Collision Avoidance – Convex Optimization

Leo Jaeeun Kim
University of Michigan
General Motors

ABSTRACT

Collision avoidance is a fundamental capability for autonomous vehicles and advanced driver-assistance systems (ADAS), forming the basis for safe and reliable operation. A typical system integrates perception, prediction, and planning modules to detect threats, estimate risk, and execute avoidance maneuvers. Among risk assessment strategies, predictive time-to-collision (TTC) estimation remains a foundational metric for proactive decision-making.

This research proposes an optimization-based iterative solver for predictive TTC estimation, incorporating vehicle dynamics and projected motion evolution. The solver combines reduced-gradient descent with Newton refinement to minimize future separation distances between the host vehicle and dynamic obstacles. To ensure robustness against non-convex motion behaviors, fallback sampling strategies are introduced. Simulation results demonstrate that the proposed method improves TTC prediction accuracy across diverse driving scenarios, supporting safer and more reliable trajectory planning in compliance with operational safety standards such as ISO 26262, SOTIF, and NCAP protocols.

INTRODUCTION

Collision avoidance systems are essential components within autonomous driving and ADAS architectures, tasked with identifying potential hazards, assessing collision risk, and executing avoidance maneuvers when necessary. A typical collision avoidance pipeline consists of several core stages: object detection and tracking, motion prediction of surrounding actors, collision risk estimation, and planning of evasive or mitigation actions. Among these stages, accurate collision risk prediction is critical for enabling timely and effective responses.

Predictive time-to-collision (TTC) estimation provides a fundamental building block for collision risk assessment, offering a quantitative measure of how soon a potential collision may occur if current trajectories are maintained. Traditional TTC estimation methods often assume constant velocities or linear motion models, which can be insufficient under real-world conditions where vehicles and obstacles exhibit acceleration, turning, or other nonlinear behaviors. Consequently, more robust and dynamic TTC estimation methods are necessary to handle diverse operational scenarios.

This research focuses on developing an iterative optimization-based framework for predictive TTC estimation. The proposed solver minimizes future separation distance over time by combining reduced-gradient descent updates with Newton-based refinement steps when local convexity conditions permit. To address non-convex dynamic behaviors and ensure stability, fallback sampling strategies are employed. By explicitly incorporating vehicle dynamics and motion uncertainties, the solver enhances prediction accuracy and supports safer trajectory planning in compliance with stringent operational safety requirements.

METHOD

1) Vehicle Dynamics Model

Accurate motion prediction of both host and target vehicles is essential for reliable Time-to-Collision (TTC) estimation. In this work, vehicle trajectories are modeled using a second-order kinematic formulation while explicitly accounting for heading angles, turning dynamics, and global frame transformations. This section presents the mathematical model underlying the predicted motion states employed in TTC calculations.

The global position of each vehicle is projected based on its initial state and assumed constant acceleration along the heading direction. The predicted $x(t)$ and $y(t)$coordinates are given by:

$$x(t) = x_0 + v_0 \cos(\theta_0)\, t + \frac{1}{2} a_0 \cos(\theta_0)\, t^2$$

$$y(t) = y_0 + v_0 \sin(\theta_0)\, t + \frac{1}{2} a_0 \sin(\theta_0)\, t^2$$

where:

x_0, y_0 denote the initial global position,

v_0 represents the initial longitudinal speed,

a_0 is the longitudinal acceleration,

θ_0 denotes the initial heading angle relative to the global X-axis.

By incorporating heading direction into the motion model, the predicted positions remain accurate even for vehicles not aligned with the global coordinate frame.

The evolution of velocity components in the global X and Y directions is described as:

$$v_x(t) = v_0 \cos(\theta_0) + a_0 \cos(\theta_0) t$$

$$v_y(t) = v_0 \sin(\theta_0) + a_0 \sin(\theta_0) t$$

These velocity components are utilized in computing relative motion, projected rates of closure, and intermediate distance-based quantities necessary for TTC estimation.

In cases involving steering inputs or curved trajectories, a nonzero yaw rate $\dot{\psi}$ must be considered. Assuming a constant yaw rate during the prediction horizon, the heading angle evolves linearly with time:

$$\theta(t) = \theta_0 + \dot{\psi} t$$

where ψ is the yaw rate measured in radians per second. This formulation captures the vehicle's changing orientation due to steering maneuvers, enhancing prediction accuracy especially during turning, merging, or lane-change scenarios.

For vehicles undergoing sustained turning motion with constant yaw rate and speed, their trajectory follows a circular arc of radius r, defined as:

$$x(t) = x_0 + r * \sin(\theta(t)) - r * sin(\theta_0)$$

$$y(t) = y_0 - r * \cos\big(\theta(t)\big) + r * cos(\theta_0)$$

This circular arc model provides a more physically accurate description of vehicle paths under constant turning conditions compared to simple linear extrapolation.

The following assumptions are adopted to balance model fidelity and computational tractability:

Short Time Horizon: Predictions are constrained to a limited future window (typically 1–2 seconds) where constant acceleration and yaw rate assumptions are reasonably valid.

Neglect of Lateral Slip: Lateral slip angles and side forces are assumed negligible, appropriate for moderate speed and non-aggressive maneuvers.

Planar Motion: Vertical dynamics, road slopes, and banking effects are neglected, assuming flat terrain.

These assumptions are consistent with typical requirements for real-time automotive collision prediction systems.

2) Relative Motion and Separation Distance

To compute the Time-to-Collision (TTC), the first step is to evaluate the relative motion between the host vehicle and a potential obstacle or object. Let $x_{host}(t)$, $y_{host}(t)$ and $x_{obj}(t)$, $y_{obj}(t)$and denote the global positions of the host and object, respectively, at time *t*.

The relative displacement between the two entities is defined as:

$$\Delta\mathrm{x}(t) = (x_{obj}(t) - x_{host}(t))$$

$$\Delta\mathrm{y}(t) = (y_{obj}(t) - y_{host}(t))$$

The Euclidean separation distance is then computed as:

$$d(t) = \sqrt{\big(\Delta\mathrm{x(t)}\big)^2 + \big(\Delta\mathrm{y(t)}\big)^2}$$

This distance d(t) quantifies the instantaneous gap between the predicted positions of the host and the object at a given future time. Its evolution over time is directly influenced by the vehicles' initial states and dynamic parameters (e.g., velocity, acceleration, heading, yaw rate).

The primary objective is to estimate the **Time-to-Collision** as the that minimizes this separation distance:

$$t^* = \underset{t \geq 0}{\operatorname{argmin}}\, d(t)$$

If the minimum distance $d(t^*)$ falls below a predefined safety threshold d_{safe} , the system identifies a potential collision risk and triggers appropriate mitigation actions such as braking or evasive steering.

$$Trigger\ if\ d(t^*) < d_{safe}$$

This formulation transforms the TTC problem into a continuous-time distance minimization problem subject to motion dynamics. Solving it robustly and efficiently is critical for safe and responsive collision avoidance.

3) Convex Optimization Formulation

3.1) Problem Statement

The prediction of Time-to-Collision (TTC) can be mathematically posed as an optimization problem where the goal is to determine the time instant $t \geq 0$ at which the separation distance between the host and the object reaches its minimum. This formulation leverages principles from convex optimization and numerical gradient-based methods to efficiently find the collision risk window.

The TTC prediction is mathematically expressed as the minimization problem:

$$\min_{t \geq 0} d(t)$$

where $d(t)$ denotes the Euclidean separation distance between the predicted positions of the host and the object at time t.

Given the quadratic nature of the vehicle motion equations, the squared separation distance $d^2(t)$ results in a quartic polynomial in t. Consequently, $d(t)$ is generally a nonconvex function over the entire time domain. However, under practical assumptions such as bounded accelerations, forward motion, and limited prediction horizons, a locally convex region typically exists around the expected collision time.

3.2) Initial Guess Estimation

To facilitate convergence, an initial estimate t_0 is generated based on relative velocity projections. Assuming initial constant velocity motion, the approximation is: Δ

$$t_0 = \frac{\Delta x_0 * \Delta v_{x_0} + \Delta y_0 * \Delta v_{y_0}}{\left(\Delta v_{x_0}\right)^2 + \left(\Delta v_{y_0}\right)^2}$$

where:

$\Delta x_0, \Delta y_0$ represent the initial relative position components,

$\Delta v_{x_0}, \Delta v_{y_0}$ represent the initial relative velocity components along the global axes.

This initial guess serves to accelerate convergence by starting the search near the region of interest.

3.3) Gradient-Based Iterative Refinement

A reduced gradient descent method is utilized to refine the TTC estimate. The iterative update rule is defined as:

$$t_{k+1} = t_k - \alpha_k \nabla d(t_k)$$

where:

α_k denotes the step size at iteration k

$\nabla d(t_k)$ denotes the derivative of the separation distance with respect to time evaluated at t_k.

The gradient $\nabla d(t)$ provides directional information indicating how the separation distance changes with time. Iterative updates along the negative gradient direction aim to reduce the separation distance toward a local minimum.

The step size α_k is either chosen as a small fixed scalar or adapted dynamically using techniques such as backtracking line search to ensure sufficient descent at each iteration.

After each update, t_{k+1} is projected onto the feasible domain to enforce the physical constraint:

$$t_{k+1} \geq 0$$

thus avoiding nonphysical negative time solutions.

3.4) Stopping Criteria

The optimization process is terminated when any of the following conditions is satisfied:

- The norm of the gradient falls below a predefined threshold ϵ:

  $$|\nabla d(t_k)| < \epsilon$$

  indicating proximity to a stationary point.

- The maximum number of allowable iterations $N_{\max}$ is reached.

- The change in successive time estimates $|t_{k+1} - t_k|$ falls below a minimal threshold, signifying convergence.

These stopping criteria ensure numerical stability while maintaining computational efficiency necessary for real-time implementation.

3.5) Convexity Considerations and Feasibility Region

Although the global structure of $d(t)$ is nonconvex, the local behavior around the initial estimate $t_0$enerally exhibits quasi-convexity under typical driving conditions. Key factors contributing to local convexity include:

- Bounded and smooth acceleration profiles
- Monotonic forward motion of both vehicles
- Prediction horizons limited to a few seconds

Within such local regions, the gradient descent method is effective in converging toward the true minimum separation point corresponding to a potential collision event.

3.6) Collision Mitigation Criterion

Upon convergence, the minimizing time t^* is obtained, and the corresponding minimum separation distance $d(t^*)$ is evaluated. Collision mitigation actions are triggered if:

$$d(t^*) < d_{safe}$$

where d_{safe} denotes a predefined threshold representing the minimum allowable safe distance between vehicles.

Triggering decisions based on the optimized TTC provide a mathematically grounded, dynamic, and responsive collision avoidance capability for active safety systems.

4) Prediction and Real-Time Considerations

The ability to compute Time-to-Collision (TTC) predictions within stringent real-time constraints is critical for integration into active safety and autonomous vehicle systems. Computational efficiency must be carefully balanced with prediction accuracy to ensure robust operation across a wide range of dynamic driving scenarios.

4.1) Iteration Limitation for Real-Time Feasibility

To guarantee that the optimization procedure remains computationally tractable, the number of gradient descent iterations per prediction cycle is strictly limited. In typical implementations, a maximum of 5 to 8 iterations is permitted within a single update window. This upper bound ensures that the TTC estimation remains within a bounded execution time budget compatible with real-time system requirements.

Empirical studies indicate that in most driving conditions, the local convexity near the initial estimate enables rapid convergence within a few iterations. Hence, excessive iteration counts provide diminishing returns and unnecessarily burden computational resources.

4.2) Dynamic Step Size Adjustment

The step size α_k used in the gradient descent updates is dynamically adjusted based on the current vehicle states and relative motion characteristics. Factors influencing step size adjustment include:

- Magnitude of relative velocity,
- Magnitude of relative acceleration,
- Gradient magnitude at the current iteration.

Larger relative velocities generally warrant smaller step sizes to prevent overshooting the minimum separation time, while smaller velocities allow for more aggressive updates. Dynamic adjustment strategies enhance convergence stability and adaptivity to varying traffic conditions.

4.3) Fallback Mechanism for Convergence Failures

In rare cases where the optimization fails to converge within the permitted iteration budget or encounters numerical instabilities, a fallback strategy is employed. Under fallback, a simplified constant-velocity TTC approximation is calculated by assuming zero acceleration for both the host and object:

$$TTC_{cv} = \frac{\Delta x_0 * \Delta v_{x_0} + \Delta y_0 * \Delta v_{y_0}}{-\left(\left(\Delta v_{x_0}\right)^2 + \left(\Delta v_{y_0}\right)^2\right)}$$

Fallback TTC estimates are typically conservative and ensure that collision risk assessment remains available even under degraded computation conditions. This approach preserves system safety and continuity of service.

5.4 Continuous Update Cycle

The TTC prediction framework operates within a continuous update cycle synchronized with the vehicle's perception and control loops. A typical update period is 25 milliseconds (40 Hz update rate), consistent with sensor fusion and motion planning pipelines in modern automotive architectures.

At each prediction cycle, the system performs a complete sequence of operations, beginning with the acquisition of vehicle and object states, including position, velocity, acceleration, heading, and yaw rate. Based on these states, an initial guess for the time-to-collision is generated. Subsequently, a gradient-based optimization procedure is executed to refine the TTC estimate; if convergence criteria are not met within the allowed iteration budget, a fallback calculation based on constant velocity assumptions is applied. Finally, the collision risk is evaluated by comparing the predicted minimum separation distance against a predefined safety threshold.

Continuous updates enable the system to account for rapidly changing environmental conditions, such as sudden braking maneuvers, lane changes, or dynamic obstacles, thereby enhancing the robustness and responsiveness of collision avoidance functions.

IMPLEMENTATION

1) Solver Configuration

The implementation was developed in Python using NumPy and Matplotlib libraries. Key solver parameters are configured as follows:

- Minimum separation distance threshold $d_{safe} = 2.0$ meters
- Maximum number of iterations per prediction cycle $N_{max} = 8$
- Initial time estimate $t_0 = 0$ seconds
- Step gain for updates $\alpha = 1.0$
- Convergence tolerance $\epsilon = 10^{-6}$ seconds
- Fallback sampling enabled upon convergence failure

The vehicle and target are initialized with global position, velocity, and acceleration vectors, assuming planar motion without lateral slip effects.

2) Motion Prediction and Relative Dynamics

The motion of both the host vehicle and the target object is predicted using second-order kinematic models that incorporate initial positions, velocities, and accelerations along the global X and Y axes. At each prediction time step, the future positions are calculated by integrating the initial velocity and acceleration over time.

The relative displacement between the host and the target is obtained by computing the difference between their predicted global positions. Similarly, the relative velocity is determined by subtracting the target's predicted velocity from that of the host.

The instantaneous separation distance is calculated as the Euclidean norm of the relative displacement vector. These quantities—relative displacement, relative velocity, and separation distance—serve as the foundation for evaluating the gradient and Hessian of the distance function with respect to time during the iterative collision time estimation process.

3) Gradient and Hessian Computation

The gradient and Hessian of the separation distance function $d(t)$ with respect to time play critical roles in guiding the optimization process toward the predicted time of minimum separation between the host and the target.

The gradient $\nabla d(t)$quantifies the instantaneous rate of change of the separation distance with respect to time. Specifically, it captures how the distance between the two vehicles will evolve if time progresses slightly forward. A positive gradient indicates that the vehicles are moving apart, while a negative gradient indicates that the vehicles are closing the gap. The gradient at a given time t is computed by projecting the relative velocity vector onto the relative displacement vector and normalizing by the current separation distance. This directional sensitivity is essential for determining whether a time update should move forward or backward in order to approach the minimum separation.

$$\nabla d(t) = \frac{\Delta x(t)\Delta v_x(t) + \Delta y(t)\,\Delta v_y(t)}{d(t)}$$

The $\nabla^2 d(t)$represents the second derivative of the separation distance with respect to time and characterizes the local curvature of the distance function. A positive Hessian implies that the local profile of $d(t)$ is convex (i.e., bowl-shaped), suggesting that a unique minimum exists nearby. Conversely, a negative Hessian indicates non-convex behavior, potentially leading to saddle points or local maxima. The Hessian computation combines the magnitudes of the relative velocity and the relative acceleration components, appropriately normalized by powers of the current distance.

The second derivative (Hessian) is given by:

$$\nabla^2 d(t) = \frac{\left(\Delta v_x(t)\right)^2 + \left(\Delta v_y(t)\right)^2}{d(t)} - \frac{\left(\Delta x(t)\Delta v_x(t) + \Delta y(t)\Delta v_y(t)\right)^2}{d(t)^3}$$

Both the gradient and Hessian are essential for performing efficient optimization:

- The gradient provides the descent direction along which the time estimate should be updated to reduce the separation distance.
- The Hessian enables the use of Newton's method when positive and well-conditioned, allowing for faster quadratic convergence toward the minimum. In cases where the Hessian is near zero or negative, the solver falls back to gradient descent to maintain stability.

Numerical safeguards are applied during gradient and Hessian evaluations to handle scenarios where the separation distance d(t) becomes very small, thereby avoiding division by near-zero quantities that could cause instability.

By leveraging both the first and second-order information of the separation distance function, the solver can more accurately and efficiently identify the predicted collision time, ensuring robust and real-time capable performance in dynamic driving environments.

4) Iterative Solver Procedure

The TTC estimation is formulated as a constrained iterative optimization problem. At each iteration:

a) The gradient and Hessian of the separation distance are evaluated.

b) A search direction is determined:

- If $\nabla^2 d(t)$is positive and sufficiently large, a Newton step is applied:

$$\Delta \mathrm{t} = -\frac{\nabla d(t)}{\nabla^2 d(t)}$$

- Otherwise, a reduced-gradient descent step is taken:

$$\Delta \mathrm{t} = -\nabla d(t)$$

c) A backtracking line search based on the Armijo condition is performed to adaptively select the step size.

d) The next time estimate t_{k+1} is projected onto the feasible domain $[0, T\_\mathrm{max}]$ to maintain physical plausibility.

Convergence is declared if either the separation distance falls below d_{safe} or the change in successive time estimates satisfies:

$$|t_{k+1} - t_k| < \epsilon$$

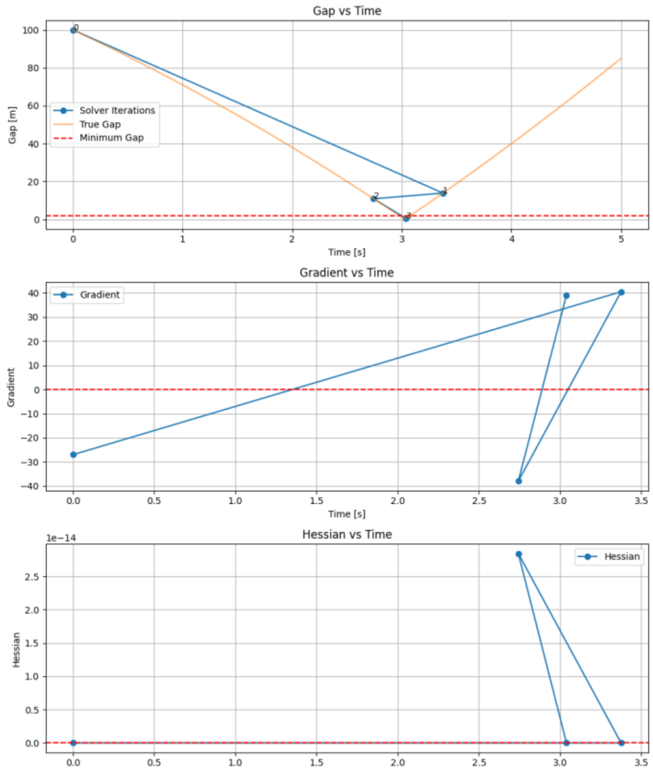
after a minimum number of iterations.

5) Fallback Strategy

If the optimization does not converge within the allowed number of iterations, a backup method is used to still estimate the collision time. Instead of continuing to iterate, the solver samples several times around the last estimated time and picks the sample where the separation distance is the smallest:

$$t^* = \underset{t\in[t_{curr}-\delta\,,\mathrm{t_{curr}}+\delta]}{\mathrm{argmin}} d(t)$$

where δ = 0.5seconds is the size of the sampling range. This fallback ensures that even if the solver struggles to find the best time exactly, the system can still predict a reasonable collision time based on nearby samples, keeping the collision avoidance system running safely.



The example code for trajectory optimization can be found in https://github.com/woa0425/Leo_ADAS_script/blob/main/Collision_Avoidance/Collision_Avoidance_python/Convex_Optimization_for_TTC_Prediction.py

The proposed optimization-based TTC estimation framework was applied to a scenario where the host vehicle traveled at an initial velocity of $20.0\frac{m}{s}$, with a longitudinal acceleration of $2.3\frac{m}{s^2}$, while the target object approached with an initial velocity of $-7.0\frac{m}{s}$ and an acceleration of $-1.7\frac{m}{s^2}$.

The iterative solver successfully converged to a predicted Time-to-Collision (TTC) of approximately **3.039 seconds**. The evolution of separation distance, gradient, and Hessian during the optimization process is shown in Figure 1.

- In the Gap vs. Time plot, the solver iterations effectively reduced the gap toward the minimum safe distance, achieving convergence within 4 iterations.
- The Gradient vs. Time plot shows that the gradient approached zero near the collision point, indicating a local minimum in the separation distance.
- The Hessian vs. Time plot remained near zero, suggesting that the problem was weakly convex or near-linear around the critical time.

The solver did not require fallback sampling, indicating stable convergence within the allowed iteration budget.
These results confirm that the optimization method can accurately and efficiently predict collision risks even under dynamic and non-trivial motion conditions.

REGULATORY AND FUNCITONAL SAFETY

Ensuring the functional safety of collision avoidance systems is a central requirement for autonomous vehicles and advanced driver-assistance systems (ADAS). As these systems assume greater responsibility for vehicle control, both industry standards and regulatory frameworks impose strict requirements to ensure predictable, reliable, and fail-safe behavior under a wide range of operational conditions. Among these frameworks, ISO 26262 stands as the primary international standard for automotive functional safety, providing structured guidelines for managing risk arising from potential hardware and software failures.

ISO 26262 defines a comprehensive lifecycle for the development of safety-related electrical and electronic systems within road vehicles. It introduces the concept of Automotive Safety Integrity Levels (ASIL), which classify the required rigor of safety measures based on the severity, exposure, and controllability of potential hazardous events. Collision avoidance functionalities, particularly those involving emergency braking, collision warning, and evasive maneuvering, are typically assigned high ASIL ratings (such as ASIL C or D) due to their direct impact on preventing life-threatening scenarios. As such, collision avoidance systems must demonstrate high levels of fault tolerance, systematic integrity, and residual risk reduction through both architectural design and software development processes.

For collision avoidance specifically, ISO 26262 emphasizes the need for early hazard analysis and risk assessment (HARA) to identify hazardous scenarios where loss or degradation of collision prediction could lead to critical outcomes. Safety goals are then derived to ensure timely detection of potential collisions, proper system intervention (e.g., braking or evasive steering), and safe state transitions in the event of system faults. Verification and validation processes under ISO 26262 require comprehensive analysis of both nominal and faulty behavior, including fault injection testing and confirmation of safety mechanisms that ensure degraded system modes still prevent unreasonable risk.

In addition to ISO 26262, collision avoidance performance is subject to consumer evaluation under New Car Assessment Program (NCAP) protocols. NCAP assessments typically evaluate the system's ability to autonomously initiate braking or steering to mitigate frontal collisions, pedestrian impacts, and cyclist crashes, providing safety ratings that influence public perception and commercial success. Furthermore, Federal Motor Vehicle Safety Standards (FMVSS) in the United States establish regulatory requirements for fundamental vehicle safety features, including crashworthiness and minimum braking performance, indirectly impacting the baseline expectations for collision mitigation capabilities.

Together, these standards and regulatory frameworks drive the design, validation, and deployment of collision avoidance systems, ensuring not only technical effectiveness but also systematic compliance with safety-critical operational demands. The predictive time-to-collision estimation approach proposed in this research is developed with these safety objectives in mind, providing a foundational capability to enable timely and reliable collision risk assessment.

CONCLUSION

This research presented an optimization-based framework for predictive Time-to-Collision (TTC) estimation, explicitly incorporating vehicle dynamics, relative motion modeling, and numerical optimization techniques to enhance collision risk prediction. By formulating the TTC prediction problem as a continuous-time separation distance minimization, the proposed solver leveraged reduced-gradient descent with Newton refinement to efficiently identify the critical time to minimum separation. A fallback strategy was also introduced to ensure robust performance under non-convex or degraded convergence conditions.

Simulation results demonstrated that the iterative solver could accurately predict collision timing under dynamic scenarios involving acceleration and deceleration of both the host and target vehicles. The method achieved convergence within a limited number of iterations and maintained computational efficiency suitable for real-time automotive applications. The ability to maintain continuous collision risk assessment, even under challenging motion profiles, enhances the safety and reliability of collision avoidance systems.

Moreover, the proposed TTC estimation approach aligns with functional safety standards such as ISO 26262 and regulatory assessment protocols like NCAP, supporting the development of advanced driver-assistance and autonomous vehicle systems that meet both technical and safety-critical requirements.

Future work may focus on extending the framework to incorporate lateral motion uncertainties, variable acceleration profiles, and probabilistic risk modeling to further enhance robustness in complex driving environments.

REFERENCES

[1] J. L. Meriam and L. G. Kraige, *Engineering Mechanics: Dynamics*, 7th ed., Wiley, 2012.
[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[3] G. Strang, *Linear Algebra and Its Applications*, 4th ed., Brooks Cole, 2006.
[4] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer, 2006.