MAP working paper

Tanvi Jindal, Yili Luo

1 Introduction

Automobiles are arguably the most popular and important mode of transportation, with over 270 million vehicles registered in the United States alone. Data from National Highway Traffic Safety Administration (NHTSA), the Centers for Disease Control and Prevention (CDC), and the Federal Motor Carrier Safety Administration (FMCSA) show that car crashes are the leading cause of death for young people between ages 15 and 29, and more than 30,000 Americans die each year in vehicle accidents (Beltz, 2018). To better understand vehicular collisions, scholars have developed numerous methods for measuring collision risk. Among the most popular are time-based methods, which are designed to capture how near or far a particular vehicle is from collision at any instant. In this manuscript we focus on time-based measures, beggining with a review of previous work, then proposing a new metric for general two-dimensional scenarios, and concluding with an application of our metric to data from a driving simulator experiment assessing vehicle-to-vehicle alert characteristics.

One of the most widely used methods of assessing collision risk is Time-To-Collision (TTC), which measures the "time required for two vehicles to collide if they continue at their present speeds and on the same path" (Hayward, 1972). Since its inception, many extensions and refinements to TTC have been proposed. Brown (2005) adjusted the measurement in one-dimensional scenarios using the velocity at collision to resolve the limitation of TTC being zero at collision regardless of severity. Hou et al. (2014) developed algorithms for calculating TTC in two-dimensional freeway settings, using different geometric shapes to represent the driver's vehicle and a buffer area around the vehicle to match the driver's perception on collisions. Sobhani et al. (2012) studied right turning conflicts at intersections by grouping drivers by behavior and calculating TTC separately for each group. Nonetheless, these two-dimensional applications of TTC are limited to specific assumptions and scenarios, as a major limitation of TTC is that its values goes to infinity for non-collision trajectories. Circumventing this limitation without imposing restrictive scenario-specific assumptions is difficult, and to our knowledge there has not been a TTC-like collision measure that can be applied uniformaly to two-dimensional settings.

An alternative to TTC, developed by Schwarz (2014), is Time of Closest Approach (TCA) measurement. Instead of relying upon intersecting vehicle trajectories, TCA is based upon the distance between vehicles as a function of time, with TCA representing the time at which the minimum distance occurs. By using the distance between the vehicles, TCA avoids the propensity of TTC to yield infiinite values when vehicle trajectories are disjoint. However, TCA cannot be used to uniformly compare across different scenarios as the implications of "minimum distance" are situation-dependent. Inspired by Brown's adjusted TTC, we decide to modify the calculation of TCA to produce a generalizable, time-based method of quantifying collisions in two-dimensions. Our approach estimates vehicles trajectories and calculates the degree by which the trajectories would need to be shifted in order for the vehicles to collide, with smaller shifts indicating the original trajectory was closer to collision than larger shifts. This provides a robust measure that can be generally applied to two-dimensional scenarios to assess safety or collision severity.

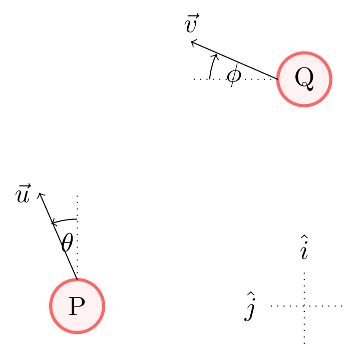
To illustrate the usefulness of our metric, we apply it data from a driving simulator experiment collected by the National Advanced Driving Simulator (NADS), using simulator inputs to quantify collision proximity at a two-dimensional intersection crossing where drivers must react to avoid colliding with a vehicle running a red light. In our application, we explore the effectiveness of using our measure to predict collision prior to impact, as well as potential advantages in statistical power that result from using our metric to retrospectively analyze experimental data.

2 Methodology

In this section we develop a metric for quantifying the proximity of two vehicles to collision. We begin with TCA as expressed by Schwarz (2014), and add complexity to better mirror real world scenarios. Then, we adjust the calculation of TCA by introducing vehicle-specific latent variables, which are used to measure proximity to collision. Our development first considers vehicles as points, and later extends the method to consider vehicles as rectangles.

2.1 Point collisions

We first start by treating vehicles P and Q as point objects, introducing our notation using the diagram below:



Here, $\hat{i}$ and $\hat{j}$ are axes of the vector plane, and $\vec{u}$ and $\vec{v}$ are the velocities of P and Q respectively. θ is the angle of rotation of $\vec{u}$ at point P from $\hat{i}$, and ϕ is the angle of rotation of $\vec{v}$ at point Q from $\hat{j}$, as illustrated. Notice, $\vec{u} = u_0 \cos(\theta)\hat{i} + u_0 \sin(\theta)\hat{j}$, and $\vec{v} = -v_0 \sin(\phi)\hat{i} + v_0 \cos(\phi)\hat{j}$.

For the purpose of this paper, we define anti-clockwise rotations to be positive and clockwise rotations to be negative. Therefore, with respect to illustration above, the angle θ is positive, and the angle ϕ is negative.

To calculate TCA for points P and Q, we express the displacement from P to Q as a function of time. The value of time which minimizes this function is the TCA of P and Q. Recognize that if P and Q collide with each other, their displacement at TCA will be 0. We express P and Q at time t, using the following notation:

$$P(t) = \begin{pmatrix} f_{\hat{i}}(t) \\ f_{\hat{j}}(t) \end{pmatrix}$$

where $f_{\hat{i}}(t) = \int_0^t u_x(t) \cdot dt$ and $f_{\hat{j}}(t) = \int_0^t u_y(t) \cdot dt$ where $u_x(t)$ and $u_y(t)$ are the $\hat{i}$ and $\hat{j}$ components of point P's velocity, and

$$Q(t) = \begin{pmatrix} g_{\hat{i}}(t) \\ g_{\hat{j}}(t) \end{pmatrix}$$

where $g_{\hat{i}}(t) = \int_0^t v_x(t) \cdot dt$ and $g_{\hat{j}}(t) = \int_0^t v_y(t) \cdot dt$ where $v_x(t)$ and $v_y(t)$ are the $\hat{i}$ and $\hat{j}$ components of point Q's velocity.

Thus the distance between P and Q at time t can be expressed as $\vec{d}(t) = P(t) - Q(t)$, which can be used to define the time to closest approach, $TCA = \text{argmin}_t\{\vec{d}(t)\}$.

2.1.1 Simple model

The simple model, as directly adopted from Schwarz (2014), assumes the positions of P and Q at time t can be characterized by their initial positions P_0 and Q_0, and the constant velocity vectors $\vec{u}$ and $\vec{v}$:

$$P(t) = P_0 + t\vec{u}$$

$$Q(t) = Q_0 + t\vec{v}$$

The distance, $\vec{d}(t)$, is then given by:

$$\begin{aligned}\vec{d}(t) &= P(t) - Q(t)\\ &= P_0 - Q_0 + (\vec{u} - \vec{v})t\\ &= s + (\vec{u} - \vec{v})t\end{aligned}$$

where $s = P_0 - Q_0$.

TCA occurs when the magnitude of $\vec{d}(t)$ is minimum, requiring $d(t) = \sqrt{\vec{d}(t) \cdot \vec{d}(t)}$ to be minimized with respect to time.

$$\frac{d}{dt}d(t) = \frac{(\vec{u} - \vec{v}) \cdot (\vec{u} - \vec{v})t + 2s(\vec{u} - \vec{v})}{\sqrt{(\vec{u} - \vec{v}) \cdot (\vec{u} - \vec{v})t^2 + 2s(\vec{u} - \vec{v})t + s^2}}$$

Equating the derivative with provides a closed-form expression for the time to closest approach.

$$TCA = \frac{-s_0}{|\vec{u} - \vec{v}|^2} \tag{1}$$

2.1.2 Acceleration and Turning

The simple model enables a straightfoward calculation of TCA, requiring only the three inputs shown in Equation (1). However, this calculation is unable to reflect the important driver behaviors of braking, accelerating, and steering.

To modify the calculation to allow for variable velocity, we define the acceleration of P as $\vec{a_P} = a_{P0}\cos(\theta)\hat{i} + a_{P0}\sin(\theta)\hat{j}$, and define the acceleration of Q similarly. Incorperating acceleration into component of velocity yields:

$$\begin{aligned}\vec{u}(t) &= (u_0 + a_{P0}t)\cos(\theta)i + (u_0 + a_{P0}t)\sin(\theta)\hat{j}\\ \vec{v}(t) &= -(v_0 + a_{Q0}t)\sin(\phi)\hat{i} + (v_0 + a_{Q0}t)\cos(\phi)\hat{j}\end{aligned}$$

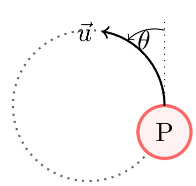
To further modify the calculation to allow for turning we use the steering wheel angle to determine change in vehicle heading using the steering ratio. Because turn of the wheel is the angle by which the vehicle turns at every instant, it can be viewed as the derivative of a vehicle's angle of motion. Let turn of wheel of point P be ω_P, then:

$$\omega_P = \frac{d\theta}{dt} \tag{2}$$

We assume ω_P is constant, so upon integrating both sides of Equation (2) with respect to t, we get the angle of motion as a function of time as follows:

$$\theta(t) = \omega_P \cdot t + \theta_0$$

where θ_0 is the angle at which P was turned at time $t = 0$.



As the illustration indicates, given a fixed steering wheel angle, point P travels with velocity $\vec{u}$ in a circular motion. We point out that in most applications a complete revolution will not occur, and only the motion in the first quarter of the circle is relavent to the calculation.

2.1.3 Standardized Degree of Collision Avoidance (SDCA)

In the event that two vehicles do not collide, TCA indicates the time at which the distance between vehicles will be minimized based upon the set of driver inputs. However, it does not directly quantify the proximity of the two vehicles to a crash. Here, we define a *reaction* as a fixed set of inputs. We use this term because it might of particular interest to calculate TCA using vehicle trajectories upon having just realized the threat of collision. The basis of our metric, Standardized Degree of Collision Avoidance (SDCA), is to measure how delayed a driver's reaction would need to be in order for the scenario to result in a collision.

To approach this task, we begin by augmenting the expressions of $P(t)$ and $Q(t)$ by introducing two latent variables Z_P, which describes the reaction delay for P, and Z_Q, which describes the reaction delay for Q. This results in the adjusted positions functions $\tilde{P}(t)$ and $\tilde{P}(t)$:

$$\tilde{P}(t) = \begin{pmatrix} f_{\hat{i}}(t) + u_0 \cdot \cos(\theta_0) \cdot Z_P \\ f_{\hat{j}}(t) + u_0 \cdot \sin(\theta_0) \cdot Z_P \end{pmatrix}$$

$$\tilde{Q}(t) = \begin{pmatrix} g_{\hat{i}}(t) - v_0 \cdot \sin(\phi_0) \cdot Z_Q \\ g_{\hat{j}}(t) + v_0 \cdot \cos(\phi_0) \cdot Z_Q \end{pmatrix}$$

Equating, $\tilde{P}(t) = \tilde{Q}(t)$ for all t$\in \mathbb{R}$, and solving for Z_P and Z_Q yields the combinations of reaction delays by P and Q that would lead to a collision. Thus, larger values of Z_P and Z_Q indicate a greater degree by which a collision was avoided. We next seek to summarize the combinations of Z_P and Z_Q to provide a single overall measure of collision avoidance.

Setting $\tilde{P}(t) = \tilde{Q}(t)$ implies that Z_P and Z_Q can be expressed functions of time, with each value of t corresponding to a pair $(Z_P(t), Z_Q(t))$. The pairs $(Z_P(t), Z_Q(t))$ can be viewed as a curve indexed by time, summarizing this curve amounts to summarizing all combinations of reaction delays that would have resulted in a collision. To summarize the curve, we calculate the expected distance, which we define as SDCA, from $(0, 0)$ to the pairs $(Z_P(t), Z_Q(t))$.

$$\text{SDCA} = \frac{\int_{t_A}^{t_B} \sqrt{(Z_P(t) - 0)^2 + (Z_Q(t) - 0)^2} \cdot \sqrt{Z'_P(t)^2 + Z'_Q(t)^2} dt}{\int_{t_A}^{t_B} \sqrt{Z'_P(t)^2 + Z'_Q(t)^2} dt} \tag{3}$$

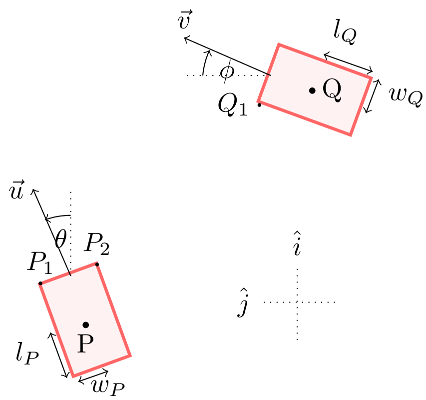
When performing this calculation, we restrict the curve to only include combinations occuring within the time indices defined by $t_A = \min(|Z_P|)$ and $t_B = \min(|Z_Q|)$. In effect, this restriction excludes (Z_P, Z_Q) combinations corresponding to large values of t where our motion projections are unlikely to still resemble the actual trajectories of each vehicle.

2.2 Object Collisions

Treating vehicles as points has substantial mathematical and computational advantages, but it represents too much of a simplification for some applications that require greater precision. In this section we outline the calculations of TCA and $SDCA$ when vehicles are represented by rectangles.

2.2.1 TCA for Objects

To begin, we use P and Q to denote the point at the center of each vehicle, we then define edges and corner points using each vehicles' length and width as defined in the diagram below:



To calculate the time to closest approach (TCA) of vehicles P and Q, we undertake the following steps:

1) Determine the corner points of the vehicle. For vehicle P, this is done by adding or subtracting the the $\vec{i}$ and $\vec{j}$ components of l_P and w_P:

$$P_k(t) = \begin{pmatrix} f_{\hat{i}}(t) + (-1)^{k-k \bmod 3} l_A \cos\theta + (-1)^{\frac{k-k \bmod 2}{2}+1} b_A \sin\theta \\ f_{\hat{j}}(t) + (-1)^{k-k \bmod 3} l_A \sin\theta + (-1)^{\frac{k-k \bmod 2}{2}} b_A \cos\theta \end{pmatrix}$$

where k $\in \{1,2,3,4\}$. Note, θ lies in the direction of $\hat{i}$, the formula for Q_k is similar, but needs to be adjusted because ϕ lies in the direction of $\hat{j}$.

2) Find equations for all the edges of the vehicle. For any two corner points on an edge, we determine the vector equation of the line on which the edge lies. For example, the line joining P_k and P_{k+1} can be written as:

$$A_k(t) = P_k(t) + (P_{k+1}(t) - P_k(t)) \cdot \lambda_k$$

where $\lambda_k \in \mathbb{R}$ indicates position on A_k relative to P_k, and $k+1 = 1$ when $k = 4$.

3) Find the minimum distance from each corner point of one vehicle to each edge of the other vehicle. To find the minimum distance from a point Q_{k*} to a line A_k, we first find the line passing through Q_{k*} and perpendicular to A_k, which we label N_k. Then, we find, m_k, the point of intersection of A_k and N_k by equating A_k and N_k. The minimum distance between line A_k, and corner point Q_{k*} can be written as:

$$d_{k,k*}(t) = \begin{cases} dist(Q_{k*}, m_k) & P_k \leq m_k \leq P_{k+1} \\ \min\{dist(Q_{k*}, P_k), dist(Q_{k*}, P_{k+1})\} & otherwise \end{cases}$$

where $dist(a,b)$ indicates distance between points a and b.

4) Minimize over all point-edge combinations of both vehicles to get TCA. Each of the 32 point-line combinations has its own distance as a function of time. TCA is then found by finding the time at which the overall minimum distance, across all combinations, occurs. This could be done by taking $\mathcal{D}(t) = \min\{d_{1,1}, d_{1,2}, ..., d_{4,4}\}$, differentiating $\mathcal{D}(t)$, and solving for a minimum. Or TCA could be approximated numerically using a grid search.

2.2.2 SDCA for Objects

When treating vehicles as points, $SDCA$ was based upon adjusted driver reactions, which were characterized by introducing latent variables Z_P and Z_Q which provided modified positions $\tilde{P}(t)$ and $\tilde{Q}(t)$. Because the rectangular representations of P and Q arise from center points, the calculation of $SDCA$ in this framework can be approached similarly. We outline the calculation below:

1) Modify the center points $P(t)$ and $Q(t)$ by introducing latent variables Z_P and Z_Q, resulting in adjusted center points $\tilde{P}(t)$ and $\tilde{Q}(t)$.

2) Using the procedure described in Section 2.2.1, determine the adjusted corner points $\tilde{P}_k(t)$ and $\tilde{Q}_k(t)$, and use them to find distance functions $\{d_{1,1}, d_{1,2}, ..., d_{4,4}\}$ for each point-edge combination.

3) For each value of t, find the (Z_P, Z_Q) pair closest to $(0, 0)$ that results in at least one of the distances $\{d_{1,1}, d_{1,2}, ..., d_{4,4}\}$ equaling zero. This represents the minimum combined reaction delay that would result in vehicle P colliding with vehicle Q.

4) Having obtained a curve of (Z_P, Z_Q) pairs as a function of time in Step 3, we proceed to calculate $SDCA$ using Equation (3).

3 Application

To demonstrate the utility of $SDCA$, we apply the measure to driving simulator data collected by the NADS. In doing so, we illustrate two possible uses of $SDCA$. First, we see how $SDCA$ can be used to forcast the occurrence of a collision based upon a vehicle's reaction profile. Second, we retrospectively analyze experimental data, replacing the binary outcome of collision/no collision using a numeric measure capable of capturing varying degrees of collision severity/avoidance.

3.1 Data Description

Haptic alerts are vibrations aimed drawing attention to a potentially dangerous situation. In this experiment, participants drove a simulated route while sitting on a custom haptic seat that could be controlled by the experimenters. The seat contained a 4x4 array of tactors used to convey alerts. Three different alert characteristics were manipulated, inter-pulse interval (IPI), intensity (INT), and directional pattern (DIP). A randomized block design was used to allocate 180 participants into various experimental conditions, using IPI = 0, INT = 50 Hz, abd DIP = "left only" as the default condtions. Within each block, one condition was varied, leaving the other two at their default values. Additionally, the design included a control group, which did not receive an alert.

During the experiment, particpants drove on urban street with a posted speed limit of 35 mph, crossing three traffic signal-controlled intersections. As they approached the third intersection, an incursion vehicle approached the intersection as cross-traffic from the left. The incursion vehicle was created such that a collision was eminent if the participant did not react. Roughly 4 seconds prior to collision the driver experienced a haptic alert, at which time the incursion vehicle was not yet visible. Roughly 2.9 seconds prior to collision the vehicle became visible.

Of these 180 participants who were randomized, 156 yielded time-series data suitable for analysis. These data contain vehicle speed and heading angle collected at a rate of 60 frames per second. Additionally, a dummy variable recorded whether or not the participant collided with the incursion vehicle.

3.2 SDCA Calculation

In our analysis we use the time-series measurements of vehicle speed and heading angle to derive the scalar acceleration and turn of wheel variables needed to calculate $SDCA$ by determining the rates of change in speed and heading angle at every instance of time. We define t_0, the reaction start-point, to be the first time after the onset of the alert where the scalar acceleration or turn of wheel changes significantly. We defined a "significant change" as a rate of change larger than 0.3^c for turn of wheel and larger than 3 m/s^2 for scalar acceleration. Based on the driving behaviors described by **?**, we split participants' reactions into five categories:

1. Scalar acceleration
2. Scalar deceleration
3. Turning
4. Scalar deceleration and turning

5. No reaction

Category 4, which is combination of two reactions, requires both acceleration and turn of wheel to significantly change within 0.167 seconds of each other. In this category the reaction start-point is determined by the first variable to significantly change.

Depending on which category the initial reaction falls under, we must determine the values of scalar acceleration or turn of wheel (or both) to use in the $SDCA$ calculation. We do this using the average from the start of the reaction to the variable's maximum or minimum value.

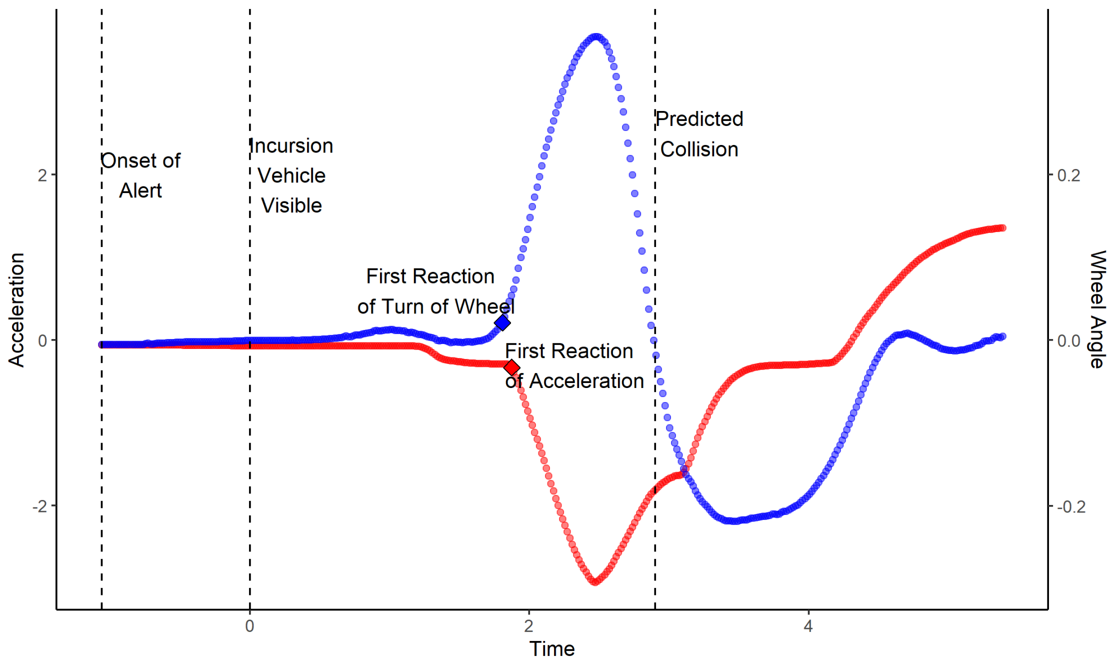


Figure 1: Time Series Data for Participant 65

Figure 1 displays the time-series data for participant 65, with the scalar acceleration and turn of wheel shown in red and blue respectively for every time frame. For this participant, the start of their reaction is the time at which turn of wheel first changes significantly and marked accordingly on the graph. Because scalar acceleration changes significantly within 0.167 seconds, participant 65's reaction is classified into Category 4. Upon receiving this classification, we calculate the necessary inputs of acceleration and turn of wheel by taking their average values from the start of reaction to their minimum and maximum values respectively.

We point out that this procedure creates a lag in when $SDCA$ can be calculated; however, $SDCA$ is still calculated prior to collision, and the amount of lag time could be reduced (potentially to zero) by determining reaction inputs differently.

Using the reaction start-points and the inputs described above are able to estimate the motion of each participant's vehicle. The incursion vehicle's motion is fixed, and easily accounted for. Using these charactizations, we calculate $SDCA$ for each participant via Equation (3). Because this calculation depends upon estimated quantities we denote the resultant value as $\hat{D}$, owing to the fact that it can vary depending upon how the reaction is estimated.

3.3 Forcasting Collision

In this section we consider the use of $SDCA$ as a classifer, comparing $\hat{D}$ values of each participant with whether they collided with the incursion vehicle.

Figure 2 displays a the distribution of $\hat{D}$ values for collsions and non-collisions, as well as the Receiver Operating Characteristic (ROC) curve illustrating the diagnostic ability of $\hat{D}$. A decision boundary of

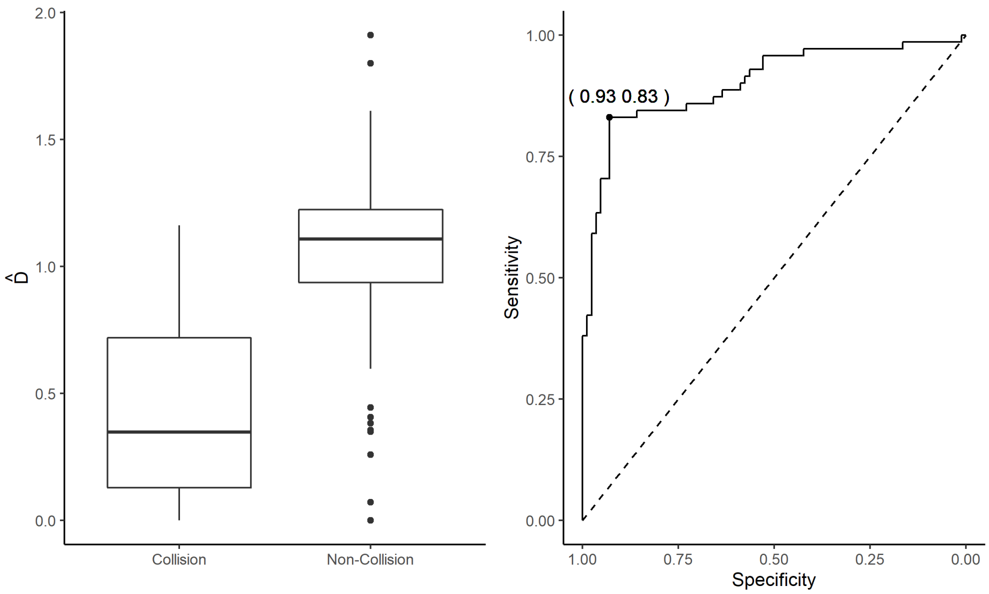


Figure 2: The left panel shows the distribution of $\hat{D}$ by collision, the right panel shows the ROC Curve of $\hat{D}$ as classifier of collision. The sensitivity and specificity corresponding to a decision threshold of $\hat{D} = 0.9$ are highlighted on the ROC Curve.

$\hat{D} = 0.9$ yields 93% specificity and 83% sensitivity, implying that the measure will correctly identify 93% of collisions as such, while only misclassifying 17% of non-collisions as collisions. Additionally the Area Under the Curve (AUC) of the ROC curve is 0.891, affirming $\hat{D}$ as a good classifier for a range of decision thresholds, and indicating expectional classification performance.

3.4 Evaluating Alert Characteristics

As previously indicated, the intent of this study was to determine in the impact of the haptic alert characteristics INT, IPI, and DIP on how the participant reacted to the incursion vehicle. Visual inspection shows that INT might have a relationship with whether a collision occured, a natural outcome of interest. INT also appears related to $\hat{D}$, an alternative measure for quantifying the participant's reaction.

Figure 3 illustrates differences in response for different levels of intensity, with IPI and DIP set at their default values. The expection is INT = 0, which represents the control group where no alert occurred. These graphs suggest that lower INT levels tended to result in fewer collisions, relative to both the control group and the higher INT conditions.

We might more formally assess the impact of these chracteristics using statistical modeling. Here we use logistic regression to model whether a collision occurred using the alert characteristics as explanatory variables. For comparison, we use linear regression to model the $\hat{D}$ using the alert characteristics.

Outcome	Family	Link	χ^2	df	p-value
$\hat{D}$	Gaussian	Identity	3.29	1	0.07
Collision	Binomial	Logit	2.42	1	0.12

Table 1: Likelihood ratio test results for INT for linear and logistic regression models

Table 1 shows the likelihood ratio tests for INT in the respective linear and logistic regression models with INT, IPI and DIP as predictors, and $\hat{D}$ or collision as the response variable. Both show some indication that INT is an important predictor, but the statistical confidence in the effect of INT differs. Although neither p-values reaches the convential significance threshold of $\alpha = 0.05$, using $\hat{D}$ as an outcome results

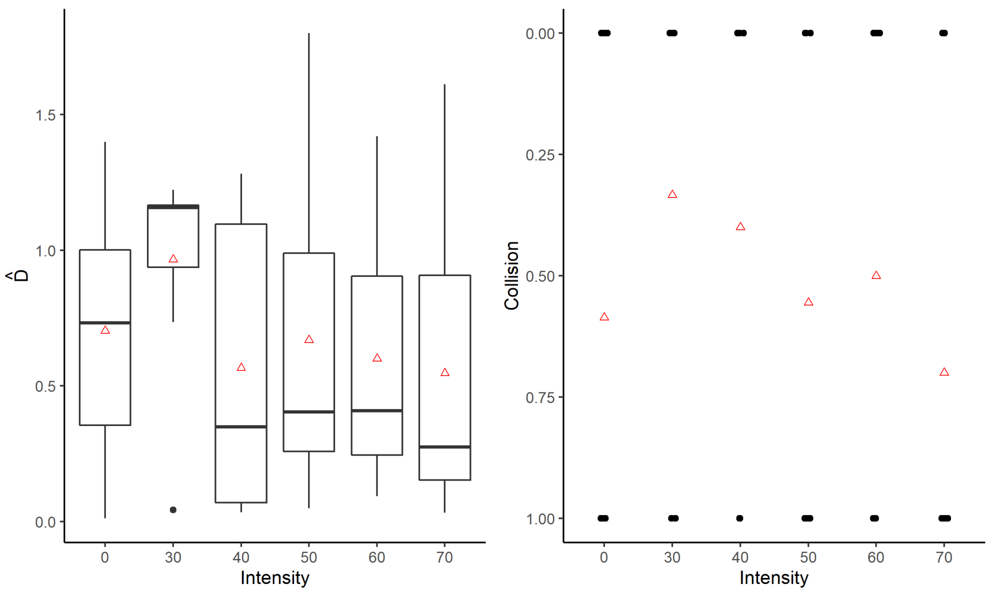


Figure 3: Trends in Alert Intensity using $\hat{D}$ and Collision

in a p-value that is lower by almost half. This illustrates how the added information contained $SDCA$ considerably improves statistical power in this application.

4 Discussion

TCA is the time to closest approach of two vehicles with set reaction profiles and a fixed reference point. In the event of a collision, TCA is an effective method of determining the time at which a collision occurs relative to the start of the reaction. Therefore, in case of collisions, smaller values of TCA indicate greater imminence of the collision. However, we have three salient criticisms of the metric:

1. Comparing across different instances poses a challenge, as the reference point needs to be fixed in order for TCA to be meaningful across instances. This is a substantial limitation, as reference points are not generally consistent when comparing across real-world crash situations.

2. TCA has limited applicability in the absence of a collision across multiple scenarios, as it is not reflective of imminence of danger. For example, a TCA of 4s does not distinguish between two sets of vehicles that are 5m and 15m apart at TCA. Therefore, TCA does not uniformly reflect proximity to collision.

3. In case of a collision, TCA does not include information about severity of a collision and hence, does not distinguish between major and minor collisions.

$SDCA$ addresses these limitations. $SDCA$, or standardized degree of collision avoidance, can be understood as the proximity to collision of two vehicles. Thus, $SDCA$ turns the binary variable of collision into a spectrum of positive values, with values further from 0 indicating greater levels of safety from collision.

$SDCA$ is capable of quantitatively measuring the degree of collision (or non-collision) between two vehicles with set reaction profiles, thereby addressing our second and third criticisms of TCA. Additionally, SDCA naturally equalizes different reference points because it uses delay in reaction times, rather than the time from the reference point itself. For example, if two vehicles are on track to collide and are currently two seconds away from their collision point their TCA is 2s while SDCA is 0. If we change the reference point to when the two vehicles are only 1s away from each other, their TCA becomes 1 while the SDCA remains 0. This simple example illustrates how SDCA is not reference-point dependent, which allows different scenario to have their own reference points while still providing for meaningful comparisons across scenarios.

In Section 2 we present two separate SDCA calculations, one treating vehicles as points, and another treating them as rectangles. We argue that both frameworks have utility.

Under the point framework, each point represents the center of one of the two vehicles. Here, $\hat{D} = 0$ indicates a collision of the centers of the vehicles, which, in many cases, represents the most severe collision possible. As $SDCA$ increases, collision severity can be thought to decrease, and after a certain point a collision no longer occurs and increases in $SDCA$ go on to represent decreasing proximity to collision. When applying the metric to data we can set data-dependent thresholds and interpret them accordingly. Figure 4 illustrates how $SDCA$ might be interpretted in this regard.

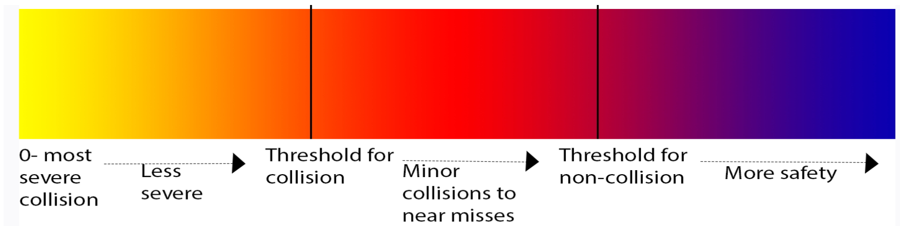


Figure 4: A visual depiction of how $SDCA$ can be used to represent varying degrees of severity when a collision occurs, and safety when a collision does not occur.

In Section 3.3 we consider the ability of $SDCA$ to discriminate between collisions and non-collisions. And, as Figure 2 indicates, the distributions of $\hat{D}$ for collision and non-collision differ greatly. In this application a threshold of $\hat{D} = 0.9$ yielded expectional classification performance.

Under the vehicles as rectangles framework, all non-zero values of $SDCA$ signify non-collision, with increasing values indicating less proximity to collision. If $SDCA = 0$, we can identify the point-edge pair that collides by observing the distance function that first returns 0. However, this information is not encoded in $SDCA$ itself but in its calculation, presenting a possible limitation.

Comparing these two frameworks, point-based calculation of $SDCA$ gives range of values that represent severity and proximity of collision, suggesting it to be a superior method for retrospective data analysis. Conversely, object-based calculation of $SDCA$ is better suited to identify collisions, suggesting it might do a better job forcasting collisions than the point-based calculation.

At this point we mention that there are many ways to use $SDCA$ for forecasting. In Section 3.2 we estimate a reaction profile starting a fixed point, and using information until the response peaks. In a real-time application where $SDCA$ is calculated dynamically this would require a lag before $SDCA$ could be calculated, presenting a minor barrier to using the measure to inform potential in-vehicles alerts. Nevertheless, even the lag used in the application in Section 3.2 is still short enough to provide information prior to the collision occuring.

Alternatively, the inputs needed to calculate $SDCA$ could also be estimated in real-time. This eliminates the need for a lag period, but introduces an additional challenge in making sure the instaneous inputs are reflective of the overall reaction and not overly specific to minute changes.

As mentioned above, using object based calculation of $SDCA$ might be better at forecasting as it considers all points of collision between two vehicles. By having a clear threshold that separates collisions and non-collisions, it can more accurately help vehicles avoid collisions entirely. Conversely, if a collision is imminent, point based calculation of SDCA would be useful as it consists of a spectrum of values that separate severe collisions from mild collisions. In a real-world situation, therefore, both calculations of SDCA should be used in tandem.

As demonstrated in Section 3.4, $SDCA$ improves statistical analysis relative to a dummy variable indicating collision. By using a continuous variable, we are able to distinguish between collisions of different levels of severity. For example, a head-on collision and a minor scrape are both classified as a collision, but their levels of severity differ greatly. Additionally, $SDCA$ also allows us to distinguish between non-collisions, as some collisions are barely avoided while others have quite a bit of leeway. Figure 3 and Table 1 both illustrate the superiority of $SDCA$ to a binary collision variable, which depict similar trends for INT across both response variables, but produce more statistically significant results with SDCA as the response. It is our opinion that using the point framework to calculate $SDCA$ is better for analysis because it incorporates theses distinction for both collisions and non-collisions.

5 Conclusion

In this manuscript we develop a time-based metric for measuring proximity to collisions. We present point-wise and object-wise calculations to determining $SDCA$, with both frameworks having value for different applications. Using data from a realistic driving simulator experiment, we demonstrate how $SDCA$ can be used to forecast the occurrence or non-occurrence of collision, as well as replace the binary collision outcome to strengthen statistical analysis.

References

Beltz, B. (2018). 100+ car accident statistics for 2019.

Brown, L. T. (2005). Adjusted minimum time-to-collision (TTC): A robust approach to evaluating crash scenarios 40–48.

Hayward, J. C. (1972). Near-miss determination through use of a scale of danger. *Highway Research Record* 24–34.

Hou, J., List, F. G. and Guo, X. (2014). New algorithms for computing the time-to-collision in freeway traffic simulation models. *Hindawi*, **2014**.

Schwarz, C. (2014). On computing time-to-collision for automation scenarios. *Transportation Research Part F: Traffic Psychology and Behaviour*, **27** 283–294.

Sobhani, A., Young, W. and Bahrololoom, M., Sareh amd Sarvo (2012). Calculating time-to-collision for analysing right turning behaviour at signalised intersections.