2014 IEEE Intelligent Vehicles Symposium (IV)
June 8-11, 2014. Dearborn, Michigan, USA

Vehicle Collision Probability Calculation for General Traffic Scenarios Under Uncertainty

James Ward, Gabriel Agamennoni, Stewart Worrall and Eduardo Nebot

***Abstract*— Vehicle-to-vehicle (V2V) 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.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 for collision probability calculation that addresses these problems.**

I. 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. In particular, there have been two special issues in the ITS Journal covering cooperative driving challenges and emerging cooperative technologies in the field of intelligent transportation systems [10], [12]. A number of trials of the communications technology have been conducted around the world [13], [2]. These trials tend to focus on lower level issues around reliability of connectivity, bandwidth or reliability. Although the trials generally use the transmitted information to evaluate whether the communication systems provide sufficient warning to avert conflict [5], [15], 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 associated with the method used to ascertain that information. A method to exploit this information must be able to deal with this uncertainty.

II. Time To Collision

Time to Collision (TTC) is a proximal safety indicator [3], [14] defined by Hayward [6] as "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.

III. Existing Techniques

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 real time calculation of TTC. The *MobileEye* system [4] is an example of such an implementation. A technique for alerting drivers to potential collision was developed by Huang et al [7] 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.

All authors are with the Intelligent Vehicles and Safety Systems Group at The Australian Centre for Field Robotics, The University of Sydney, NSW 2006, Australia (emails: {j.ward, g.agamennoni, s.worrall, nebot}@acfr.usyd.edu.au)

IV. Vehicle State Information

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 (°); and yaw rate, ω (°/s).

We model the movement of vehicles using an Unscented Kalman Filter (UKF) [8] and thus we also have the uncertainty in the measurements of these states available as well. 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 [1] 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. GNSS derived position information is used for UKF update steps. If communications between vehicles is lost, predict steps are continued with the most recent speed and yaw rate. When communications are established, the UKF is reinitialised with the most recent state transmitted by the intruder vehicle.

V. 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 moment 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.

A. Time-To-Collision in the Plane

Studies using the Time To Collision metric often treat the movement of the vehicles as a 1D manifold problem. That is, the calculation is performed on a one dimensional manifold, such as the position along the road that the vehicles are

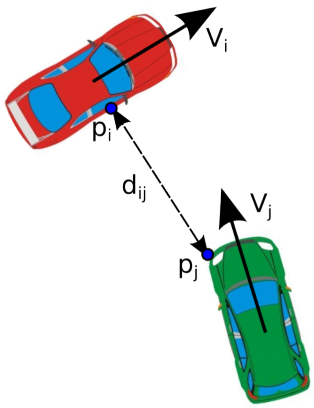


Fig. 1: Definition of vehicle state parameters. The position of each vehicle is the point on each vehicle closest to the other.

driving along. 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. Approaches such as [11] apply to intersections, but have reduced the problem to a 1D manifold by fitting the vehicle movement to a ”best fit“ curve. 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. It cannot always be assumed that the vehicle will follow the lanes of the road for example. 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.

B. 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. 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. The definitions are illustrated in Fig. 1.

An assumption is made that the vehicles move at an almost constant velocity, so that the accelerations $\mathbf{a}_i$ and $\mathbf{a}_j$ are sufficiently small to be omitted. Under this assumption the separation, d_{ij}, between vehicles i and j and its first and second derivatives can be computed as follows:

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

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:

$$
\begin{aligned}
d_{ij} + \dot{d}_{ij} T_1 &= 0 \\
T_1 &= -\frac{d_{ij}}{\dot{d}_{ij}}
\end{aligned}
\tag{1}
$$

The second order case:

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

Equation 2 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:

$$
T_2 = \begin{cases}
T_1 & \text{if } \ddot{d}_{ij} = 0 \\
\frac{\dot{d}_{ij}}{\ddot{d}_{ij}} & \text{if } \Delta < 0 \\
\min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) & \text{if } \min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) \geq 0 \\
\max\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) & \text{if } \min\left(\frac{-\dot{d}_{ij} \pm \sqrt{\Delta}}{\ddot{d}_{ij}}\right) < 0
\end{cases}
\tag{3}
$$

It will be noted in (1) and (3) 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.

VI. 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

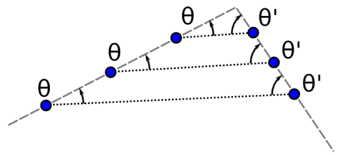


Fig. 2: Relative bearing of the other object remains constant when on a collision course.

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.

A. Limitations of Planar TTC

The calculations described in Section V-A 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 seconds, 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.

B. 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

$$\dot{\beta} \leq 0 \wedge \dot{\alpha} \geq 0 \implies \exists \theta : \beta \leq \theta \leq \alpha \wedge \dot{\theta} = 0 \tag{4}$$

If the leftmost point of the object is moving anticlockwise relative to the observer (i.e. $\dot{\alpha} > 0$) and the rightmost point

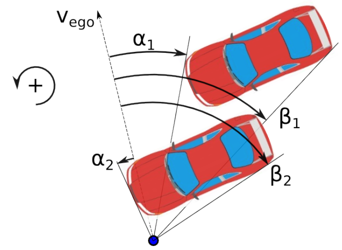


Fig. 3: Behaviour of relative bearings of leftmost and rightmost points of intruder vehicle.

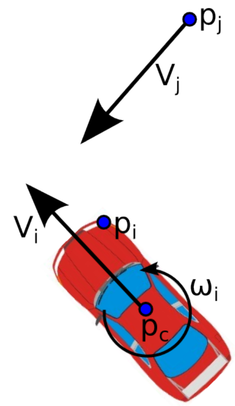


Fig. 4: Calculation of loom rates.

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 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, $\boldsymbol{\omega}_i$:

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

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, $\dot{\theta}$:

$$\dot{\theta} = \frac{(\mathbf{p}_j - \mathbf{p}_i) \times \bar{\mathbf{v}}_i + (\mathbf{p}_j - \mathbf{p}_i) \times \mathbf{v}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|^2} \tag{5}$$

Equation 5 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. Equation 5 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, $\mathbf{v}_j$.

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

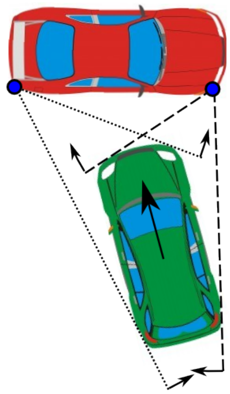


Fig. 5: Sparse loom point placement can lead to potential collisions going undetected.

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*. 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.

By only considering the TTC when Equation 4 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*.

VII. 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.

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.

A. Experimental Trajectories

A vehicle with GPS fused with 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. Typical uncertainties were around 1 metre for positions and 2 degrees for headings.

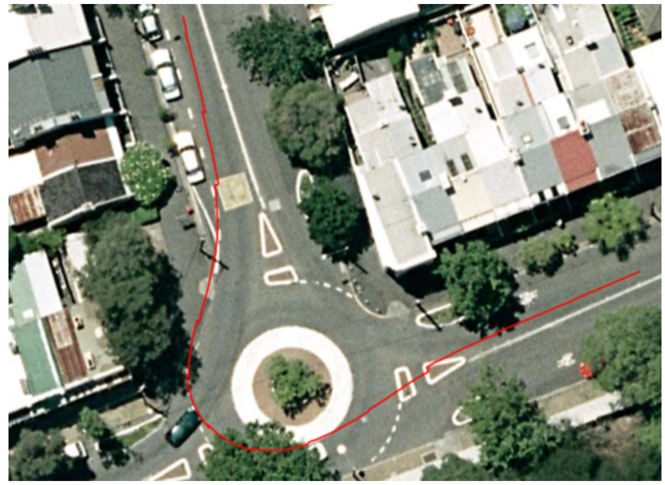


Fig. 6: Detailed view of one of the collected trajectories showing the noise in the measurements.

In order to create interactions for the algorithm, the state data were time shifted to create overlapping trajectories. By controlling the time shift, the different trajectory segments 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 different trajectory segments was used to overcome this experimental limitation.

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.

A typical sequence of data is shown in detail in Fig. 6. 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.

B. 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: there was at least 6 s of playback available, at least 3 s of playback before a collision occured and a 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 a variety of loom points placed around the vehicle perimeter. The TTC, T_2, and first order TTC, T_1 were also 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.

C. Performance Metrics

When building statistical models a number of metrics can be calculated to explore the performance of the model. 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.

We favour the F_1 score which is the harmonic mean of precision and recall. 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 positive 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.

We examined the performance of the algorithm when used with a threshold on T_1 and T_2. The TTC threshold was varied between 0 and 10 seconds. Values below the threshold would result in a positive collision prediction. Loom gating was performed as described in Section V - 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 rad/s.

Results showed that the method using a threshold on the first order TTC was 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 seconds). Predicting collisions within the 2 second 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 (Support Vector Machines and Logistic Regressors) 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.

D. 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. 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 [9], and uses an Radial Basis Function (RBF) kernel.

E. 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 seconds was capped at 30 seconds as these values lie outside the horizon of prediction that we are interested in.

The SBF was used so that any scaling effect of a value over the 30 second threshold for T_1 or T_2 (see Section V-A) would not skew the SVM.

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.

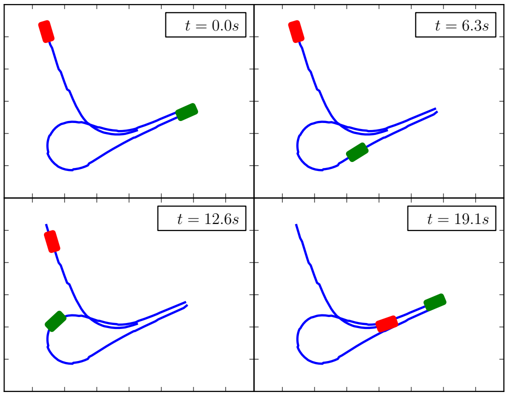
The preprocessed values were fed as inputs to the SVM. The inputs were: T_2 after processing with the SBF, T_1 after processing with the SBF and the 14 loom rate values corresponding to left and right loom rates at the 7 loom points, after processing with the SBF.

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 second horizon.

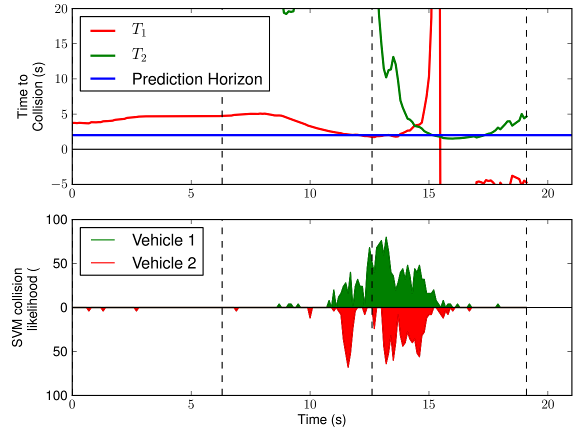
VIII. Collision Likelihood Under Uncertainty

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. 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 (i.e. position, velocity and yaw rate) 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 at each time step[1]. The SVM returns a positive or negative result for each sample and these results are aggregated to produce a collision probability. This

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

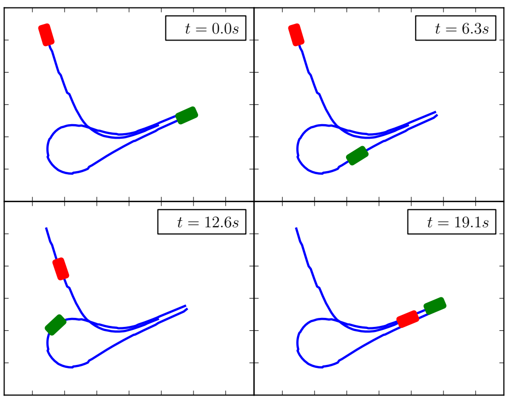


(a) Trajectories.

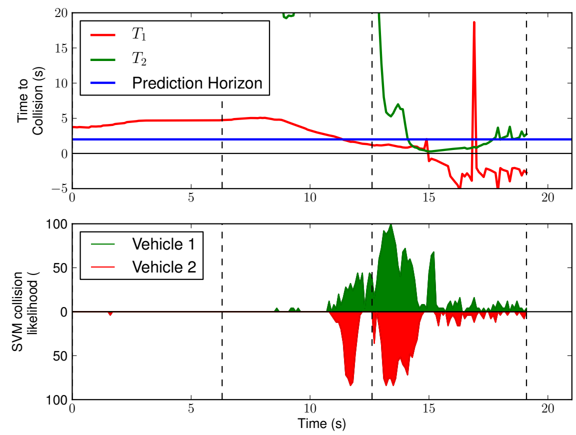


(b) Time to collision and collision likelihoods.

Fig. 7: Turning across paths at a roundabout. Both vehicles remain clear of one another.

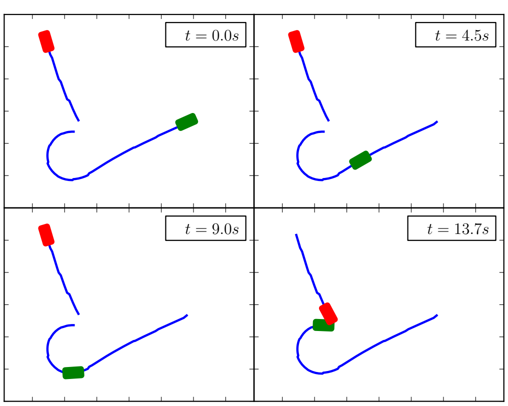


(a) Trajectories.

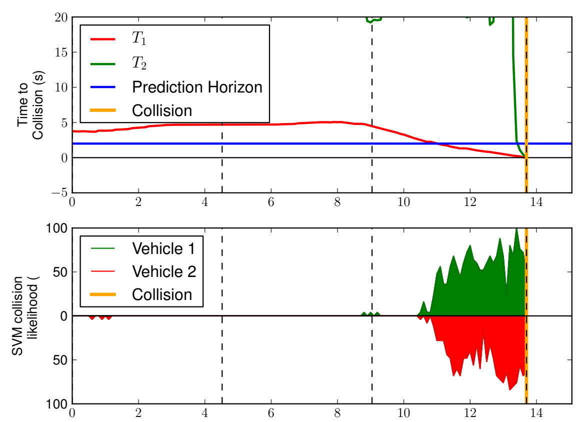


(b) Time to collision and collision likelihoods.

Fig. 8: Turning across paths at a roundabout. Both vehicles pass closely.



(a) Trajectories.



(b) Time to collision and collision likelihoods.

Fig. 9: Turning across paths at a roundabout. Vehicles collide.

allows the model to provide a probability of collision rather than a binary assessment of "collision/no collision".

IX. Performance

In order to evaluate the usefulness of the methods described in this paper, they were run on the empirical dataset described in Section VII-A. The point of the signals generated by this techniques is to provide a likelihood of imminent collision and not to provide an alarm for the driver in the form of an Advanced Driver Assistance System (ADAS). We submit the signals shown here could be used by an ADAS but further processing would be required before they were suitable to be conveyed to the driver. 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.

Fig. 7, 8 and 9 show the same roundabout with two vehicles crossing one another's paths.

Fig. 7 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 metres. 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. 8. In this scenario the vehicles execute the same manoeuvres but this time pass within 10 metres of one another. As they reach their closest point at around $t = 12s$ 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. 9 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.

In summary, the SVM model fed with loom rate and TTC data demonstrates that: the features are discriminative (they allow us to differentiate collisions from non-collisions) in all three scenarios; the features generalise the TTC non-trivially and behave as we would expect them to behave from intuition (by looking at the graphs); the SVM is sensitive and reacts quickly to sudden changes in vehicle velocity and acceleration; and 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.

X. Conclusion

This paper has presented a new method for predicting the collision likelihood between two vehicles communicating over a V2V link. The approach does not require that the method of interaction be constrained in any way - it works for all general traffic scenarios. It is robust to uncertainty in the state measurements for the vehicles, and is robust to lost communications. This makes it 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.

References

[1] *Dedicated Short Range Communications (DSRC) Message Set Dictionary*, SAE J2735, 2009.

[2] Paul Alexander, David Haley, and Alex Grant. Cooperative intelligent transport systems: 5.9-ghz field trials. *Proceedings of the IEEE*, 99(7):1213–1235, 2011.

[3] J. Archer. *Indicators for traffic safety assessment and prediction and their application in micro-simulation modelling: A study of urban and suburban intersections*. PhD thesis, Karlstad University, 2005.

[4] Erez Dagan, Ofer Mano, Gideon P Stein, and Amnon Shashua. Forward collision warning with a single camera. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 37–42. IEEE, 2004.

[5] M.R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio. Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1162–1175, 2013.

[6] John C Hayward. Near-miss determination through use of a scale of danger. *Highway Research Record*, (384), 1972.

[7] Chung-Ming Huang, Shih-Yang Lin, Chia-Ching Yang, and Chih-Hsun Chou. A collision pre-warning algorithm based on v2v communication. In *Ubiquitous Information Technologies & Applications, 2009. ICUT'09. Proceedings of the 4th International Conference on*, pages 1–6. IEEE, 2009.

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

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] J. Ploeg, S. Shladover, H. Nijmeijer, and N. van de Wouw. Introduction to the special issue on the 2011 grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):989–993, 2012.

[11] Amir Sobhani, William Young, Sareh Bahrololoom, and Majid Sarvi. Calculating time-to-collision for analysing right turning behaviour at signalised intersections. *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, 22(3):49, 2013.

[12] M.A. Sotelo, J. W C Van Lint, U. Nunes, L.B. Vlacic, and M. Chowdhury. Introduction to the special issue on emergent cooperative technologies in intelligent transportation systems. *Intelligent Transportation Systems, IEEE Transactions on*, 13(1):1–5, 2012.

[13] Hagen Stubing, Marc Bechler, Dieter Heussner, Thomas May, Ilja Radusch, Horst Rechner, and Peter Vogel. simTD: a car-to-X system architecture for field operational tests. *Communications Magazine, IEEE*, 48(5):148–154, 2010.

[14] Ase Svensson. *A method for analysing the traffic process in a safety perspective*. Lund Institute of Technology, 1998.

[15] Stewart Worrall, Gabriel Agamennoni, Juan Nieto, and Eduardo Nebot. A context-based approach to vehicle behavior prediction. *Intelligent Transportation Systems Magazine, IEEE*, 4(3):32–44, 2012.