

# Collision Prediction and Optimal Trajectory Generation for Collision Avoidance Systems in Trackless Mobile Machines

Jesse Declercq<sup>1,\*</sup>, Theunis Botha<sup>1,\*\*</sup>, and Herman A. Hamersma<sup>1,\*\*\*</sup>

<sup>1</sup>University of Pretoria

Department of Mechanical & Aeronautical Engineering

Hatfield, Pretoria, 0002, South Africa

## Abstract.

The continued high number of fatalities associated with Trackless Mobile Machines (TMMs) in South Africa has led to the introduction of Collision Avoidance System (CAS) regulations in the Mine Health and Safety Act in 2015. These regulations have engendered the profusion of technologically-immature CASs from third-party vendors — all of which are centred on automatic stopping and braking systems. These braking systems often result in trivial or ineffective solutions, proving costly to mining operations. The amalgamation of braking and steering control in CASs may substantially increase the solution space, providing far safer and more efficient manoeuvres. The contribution made by steering control is evaluated through the development and implementation of a recursive non-linear collision prediction estimator and optimal trajectory generation model. Two independent optimal trajectory generation models are proposed to compete against one another in an attempt to synthesize the safest, most predictable, and efficient trajectory. These models are differentiated primarily by their path planning approaches: lattice optimization and Monte Carlo hyper sampling. The collision prediction estimator and trajectory generation models are evaluated in simulation using the Earth Moving Equipment Safety Round Table (EMESRT) mining interaction scenarios. Initial results indicate increased CAS solution spaces when adding steering control in collision-avoiding scenarios, providing safer and more effective solutions in high velocity vehicle interactions.

## 1 Introduction

The field of autonomous mobility has experienced radical expansion in recent years, with implementations progressing in commercial, industrial and passenger vehicles. This expansion has been incentivised with the potential reduction of mortality rates as human error is eliminated from public roads [1]. The Department of Mineral Resources' Occupational Health and Safety 2017 annual safety report recorded that 30-40% of mining industry fatalities were caused by mobile equipment, while 9% of the fatalities were caused by Trackless Mobile Machines [2], [3]. These concerning statistics lead to the amendment of the South African Mine

---

\*e-mail: [Jesse.Declercq@tuks.co.za](mailto:Jesse.Declercq@tuks.co.za)

\*\*e-mail: [Theunis.Botha@up.ac.za](mailto:Theunis.Botha@up.ac.za)

\*\*\*e-mail: [Herman.Hamersma@up.ac.za](mailto:Herman.Hamersma@up.ac.za)

Health and Safety Act (MHSA) legislation in 2015, compelling mine employers to introduce automatic emergency braking and proximity detection system at each mine where significant risk of collision exists. These systems must at least include: (a) the automatic detection of a pedestrian or TMM within its vicinity and (b) in the event where no action is taken to prevent a potential collision, further means to retard the TMM until a safe speed is reached [4] [5].

The Earth Moving Equipment Safety Round Table (EMESRT) developed a performance requirement in 2016 in order to augment the interpretation of operation controls in several TMM potential unwanted events [6] However, the MHSA's proposal of proximity detection systems is unclassified and neglected by EMESRT performance requirements, leaving substantial gaps in the collision prediction performance domain. In addition, in the South African mining industry, all current CASs developed by third party vendors make exclusive use of Collision Avoidance By Braking (CABB) systems due to the requirement by the MHSA legislation. These systems are often insufficiently tested or developed to address the significant and unique challenges associated with the practical implementation of such systems in mining environments [5]. Due to the single Degree-Of-Freedom (DOF) constraint, these CABB solutions operate in highly restricted solution spaces, often leading to trivial or energy-inefficient solutions. These solution spaces may be expanded by improving sensing ability, decision-making and road design, however, the solution space is still bound by the system's single DOF manoeuvrability, irrespective of system intelligence. Therefore, by introducing steering control as an additional DOF to the system, the solution space will be expanded significantly, potentially providing far safer and more efficient manoeuvres [7].

## 2 Collision Prediction

Collision avoidance systems make use of proximity detection systems as instructed by the MHSA in order to detect pedestrians, other TMMs and potential hazards on the road. Once a potential hazard has been detected, this information needs to be relayed to the CAS in a manner in which the appropriate control level can be articulated. This paper presents a probabilistic non-linear collision prediction model which integrates robust TTC calculations with non-linear state estimations and a lucid collision probability method.

### 2.1 Non-linear State Estimation

All collision prediction techniques rely on a form of vehicle modelling in order to predict future vehicle states. These states are in turn used to predict whether two vehicles or objects will, in future, occupy the same space in time. Some vehicle modelling techniques make use of rudimentary particle kinematic models whilst others make use of more sophisticated kinematic or dynamic vehicle models. The vehicle models are non-linear, capable of accurately tracking the vehicle's motion during cornering. Conversely, the particle model is linear, leading to large prediction errors during cornering manoeuvres. These errors become even more evident during large time steps — which are critical for collision prediction applications. Therefore, the motion modelling is limited to the kinematic or dynamic vehicle models.

These vehicle model, however, assume ideal initial state inputs and do not consider noisy or inaccurate state inputs. These state inputs are acquired from copious sensor and communication network measurements which are subject to vibrational and signal noise as well as accuracy limitations. Henceforth, there is a level of measurement uncertainty when considering the state input to the vehicle model for state estimation. State estimation filters are designed to synthesize better state estimates from noisy or inaccurate sensor measurements by minimizing the error covariance of the estimated state given the noise distributions. As

mentioned the motion of a vehicle is inherently non-linear during cornering and therefore, the state estimation filters must be capable of tracking non-linear manoeuvres. Filters which are capable of such tracking are the particle [8], unscented Kalman [9] and extended Kalman filters [10].

Particle filters are excellent when dealing with highly non-linear motion and when the state transition function is easy to compute. However, due to the expensive integration process within the vehicle motion models, the particle filter is infeasible due to the linear coupling of computation time and particle count [8]. Although the Extended Kalman Filter (EKF) and Unscented Kalman Filter's (UKF) accuracy is highly comparable, their computational efficiencies vary greatly. The UKF often takes more than double the computation time of the EKF [9]. Therefore, the choice between the two filters is one made solely on the computational resources available.

### 2.2 Time-To-Collision

The TTC metric identifies how much time the operator has to evade any potential oncoming collisions, as well as the CAS's required control level according to the EMESRT performance requirements. Although this paper focuses exclusively on level 9 control applications, the TTC metric built into the collision prediction model considers all possible future control level functionality. In the proposed collision prediction model, the TTC metric is calculated using two state estimation stages. The first stage makes use of a fixed step state series, where future states are estimated at a set of fixed time step intervals. The second stage makes use of a dynamic step state series using a calculated integration step size, where future states are estimated at a step size determined by a defined spatial resolution and the relative velocity between the host TMM and potential hazard. The time domain used for the second phase is determined using a standard bisection approach. This time domain is the domain in which the most accurate TTC will be determined. The TTC is then determined using the time in which the points of contact of the two bodies are closest to one another as seen in Figure 1.

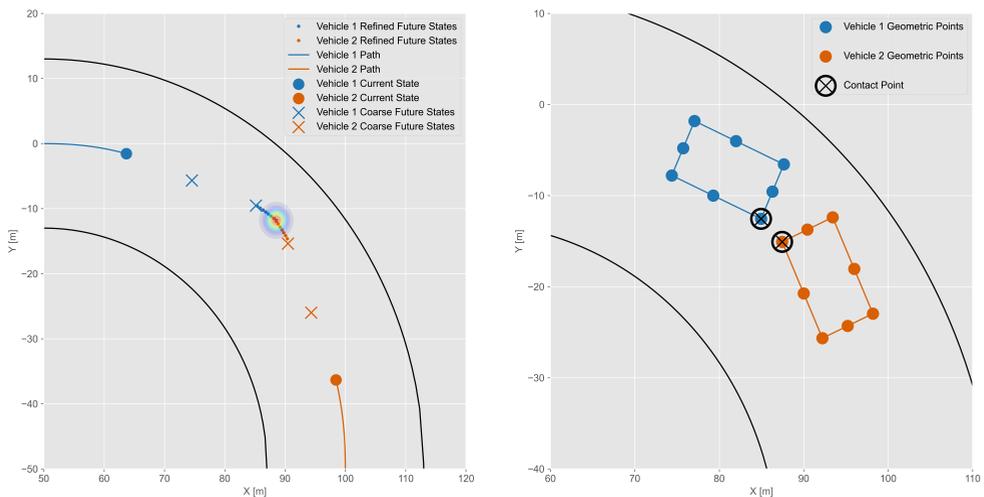


Figure 1: The collision prediction for a head-on collision around a bend. The multivariate Gaussian is shown using the filter error covariance matrix. The second plot indicates the eight contact points considered in collisions and the actual point of contact in the scenario.

Using the mean state estimate and error covariance matrix from the non-linear filter, the probability of collision between the two contact points can be determined.

### 2.3 Collision Probability

A probabilistic collision prediction model is capable of accounting for measurement uncertainties and noise present in sensors and communication systems. This is accomplished by exploiting filter state distributions. The mean estimated state vector and covariance matrix are used to model multivariate Gaussian's for each state estimate as per the filter. These distributions are used to calculate how likely the two vehicles or bodies are going to occupy the same spatial domain in time.

The probability of collision,  $\mathcal{P}$ , is calculated by integrating the joint distribution's probability over a safety region, parameterized by  $\ell$ , as shown in Equation 1. The probability of the joint distribution is determined by the product of the likelihood,  $\mathcal{L}$ , in Equation 2

$$\mathcal{P} = \int_{\bar{y} \in \ell} \int_{\bar{x} \in \ell} P(\Delta \mathbf{x}_k | \mathbf{x}_{k,1}, \mathbf{x}_{k,2}) d\bar{x} d\bar{y}, \quad (1)$$

where the posterior probability is calculated using Equation 2

$$P(\Delta \mathbf{x}_k | \mathbf{x}_{k,1}, \mathbf{x}_{k,2}) = \prod \mathcal{L}(\Delta \mathbf{x}_k | \mathbf{x}_{k,1}, \mathbf{x}_{k,2}) dA, \quad (2)$$

according to the assumed multivariate Gaussian state distribution defined by Equation 3.

$$\mathcal{L}(\mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right). \quad (3)$$

The multivariate Gaussian parameters are that of the non-linear filter's state estimation mean and error covariance matrix. A collision is flagged as true if  $\mathcal{P} > 1 - \rho$ , where  $\rho$  is the allowable probability of intersection. Once the collision is flagged as true, the final TTC is then relayed back to the CAS for possible control level evaluation.

## 3 Trajectory Generation

Trajectory generation is the development of a joint time-space path, planned according to a set of directives to accomplish an objective. In this paper, trajectory generation is split into two key phases: path planning and velocity profiling. Path planning is responsible for the spatial path development in which the vehicle must follow in order to safely achieve an end goal state. Two path planning approaches are proposed: the Dense Lattice sampling (DLS), Monte Carlo Lattice Hyper Sampling (MCLHS) models. Velocity profiling is the construction of a velocity map propagating longitudinally along the planned path which adheres to vehicle dynamic constraints. Consequently, the velocity profiling phase is contingent upon the path planning phase. The lateral, longitudinal, and combined loading effects are taken into account in order to ensure that neither the friction nor roll-over limits are exceeded [11].

### 3.1 Path Planning

Path planning aims to synthesize a smooth, collision-avoiding path along which vehicle stability is maintained. The intrinsic non-convexity and high dimensionality of the solution space proves challenging for most optimization algorithms to traverse. The DLS model

proved the most effective solution for the defined problem statement from literature; however, often results in either high curvature or collision inducing paths due to the independent curvature and collision avoidance objective functions failing to converge on global optima. Therefore, a novel MCLHS model was proposed to overcome these pitfalls, by propelling particles through the search space using a hyper sampling methodology and a chained Bayesian backpropagation.

### 3.1.1 Dense Lattice Sampling

The DLS model generates a lattice-bound path according to a set of obstacle avoidance and curvature objective functions [12]. The DLS model achieves an optimal path by independently minimizing both the obstacle avoidance and curvature objective functions by changing the nodal selection and seeded path offset respectively. Formally, the DLS model uses a non-parametric trajectory optimized for road geometry, obstacles and higher order motion directives according to a reference frame as shown in Figure 2.

Obstacle avoidance is prioritized over curvature minimization in the optimization execution order. Obstacle avoidance is achieved by making use of a cost function across all edges  $e_{n_k \rightarrow n_{k+1}}$  within the lattice as described by Equation 4

$$\operatorname{argmin}_{\{e_{n_k \rightarrow n_{k+1}}\}} \sum_{k=0}^{K-1} w_d d(e_{n_k \rightarrow n_{k+1}}) + (1 - w_d) o(e_{n_k \rightarrow n_{k+1}}) + \delta_{\text{obstacle}}(e_{n_k \rightarrow n_{k+1}}), \quad (4)$$

where  $d(e)$  is responsible for penalizing long edges,  $o(e)$  for penalizing lateral offsets, and the weighting scalar,  $w_d$ , determines the trade-off between these two terms. The term  $\delta_{\text{obstacle}}(e)$  — set to infinity — is responsible for preventing paths which result in collisions, ensuring edges which lead to obstacle nodes will never be sampled. Once the obstacle-avoiding path of lowest cost is chosen, known as the seeded path, it is then optimized for curvature, using Equation 5

$$\operatorname{argmin}_{\{o_i\}} \sum_{i=0}^{N-1} w_k \|\kappa_i\| + (1 - w_k) |o_i|. \quad (5)$$

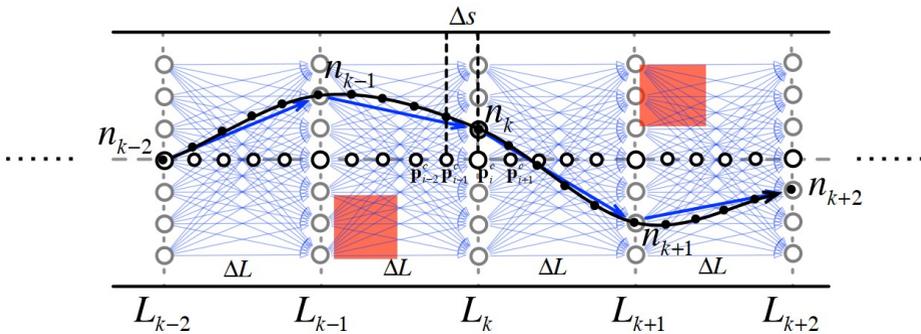


Figure 2: A sample dense lattice indicating the edge ( $e_{n_k \rightarrow n_{k+1}}$ ), layer ( $L_k$ ), node ( $n_k$ ) and sampling distance ( $\Delta s$ ) formation. This diagram also indicates the obstacle detection and avoidance capabilities within the generated seeded path [12].

where the weight  $w_k$  trades off between minimizing cumulative curvature  $\kappa$  and minimizing cumulative lateral offset  $o$ . This curvature optimization is performed by the Levenberg-Marquardt (LM) algorithm, however, the Sequential Least-Squares Quadratic Programming (SLSQP) algorithm proved to be far more effective in implementation. Lastly, a clamped cubic B-Spline is then used to smooth the discretized trajectory before being passed to the velocity profiler [13].

### 3.1.2 Monte Carlo Lattice Hyper Sampling

The novel MCLHS model makes use of a Monte Carlo path exploration technique coupled with Bayesian backpropagation. A series of particles are set to explore the reference path by means of a dense lattice. The connection between nodes from two different layers — "nerves" — are defined by probabilistic weights according to agility directives and lane biases. The goal of the MCHLS model is to overcome the considerable downfalls in most path planners such as DLS, adaptive potential fields, and curvature optimization-based models. Curvature optimization-based models often rely on global solutions attained through gradient method convergence — which is not always possible due to the non-convexity of the solution space. Instead, the MCLHS model propels a collection of particles propagating through the lattice based on a sampling methodology where particles decide on which nerve to choose based on its probabilistic weight,  $w_{m,n}$ , from the weighting matrix,  $\overline{\mathbf{W}}$  as defined in Equation 6.

$$\overline{\mathbf{W}} = \begin{bmatrix} \{w_{0:0} & w_{0:1}\}_{0,0} & \{w_{1:0} & w_{1:1} & w_{1:2}\}_{0,1} & \cdots & \{w_{n:n-1} & w_{n:n}\}_{0,n} \\ \{w_{0:0} & w_{1:1}\}_{1,0} & \{w_{1:0} & w_{1:1} & w_{1:2}\}_{1,1} & \cdots & \{w_{n:n-1} & w_{n:n}\}_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \{w_{0:0} & w_{0:1}\}_{l,0} & \{w_{1:0} & w_{1:1} & w_{1:2}\}_{l,1} & \cdots & \{w_{n:n-1} & w_{n:n}\}_{l,n} \end{bmatrix} \quad (6)$$

All possible nerves in which the particle can sample — according to the agility directive — sum to one and are known as nerve groups. Nodes which are classified as obstacle nodes have all corresponding nerves connecting them set to zero so that they cannot be sampled. The weight of each nerve chosen in the particle's path is updated according to how likely this path's curvature is the lowest amongst all possible paths. Using a Bayesian approach the posterior probability is defined by Equation 7 [14]

$$p(w_i|\kappa_o) = \frac{p(\kappa_o|w_i)p(w_i)}{p(\kappa_o)} \quad (7)$$

where

$$k_o = \{\kappa_p \in \mathbf{K} : \kappa_p < \kappa_i \forall \kappa_i \in \mathbf{K}\}. \quad (8)$$

The prior probability,  $p(\kappa_p)$ , is challenging to obtain due to its lattice architecture dependence. Nonetheless, this prior is only necessary for normalization and therefore the normalization constant,  $\alpha$ , is substituted across nerve groups instead. Henceforth, the weights are constantly renormalized at each nerve group after backpropagation. By evaluating the structure of the lattice, the likelihood,  $p(\kappa_o|w_i)$ , can be found by simply applying the product of the chosen weights after weight  $i$  until the end of the path which resulted in curvature  $\kappa_p$ . The Bayesian backpropagation is then represented mathematically in Equation 9

$$p(w_i|\kappa_o) = \alpha \cdot p(w_i) \cdot \prod_{j=i+1}^l p(w_j). \quad (9)$$

This is done for every particle after it has fully propagated through the lattice and has obtained a resultant curvature. The final nerve, however, is not accounted for in this Bayesian backpropagation. Instead,  $p(w_l)$  is determined using Equation 10

$$p(w_l) = \begin{cases} \frac{1}{N_p}, & \text{if } \kappa_p < \kappa_o \\ 1, & \text{if } \kappa_p \geq \kappa_o \end{cases} \quad (10)$$

where  $N_p$  is the total number of particles which have already explored unique paths. After a particle has completed its path, the path's curvature is then calculated using Mengers Equation. Finally, the path with the lowest curvature is then selected and a clamped cubic B-Spline is then used to smooth the discretized trajectory before being passed to the velocity profiler [13]. Comparing the DLS and the MCLHS models using five randomized obstacles is shown in Figure 3. The DLS was modelled using an optimized  $w_d$  and  $w_k$  parameters. The DLS model in Figure 3 (a) fails to converge upon the optimal solution using both the SLSQP and LM algorithms. The MCLHS converges upon the optimal solution within 100 particles. The particle paths are indicated in Figure 3 (b) highlighting the Bayesian backpropagation of the model.

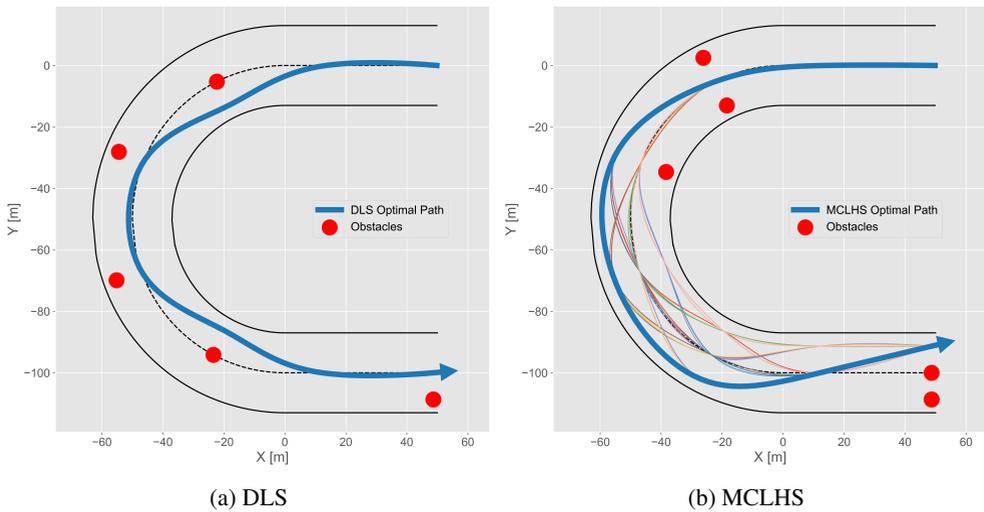


Figure 3: The comparison between the DLS and the MCLHS using five randomized obstacles with the DLS model indicated in (a) and the MCLHS in (b).

### 3.2 Velocity Profiling

Once the vehicle's path has been planned, the coupled velocity profile is developed in order generate the final trajectory. The velocity profile is developed in the spatial domain using only an initial vehicle state, planned path and its associated curvature. The maximum allowable speed is dictated by the minimum between the lateral acceleration and legal speed limit. The friction circle represents a tyre's maximum grip in both the longitudinal and lateral directions. In order to account for the mining haul trucks susceptibility to roll-over, the friction circle is often modelled as an ellipse rather than a circle, narrower in the lateral direction [11].

A lookahead distance is used to avoid the maximum velocity vector's discontinuous nature. Simply, the lookahead distance  $x_{ld}$ , is the maximum expected braking distance  $x_{b,local}$  at a given current speed  $v_c$ . The lookahead distance introduces a search space from the current

vertex to the lookahead distance vertex. At each vertex the local maximum allowable speed  $v_{local}$  is defined. The distance required to brake from the current velocity  $v_c$  to the propagated local maximum velocity  $v_{local}$  is calculated if  $v_c > v_{local}$  and termed  $x_{b,local}$ . If  $\Delta x_{local}$  is larger than the vertices longitudinal difference  $\Delta x_{c \rightarrow p}$ , then braking is flagged as required.

This search space is utilized to explore longitudinal acceleration solutions at the current longitudinal position. The longitudinal acceleration  $\ddot{x}$  for the velocity profile can be calculated using Equation 11, where the maximum allowable deceleration and acceleration is defined by  $b_{max}$  and  $a_{max}$  respectively.

$$\ddot{x} = \begin{cases} \mathcal{I}b_{max} & \text{if } x_{b,local} + x_\epsilon > \Delta x_{c \rightarrow p} \\ \mathcal{I}a_{max} & \text{elif } v_{max,c} > v_c + v_\epsilon \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The allowable longitudinal acceleration as a percentage of the maximum allowable longitudinal acceleration is termed the intensity, denoted by  $\mathcal{I}$ . The longitudinal acceleration intensity can therefore be calculated by considering the friction ellipse as shown in Equation 12

$$\mathcal{I} = \sqrt{1 - \left( \frac{v_r^2 \kappa_l}{\ddot{y}_{lim}} \right)^2}, \quad (12)$$

with  $x_\epsilon$  and  $v_\epsilon$  is a safety distance and speed respectively. If no longitudinal acceleration or braking is required, the previous velocity will be continued as the expected velocity. Therefore, the velocity profiler requires the initial velocity of the vehicle as the trajectory is generated. The expected speed vector  $v_e$  is then generated for each longitudinal vertex along the planned path, allowing for a spatial domain for the velocity profile. The velocity profile ensures safe and legal trajectories for the ego vehicle and maintains vehicle stability. In order to execute this trajectory, however, a controller is required.

### 3.3 Dynamic Controller

The dynamic controller is responsible for executing the velocity profile as well as following the desired planned path. This dynamic controller is a simplified model of the steering controller proposed by T.R. Botha in 2011 [15]. This steering controller makes a combined use of yaw  $e_{yaw}(t)$  and lateral offset  $e_{lat}$  errors. The yaw component ensures the vehicle's path is parallel to that of the planned path and the lateral offset component ensures the vehicle does not drift from the path due to cross-track errors [15]. The yaw component is given a proportional, integral and derivative term, whereas, the lateral offset component is only given the proportional term. The required steering rate for the dynamic controller is therefore calculated in Equation 13

$$\dot{\delta}(t) = f_{p,yaw}(\dot{x})e_{yaw}(t) + f_i(\dot{x}) \int e_{yaw}(t)dt + f_d(\dot{x}) \frac{de_{yaw}(t)}{dt} + f_{p,lat}(\dot{x})e_{lat}(t). \quad (13)$$

The proportional ( $f_p$ ), integral ( $f_i$ ) and derivative ( $f_d$ ) gains are all functions of velocity. This modelling choice was made as steering becomes more sensitive as vehicle speed increases [15]. The predicted lateral acceleration due to the calculated steering angle is compared to the allowable lateral acceleration. Should the predicted lateral acceleration exceed the allowable lateral acceleration, the steering angle  $\delta_t$  is adjusted using Equation 14

$$\delta_t < \tan^{-1} \left( \sqrt[2]{\left( \frac{v_r^2 \left(1 - \frac{l_r}{l_r + l_f}\right)}{\ddot{y}_{roll,t} l_f} \right)^2 - \frac{l_r}{l_r + l_f}} \right). \quad (14)$$

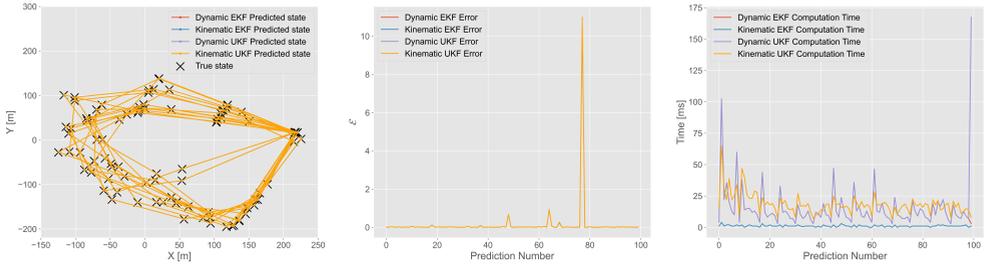


Figure 4: The comparison between the four different non-linear state estimators across positional estimates, relative errors, and computational speeds.

The longitudinal distance between the centre of gravity and the front and rear axles, denoted as  $l_f$  and  $l_r$  respectively, are simple geometric measurements, playing a crucial role in the maximum allowable steering angle. This Equation is derived using both centripetal acceleration and vehicle kinematic relations. It is capable of determining the maximum allowable steering angle to prevent roll-over. Equation 14 is in fact deterministic in implementation; the maximum allowable steering angle is always desired if triggered in extreme manoeuvres.

## 4 Preliminary Results

A realistic, physics-based simulation environment was developed in order to validate the collision prediction and trajectory generation model’s performance. The geometric parameters of the CAT 789D mining haul truck was used alongside the validated tyre and dynamic vehicle parameters of the Land Rover Defender 2013 110 from the Vehicle Dynamics Group at the University Pretoria. A series of simulations were performed in order to evaluate model capabilities. These simulations provided initial potential value in both the collision prediction and trajectory generation model for steering and braking based CASs.

### 4.1 Non-linear State Estimation

The UKF and EKF were coupled with the kinematic and dynamic vehicle models to create four different non-linear state estimators. They were evaluated across two metrics: accuracy and computational speed. The accuracy of the filter is measured using the relative error between the estimate and ground truth positional and yaw state components. The computational speed is evaluated as the time taken to complete a single prediction given a static time step size.

Using randomized paths and velocities, a total of 100 prediction steps were used across various time step sizes. The accuracy measurements between the UKF and EKF were superfluous as indicated in Figure 4. At low noise/uncertainty, both filters performed identically. At high noise/uncertainty, the UKF boasted a 0.4 - 0.9% higher accuracy than the EKF, with the dynamic vehicle model always outperforming the kinematic model as expected. However, as seen in Table 1 and Figure 4, the EKF boasted a nearly 6x faster computation time compared to the UKF when using the dynamic model and a near 10x faster computation time when using the kinematic model.

Table 1: Non-linear state estimator computation time comparisons, where the time step  $\Delta t$  is measured in seconds and all statistical measures are evaluated in milliseconds.

$\Delta t$	Dynamic Models				Kinematic Models			
	EKF		UKF		EKF		UKF	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
1	6.96	3.22	169.77	46.35	0.48	0.52	7.88	2.21
5	10.65	4.42	25.79	64.64	0.80	0.50	10.08	2.70
10	10.05	5.30	28.04	66.54	0.71	0.49	11.36	2.51
20	14.50	16.50	34.94	80.56	1.00	0.85	17.80	5.17

### 4.2 Trajectory Generation

The trajectory generation model was evaluated using an L1 head-on collision interaction scenario according to EMESRT requirements. This interaction scenario is used to evaluate the ability of the path planner to develop a collision avoiding smooth path, the velocity profile to synthesize a coupled safe trajectory and the dynamic controller to execute the final trajectory. Furthermore, a comparison between CABB and the proposed CAS is compared in this interaction scenario to advocate the potential value added by the addition of steering control. The performance of the MCLHS model coupled with the velocity profiler and dynamic controller in an L1 interaction scenario is indicated in Figure 5.

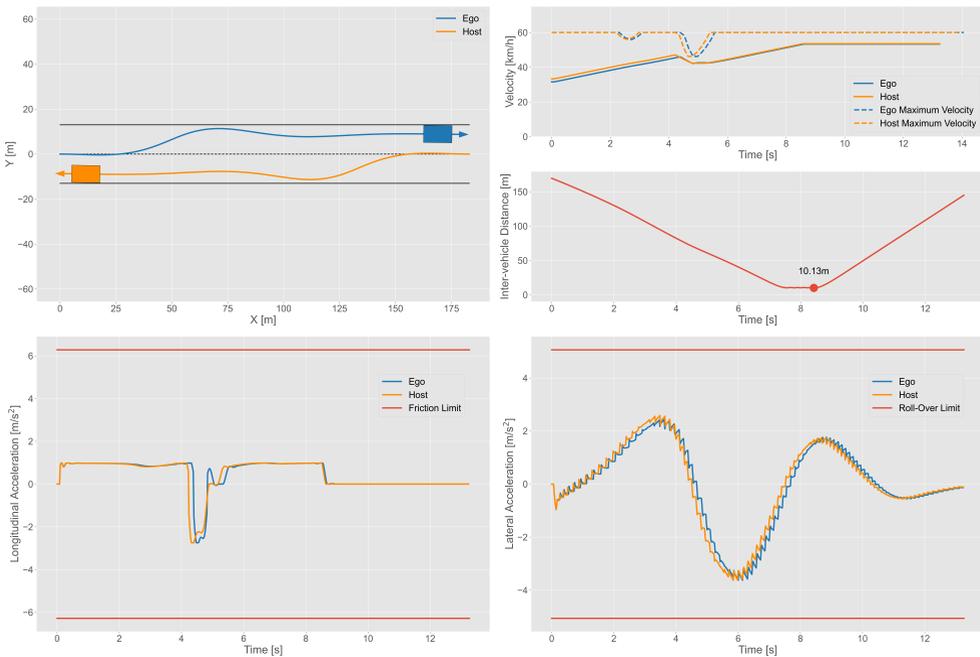


Figure 5: The simulation results in a L1 EMESRT interaction scenario using the MCLHS model. These plots indicate the planned path, velocity profile intervehicle distance and combined loading effects on each vehicle.

The model yielded exceptional results as the collision was avoided and vehicle stability was maintained whilst complying to left-lane standard in South African traffic law. The results are promising; however, additional EMESRT interaction scenarios are required to be evaluated. These interaction scenarios should be simulated many times in order to develop a statistical analysis of the model’s performance.

### 4.3 Steering Contribution

The potential value added by steering control was evaluated over two key performance metrics: time to avoidance and distance to avoidance over a range of velocities. The results in Figure 6 highlight how the avoidance distance increases quadratically during braking manoeuvres and increases linearly during steering manoeuvres. When analysing the avoidance time, braking manoeuvres increased the time required linearly, whereas steering manoeuvres decreased the time required exponentially. The transition speeds between the avoidance time and distance, differ significantly; the transition speed when analysing the avoidance distance is 33km/h and 24km/h when comparing the avoidance time. The shaded regions of the graph indicate which manoeuvre proves to be more effective with respect to its relative metric. Clearly, at higher speeds steering manoeuvres are superior to braking due to the decreased required response time and distance. At lower speeds, braking may prove to be more effective if the collision may not be averted with steering or the response time required is exceptionally low.

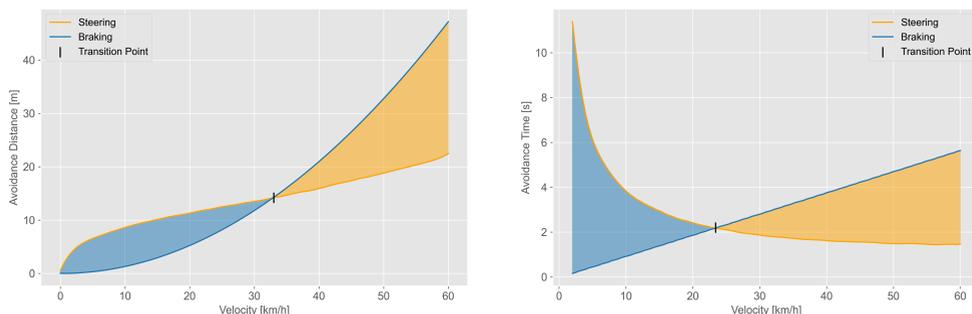


Figure 6: Comparison between the braking distance and steering distances and times to avoid collisions in a head-on collision scenario. The distances are the minimum distance the vehicle travels longitudinally to avoid the collision.

## 5 Discussion

This paper introduces a probabilistic non-linear collision prediction and optimal trajectory generation model for collision avoidance systems which take advantage of the addition of steering control. The collision prediction model incorporates non-linear state estimation with probabilistic TTC’s and geometric point-contacts in order to flag potential collisions. The trajectory generation model is split into path planning and velocity profiling. This model creates a cohesive spatial path and velocity profile for the vehicle to follow, avoiding collisions and maintaining vehicle stability.

Initial results illustrate the collision prediction model’s competence during non-linear manoeuvres and high measurement uncertainty. The trajectory generation model’s velocity profiler and dynamic controller proved competent in simulation, maintaining vehicle stability

and safety in all tests. The MCLHS model overcomes many of the shortcomings synonymous with common path planners — such as the DLS model — and demonstrates predictable and intelligent decisions in collision avoidance scenarios. These models have all shown exceptional value in a small selection of applications; however, a series of simulations and statistical analyses are required before any performance conclusions can be made.

## References

- [1] M. Green, J. Senders, *Human error in road accidents* (Visual Expert, 2004)
- [2] P. Moore, *Mine collision management system testing accelerates with South Africa's 2020 legislation deadline looming* (IM Mining, 2019)
- [3] Occupational Health and Safety, *Trackless Mobile Machinery* **1**, 1 (2017)
- [4] Mine Health and Safety Act, Amendment of Chapter 8 of the regulations relating to machinery and equipment (South Africa Proclamation No. R. 125). Government Gazette 38493:22 **29**, 72 (2015)
- [5] C.D. H.A. Hamersma, P.S. Els, *Mining Goes Digital* (CRC Press, 2019)
- [6] EMESRT, *Vehicle Interaction Systems*, in *PR – 5A* (EMESRT, 2016), pp. 1–11
- [7] J. Jansson, Ph.D. thesis, Linköping University (2005)
- [8] G. Hendeby, R. Karlsson, F. Gustafsson, *EURASIP Journal on Advances in Signal processing* **2010**, 1 (2010)
- [9] M. Wielitzka, A. Busch, M. Dagen, T. Ortmaier, in *Kalman Filters-Theory for Advanced Applications* (InTech, 2018), pp. 56–75
- [10] J. Jansson, Ph.D. thesis, Linköping University Electronic Press (2005)
- [11] T.D. Gillespie, *Fundamentals of vehicle dynamics*, Vol. 400 (Society of automotive engineers Warrendale, PA, 1992)
- [12] T. Gu, J. Snider, J.M. Dolan, J.w. Lee, *Focused trajectory planning for autonomous on-road driving*, in *2013 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2013), pp. 547–552
- [13] I. Noreen, MDPI *Symmetry* (2020)
- [14] R. Swinburne, *Revue Philosophique de la France Et de l* **194** (2004)
- [15] T.R. Botha, Master's thesis, University of Pretoria (2011)