. Part C: Emerg. Technol. 26 (0), 233–255, URL http://www.sciencedirect.com/science/article/pii/S0968090X12001258.

Stiller, C., Farber, G., Kammel, S., 2007. Cooperative cognitive automobiles. In: Intelligent Vehicles Symposium, 2007 IEEE. IEEE, pp. 215–220.

Stubing, H., Bechler, M., Heussner, D., May, T., Radusch, I., Rechner, H., Vogel, P., 2010. simTD: A car-to-X system architecture for field operational tests. IEEE Commun. Mag. 48 (5), 148–154.

Svensson, A., 1998. A Method for Analysing the Traffic Process in a Safety Perspective. Lund Institute of Technology.

Williams, T., Alves, P., Lachapelle, G., Basnayake, C., 2012. Evaluation of GPS-based methods of relative positioning for automotive safety applications. Transp. Res. Part C: Emerg. Technol. 23 (0), 98–108, Data Management in Vehicular Networks. URL:http://www.sciencedirect.com/science/article/pii/S0968090X11001215.

Worrall, S., Agamennoni, G., Nieto, J., Nebot, E., 2012. A context-based approach to vehicle behavior prediction. IEEE Intell. Transp. Syst. Mag. 4 (3), 32–44, Fall.