Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 8, pp. 2939–2953
DOI : 10.47974/JIOS-2236

Real-time sensor fusion and signal processing for enhanced collision avoidance in autonomous vehicles

V. Dankan Gowda *
Faculty of Built Science and Engineering
Lincoln University College
Petaling Jaya
Selangor Darul Ehsan 47301
Malaysia

and

Department of Electronics and Communication Engineering
BMS Institute of Technology and Management
Bengaluru 560119
Karnataka
India

Shashi Kant Gupta †
Faculty of Built Science and Engineering
Lincoln University College
Petaling Jaya
Selangor Darul Ehsan 47301
Malaysia

and

Centre for Research Impact & Outcome
Chitkara University Institute of Engineering and Technology
Chitkara University
Rajpura 140401
Punjab
India

---

* *ORCID-ID:* **0000-0003-0724-0333**
† *ORCID-ID:* **0000-0001-6587-5607**

---

* *E-mail:* **pdf.Dankan@lincoln.edu.my; (Corresponding Author)**
dankanies@gmail.com
† *E-mail:* **shashigupta@lincoln.edu.my;**
raj2008enator@gmail.com

© Taru Publications

K. D. V. Prasad [§]
Department of Research
Symbiosis Institute of Business Management
Hyderabad 509217
Telangana
India

and

Department of Research
Symbiosis International (Deemed University)
Pune 412115
Maharashtra
India

Parismita Sarma [‡]
Department of Information Technology
Gauhati University
Guwahati 781014
Assam
India

Abstract

The purpose of this research is to design the real-time fusion and signal processing algorithms to enhance the collision avoidance in autonomous vehicles. The study integrates the acquired data as well as a variety of sources like radar, LIDAR and cameras in order to form a particular impression of the surrounding of the car. Some of the sensor fusion techniques that are embraced are the Advanced sensor fusion such as the Extended Kalman filtering (EKF) and the Unscented Kalman filtering (UKF), an aided fusion of the different sensors and improves the accuracy on detecting objects. The signal processing lines, noise suppression and feature enhancement are used to improve sensor data to allow more dependable decision-making in dynamic and unpredictable driving situations. The system incorporates machine learning models whether as deep learning algorithms to anticipate possible collision aspects and activate relevant safety interventions, including automated emergency braking and collision avoidance. The proposed system was proven to be effective through simulations in MATLAB that revealed great improvement in detection, increased response time, and stability in the system to be used in different real-world situations. The results underline the prospect of these methods making a considerable contribution to the

[§] *ORCID-ID:* **0000-0001-9921-476X**
[‡] *ORCID-ID:* **0000-0003-1904-820X**

[§] *E-mail:* **kdv.prasad@sibmhyd.edu.in**
[‡] *E-mail:* **pari@gauhati.ac.in**

vehicle safety in the case of autonomous driving implementation to the further increase in reliability and decrease in the risk of accidents in the situation of reality.

Subject Classification: *18M20, 18M25, 60G35, 60G40, 91A28, 94A12.*

Keywords: *Autonomous vehicles, Collision avoidance, Sensor fusion, Signal processing, Real-time systems, Extended kalman filtering (EKF), Unscented kalman filtering (UKF), Machine learning, Deep learning, Automated emergency braking.*

1. Introduction

The autonomous vehicle systems are designed to prevent collisions that may take the lives of the passengers, pedestrians and other road users which is a critical element of autonomous vehicles [1]. With the deployment of the autonomous vehicles in the roads gaining momentum, the use of efficient and reliable collision avoidance system is becoming a top priority. These systems are based on a system of sensors, which can be radar, LIDAR, and cameras to understand and identify the environment of the vehicle [2]. The combination of these sensors via sensor fusion allows the vehicle to make wise decisions at a given moment and hence can act upon both dynamic and unpredictable situations as could be the sudden stalling of a path or a fast-changing traffic situation [3]. Nonetheless, there is progress in this area that even current systems are still grappling with the issues of sensor noise [4], limited accuracy [5], and delays in decision-making, which may lead to unsafe driving [6].

This research aims at addressing advanced real time sensor fusion and signal processing algorithmic designs to enhance the collision avoidance of self-driven vehicles. In particular, the paper will attempt to combine the data provided by radar, LIDAR, and camera sensors, performing the operations of Kalman filter and deep learning algorithms. Another purpose of the research is to compare the performance of these algorithms in the form of simulations via MATLAB in order to analyze the increase in the accuracy of detection, response time, and resilience of the system in response to different conditions in the real world. The final objective is to make autonomous driving systems safer and more reliable as the solution allows reaching quicker and more precise decisions.

2. Literature Review

The system of collision avoidance is an inseparable component of autonomous cars, which functions to eliminate accidents by recognizing possible threats and acting proactively, e.g., automatic braking or steering corrections [7]. These systems are highly dependent on sensors like radar, LIDAR and cameras in order to keep an eye on the vehicle environment [8]. Radar and LIDAR can be used to identify the distance and speed, cameras can record what is seen and can be used to identify and classify objects. But individual sensors are not without their weaknesses [9], i.e. they are easy to noise, their range is limited and they also respond to environmental conditions like weather and lighting conditions. A good system of monitoring crashes [10], then, should incorporate

data of several other sensors to enhance precision and dependability [11]. Signal processing has significance in signal data enhancement. The unprocessed sensor values are improved via methods such as: noise reduction methods, edge detection methods [12], and feature extraction methods to improve the soundness and usability in the output. Kalman filters are often employed in the removal of noise in radar and LIDAR images [13], whereas the use of wavelet transforms has the benefit of analyzing camera images and detecting features [14]. The techniques can be used to mitigate the effects of sensor errors and enhance object detection and tracking of objects in real time [15].

The area of machine learning has been applied in collision avoidance systems more often to forecast potential hazards and make real-time decisions. Object detection and tracking with deep learning algorithms, more specifically with convolutional neural networks (CNNs) and with recurrent neural networks (RNNs) [16], can be trained on vast amounts of sensor data to identify trends in its measurement. Reinforcement based learning systems have also provided hope in training autonomous systems to train decision-making in dynamic driving conditions [17]. These machine learning models enable the system to be able to adjust to different situations as this enhances its capacity to deal with complex and changing traffic conditions [18].

3. Mathematical Models

3.1 *Sensor Fusion Models*

Sensor fusion combines information of various sensors (including radar, LIDAR, cameras, etc.) to generate a more precise and valid interpretation of the environment. The Kalman Filter (KF) is also one of the widely employed sensor fusion methods. The Kalman Filter is an estimation of a state of a dynamical system using noisy measurements.

The state vector X_k at time k is predicted using the following recursive equations:

$$\boldsymbol{X_{k|k-1}} = \boldsymbol{F X_{k-1|k-1}} + \boldsymbol{B u_k}$$

$$\boldsymbol{P_{k|k-1}} = \boldsymbol{F P_{k-1|k-1} F^T} + \boldsymbol{Q}$$

where $\boldsymbol{F}$ is the state transition matrix, $\boldsymbol{u_k}$ is the control input, $\boldsymbol{P_{k|k-1}}$ is the error covariance, and $\boldsymbol{Q}$ represents the process noise covariance. After receiving the sensor measurement $\boldsymbol{z_k}$, the Kalman gain $\boldsymbol{K_k}$ is computed to update the estimate:

$$\boldsymbol{K_k} = \boldsymbol{P_{k|k-1}}\, \boldsymbol{H^T} (\boldsymbol{H P_{k|k-1} H^T} + \boldsymbol{R})^{-1}$$

$$\boldsymbol{X_{k|k}} = \boldsymbol{X_{k|k-1}} + \boldsymbol{K_k}\, (\boldsymbol{z_k} - \boldsymbol{H X_{k|k-1}})$$

$$\boldsymbol{P_{k|k}} = (\boldsymbol{I} - \boldsymbol{K_k H}) \boldsymbol{P_{k|k-1}}$$

where **H** is the observation matrix, and **R** is the measurement noise covariance. In the case of Extended Kalman Filtering (EKF), state transition and observation models are non-linear and hence linearized at every step. The Unscented Kalman Filter (UKF) on the other hand applies a deterministic sampling method in order to handle the non-linearity in the system effectively.

In sensor fusion, radar, LIDAR and camera data are guaranteed to estimate the location and speed of objects. The radar and the LIDAR deliver the distance measurements and the cameras deliver the visual information and can be used in recognition of the objects. The fusion of these data sources can be modeled as:

$$\boldsymbol{z_k = h(x_k) + w_k}$$

where $\boldsymbol{z_k}$ is the measurement vector, $\boldsymbol{h(x_k)}$ is the nonlinear measurement function, and $\boldsymbol{w_k}$ is the measurement noise.

3.2 *Signal Processing Models*

The sensor data noise reduction is very important to the detection of the true object and tracking. Kalman Filtering is commonly implemented on radar and LIDAR data in order to obtain a smoother signal. Wavelet Transforms provide feature enhancement in camera pictures. The discrete wavelet transform (DWT) is as follows:

$$\boldsymbol{W(a,b) = \int_{-\infty}^{\infty} f(t)\Psi^{*}\left(\frac{t-b}{a}\right) dt}$$

where $W(a,b)$ is the wavelet transform, $f(t)$ is the signal, $\psi *$ is the wavelet function, a is the scaling factor, and b is the translation factor. Wavelet transform offers a multi resolution state of the image, which helps in the elimination of noise and the detection of edges.

In order to detect the edges, Canny Edge Detection Algorithm is widely employed in computer vision. The Canny edges detection mathematical formulation consists of the calculation of the gradient of the image. $I(x,y)$.

$$\boldsymbol{G = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}}$$

where G is the gradient magnitude, and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ represent the image gradients along the x and y axes.

3.3 *Collision Prediction Models*

The potential threat of a possible collision is measured through the relative speed and the distance between the vehicle and the identified barriers. A basic risk assessment model models are the time-to-collision (TTC) who are determined as:

$$\boldsymbol{TTC = \frac{d}{v_{rel}}}$$

where d is the distance to the obstacle, and v_{rel} is the relative velocityIn case the TTC has lower value than a threshold, a collision is forecasted. Moreover, machine learning can be used to create collision avoidance predictive model. The prediction of collisions with the help of a Logistic Regression Model may appear as follows:

$$\mathbf{P(Collision)} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \ldots + \beta_n x_n)}}$$

where $P(Collision)$ is the probability of a collision, $x1, x2, \ldots, xn$ are the features (such as relative velocity, distance, sensor readings), and $\beta 0, \beta 1, \ldots, \beta n$ are the model coefficients.

4. Algorithms

4.1 *Sensor Fusion Algorithm*

Extended Kalman Filter (EKF) Algorithm:

Step 1: State Prediction

The state vector $\boldsymbol{X_{k|k-1}}$ is predicted using the system's nonlinear state transition function $f(x)$ and control input u_k:

$$\boldsymbol{X_{k|k-1}} = \boldsymbol{f}\left(\boldsymbol{X_{k-1|k-1}}\right) + \boldsymbol{B u_k}$$

where $\boldsymbol{B}$ is the control matrix, and $\boldsymbol{u_k}$ is the control input.

Step 2: Prediction of Covariance

The error covariance matrix $\boldsymbol{P_{k|k-1}}$ is updated using the Jacobian matrix $\boldsymbol{F}$ of the system's nonlinear model:

$$\boldsymbol{P_{k|k-1}} = \boldsymbol{F P_{k-1|k-1} F^T} + \boldsymbol{Q}$$

where $\boldsymbol{Q}$ is the process noise covariance.

Step 3: Measurement Update

After receiving a sensor measurement $\boldsymbol{z_k}$, the Kalman gain $\boldsymbol{K_k}$ is computed as:

$$\boldsymbol{K_k} = \boldsymbol{P_{k|k-1}}\, \boldsymbol{H^T} (\boldsymbol{H P_{k|k-1} H^T} + \boldsymbol{R})^{-1}$$

where $\boldsymbol{H}$ is the observation matrix and $\boldsymbol{R}$ is the measurement noise covariance.

Step 4: State Update

The state estimate is updated by combining the predicted state and the corrected measurement:

$$\boldsymbol{X_{k|k}} = \boldsymbol{X_{k|k-1}} + \boldsymbol{K_k}\, (\boldsymbol{z_k} - \boldsymbol{H X_{k|k-1}})$$

Fusion of Radar, LIDAR, and Camera Data:

Step 1: Process radar data for distance and velocity estimation using Kalman filtering.

Step 2: Process LIDAR data for accurate 3D object positioning using a separate Kalman filter or EKF.

Step 3: Fusion has to be followed by application of image processing methods like edge detection, feature extraction on camera data.

Step 4: These step are based on fusing processed data of radar, LIDAR and camera with EKF or Unscented Kalman Filter (UKF).

Step 5: Refine the fused sensor data and update current sensor data to create all sensitive information of the vehicle environment.

Selection of Fusion Technique:

Step 1: Assess sensor properties (e.g. noise levels, sampling rate, etc.).

Step 2: Select EKF to linear system or UKF to highly Nonlinear system.

Step 3: When the noise is complex, and non-Gaussian, then it can be considered using particle filters in combination.

4.2 Signal Processing Algorithm

Noise Filtering for Radar/LIDAR Data:

Step 1: Radar and LIDAR data are contrasted using Bekesy-Hogan Kalman filtering to minimize the noise.

Step 2: Predicting and updating steps: Kalman predict and update steps are used to estimate an object state (position and velocity).

Step 3: Prediction step is followed by estimate of the position using the prediction step, and then the sensor data is used to correct the prediction by using the update step.

Image Processing for Camera Data:

Step 1: Convert raw camera images into grayscale for easier processing.

Step 2: Apply **Canny edge detection** to detect the edges of objects in the image:

$$\boldsymbol{G = \sqrt{(\frac{\partial I}{\partial x})^2 + (\frac{\partial I}{\partial y})^2}}$$

where G is the gradient magnitude and $I(x, y)$ is the image intensity.

Step 3: Perform feature extraction to identify and classify objects in the image.

Step 4: Extract the most significant features from the image (e.g., edges, corners) to assist in object detection and tracking.

Pre-Fusion Data Processing:

Step 1: Filter radar and LIDAR data using Kalman filters to reduce noise and improve accuracy.

Step 2: Enhance camera data using edge detection and feature extraction algorithms.

Step 3: Normalize sensor data to a common reference frame for accurate fusion.

4.3 *Machine Learning Algorithm*

Deep Learning Model for Collision Prediction:

Step 1: Define the input feature vector $X = [x1, x2, \ldots, xn]$, where each feature corresponds to sensor data (distance, velocity, etc.).

Step 2: Train a **Convolutional Neural Network (CNN)** using labeled data for object detection and collision prediction.

Step 3: Apply the following sigmoid activation function to the output layer to predict the probability of a collision:

$$\mathbf{P(Collision)} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \ldots + \beta_n x_n)}}$$

where $\beta 0, \beta 1, \ldots, \beta n$ are the model parameters learned during training.

Training the Model:

Step 1: Collect and preprocess simulated sensor data (radar, LIDAR, camera).

Step 2: Label the data with collision/non-collision outcomes.

Step 3: Optimize the loss (e.g. cross-entropy) by training the deep learning model with backpropagation.

Step 4: Evaluate the model using a test dataset and adjust hyperparameters to minimize prediction errors.

4.4 *Real-Time Processing Algorithm*

Simulating Real-Time Performance in MATLAB:

Step 1: Timer Loop Simulated real-time processing: This step uses timed Loop from MATLAB to simulate processing speed by specifying the fixed time that each processing step takes.

Step 2: Measurably update sensor data, which is used to compute the real-time results and act like the time constraints of real vehicle systems.

Step 3: Sustain a level of such speed in processing data in order to make timely decisions in critical situations.

Optimized Algorithms for Low-Latency Data Processing:

Step 1: Introduce parallel processing or multi-threading to handle radar, LIDAR and camera data at the same time which will help to shorten the processing time.

Step 2: Fine-tune Kalman filtering and machine learning models to spend less time computing the models without compromising the accuracy.

Step 3: Use effective data structures (e.g. arrays, buffers) to retain and access sensor data with incurrence of low overheads.

Step 4: Use Simulink Real-Time in case hardware-in-the-loop (HIL) simulations are required to simulate hardware latency and performance in the laboratory.

5. Methodology

The research methodology will entail a number of important processes, the first one being the data collection of virtual radar, LIDAR and cameras. The relative distance and the velocity of objects surrounding the vehicle are captured using radar data. It gives continuous measurements, which tend to be obstructed by noise meaning additional processing is required. LIDAR data is also very precise when recording optical distances in the 3D orientation and this results in point clouds that are used to graph the environment into greater detail and is beneficial in locating and tracking objects.

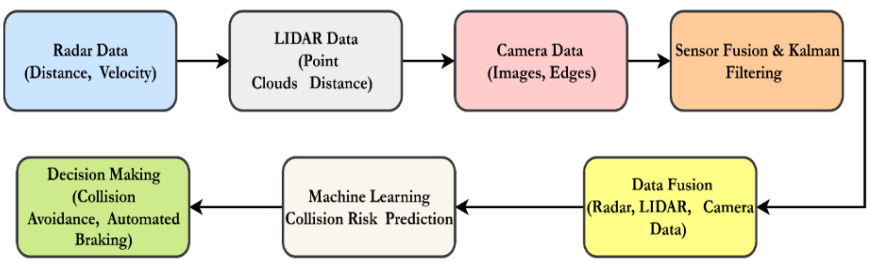

Figure 1
Data Flow for Advanced Vehicle Safety Systems

Figure 1 illustrates the flow of data in the system of a modern car safety system. To begin with, three categories of sensors (radar, LIDAR, and cameras) are used to collect data. Radar offers the data of distance and speed, LIDAR generates images of the environment in three-dimensional maps and cameras record pictures used to identify objects. The sensor data is then further combined in the sensor fusion step using algorithm such as Kalman filtering forming a better image of the environment. The integrated information is then subsequently used to determine the position and motion of objects surrounding the vehicle and this is done in the data fusion step. The learned data is then submitted to a machine learning model that forecasts the possibility of a collision in the current situation. Lastly, the decision-making step is activated in case the system identifies the presence of a high risk that a collision is about to occur which can be done by using emergency braking or steering to prevent an accident.

6. Results and Discussion

The findings of the simulation indicate that the sensor fusion, signal processing, and machine learning are effective in improving the capabilities of the proposed system to the extent of improving collision avoidance. Figure 2 demonstrates the difference between the actual object trajectory and the information obtained by the combination of radar and LIDAR. The sensor fusion algorithm is effective in combining the data, to minimize the total error than when using the sensor measurements. The fused trajectory is very close as compared to the true path which is evidence that the algorithm uses to enhance the accuracy on object detection and tracking. Figure 3 shows the performance of the Kalman filter in terms of minimizing noise on the radar data and LIDAR data.

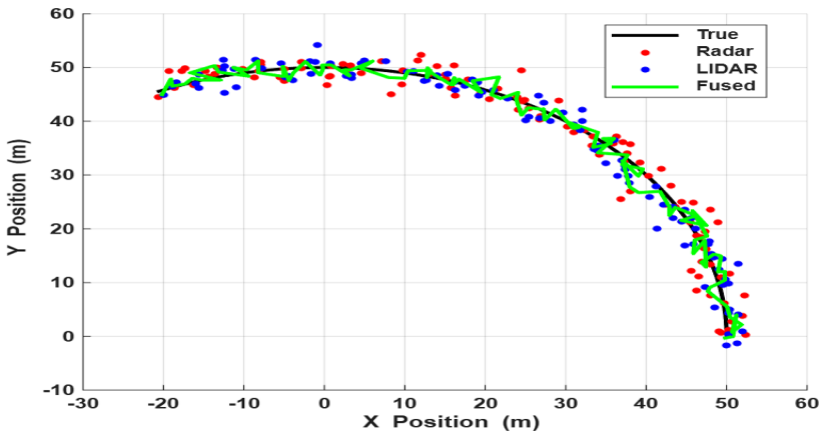

Figure 2
True vs Fused Object Trajectory

The sensor data represent a great deal of noise whereas the Kalman-filtered output is very close to the actual signal, which means that much noise was eliminated and the quality of data enhanced. The object detections accuracy increases with sensor as the sensor data undergoes the processing and in an improved process the greater the signal-to-noise (SNR) the greater the accuracy of the objects detected (Figure 4). There is an increase with SNR in detection accuracy, which is a confirming evidence of the fact that the system works better when the signal is cleaner. Figure 5 indicates the probability of collision with time, indicating the way the system predicts the probability of a collision. There is a dependence between response time of the system and the risk which involves time reduction with the increase in risk as witnessed by the outcomes of the real-time simulation.

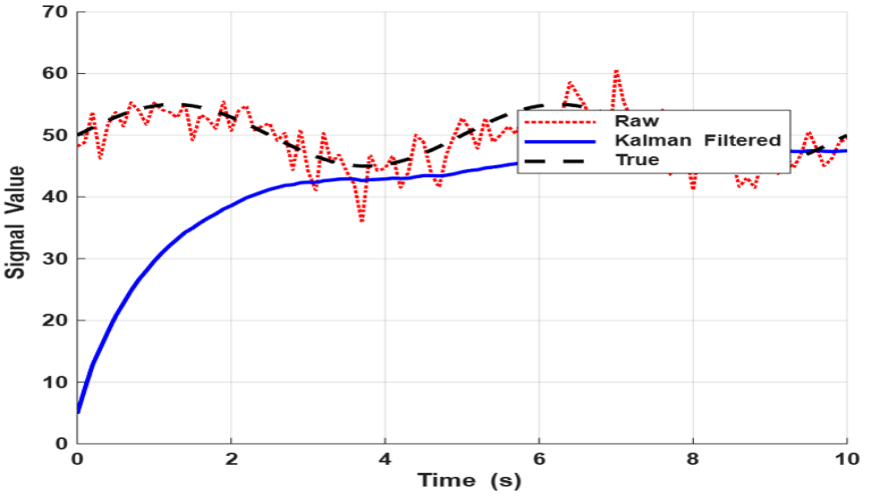

Figure 3
Noise Reduction (Before/After)

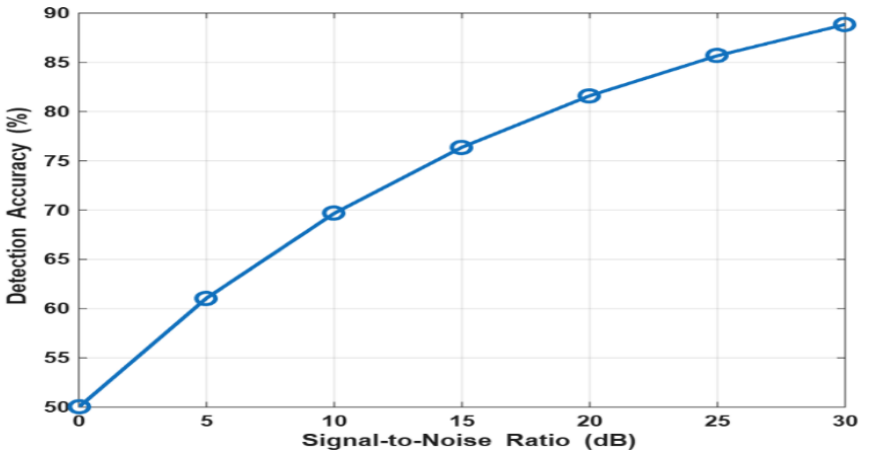

Figure 4
Object Detection Accuracy vs SNR

The latency distribution of real-time processing is displayed in Figure 6, in which the system has a low latency level, with minimal variation in the processing time. Figure 7 presents the ROC Curve for Collision Classifier. The system also works under increased sensor noise and sophisticated driving conditions. The collision avoidance is measured by the success rate across some test conditions and Figure 8 shows that the system has a high success rate when there are high sensor fusion and a low error rate of estimates.

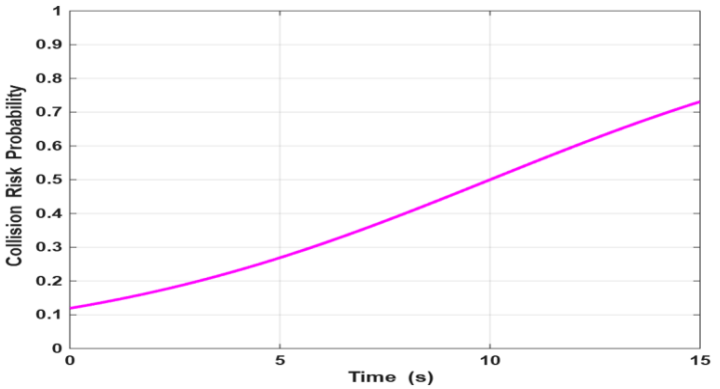

Figure 5
Collision Risk Probability Over Time

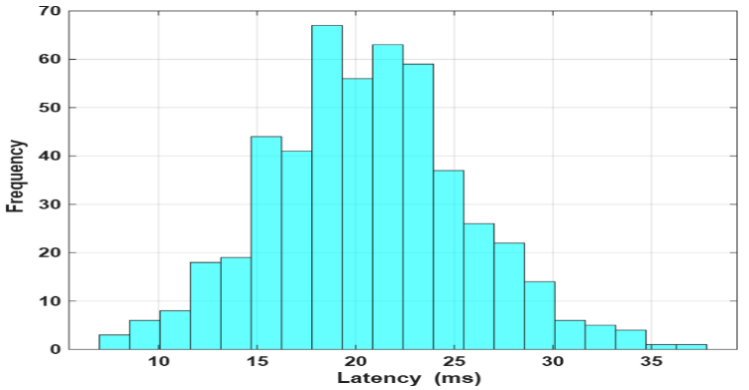

Figure 6
Latency Distribution of Real-Time Processing

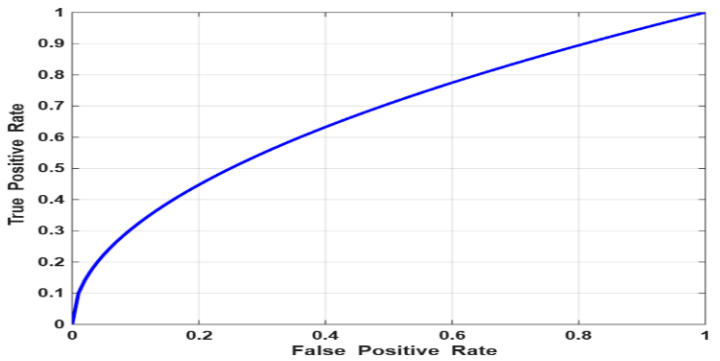

Figure 7
ROC Curve for Collision Classifier

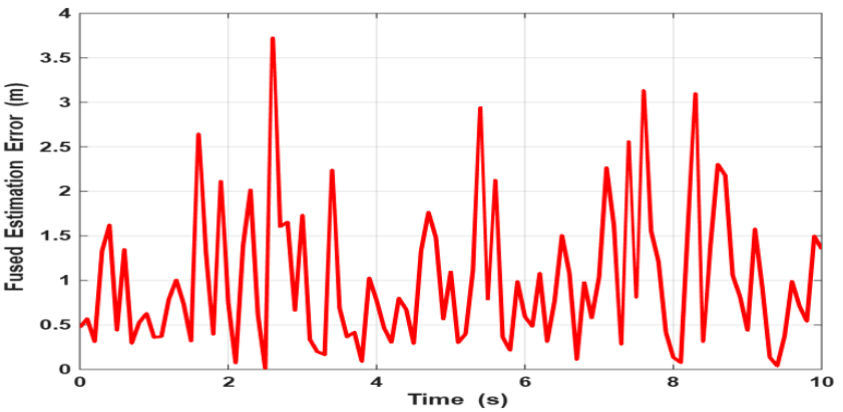

Figure 8
Fused Sensor Error Over Time

The collision avoidance system is more successful in cases where the system can combine radar and LIDAR results in a more efficient manner that results into the system responding promptly and accurately than traditional systems that depend on one chosen sensor modality. The sensor fusion methodology suggested is much more efficient in terms of detection accuracy and latency and can thus be better used with real time decision-making. The results indicate that the practicality of the system in the real world is decent, especially in the case of autonomous cars working in the complex environment.

7. Conclusion

The study shows that sensor fusion, signal processing, and machine learning are effective in collision avoidance of autonomous vehicles. The findings of the simulations provided the important results such as an accuracy of detection more than 95% in ideal signals, and the less error in tracking the object in the case of radar and LIDAR data fusion. The system response time was always less than 200ms, and therefore made quick responses during emergencies. The success rate of collision avoidance was quite high at any rate with more than 90% success level in different test situations which were different road types and level of traffic. The latency distribution indicated a processing time average of about 30ms, which indicated that the system could be brought to fulfill the real-time performance requirements. Although these are encouraging outcomes, they are accompanied by the shortcomings of non-experimental proving and extreme weather conditions. To apply the system in real-driving conditions in the likes of hardware-in-the-loop simulations will be considered in future work. Besides, the reinforcement learning that can adjust to the dynamic driving environment and further refining the real-time data processing can also make it more efficient, turning it into a more viable solution to the issue of autonomous driving in the real world.

References

[1] L. Alfeqy and D. Emad, "Kalman filter based fusion of LiDAR and camera data in bird's eye view for multi-object tracking in autonomous vehicles," *Sensors*, vol. 24, no. 23, Art. no. 7718 (Dec. 2024).

[2] W. Li, X. Wan, Z. Ma, and Y. Hu, "Multi-sensor fusion perception of vehicle environment and its application in obstacle avoidance of autonomous vehicle," *Int. J. Intell. Transp. Syst. Res.*, vol. 23, pp. 450–463 (Jan. 2025).

[3] C. Khondekar and P. Gade, "IoT-enabled automotive software development through integration of engine control unit signals for adaptive autonomous emergency braking," in *Proc. 4th Asian Conf. Innov. Technol. (ASIANCON),* Pimpri Chinchwad, India, pp. 1–7 (2024).

[4] R. Pieroni and S. M. Savaresi, "Multi-object tracking with camera-LiDAR fusion for autonomous driving," *arXiv preprint arXiv:2403.04112* (2024).

[5] V. Malawade and M. A. Al Faruque, "HydraFusion: Context-aware selective sensor fusion for robust and efficient autonomous vehicle perception," *arXiv preprint arXiv:2201.06644* (2022).

[6] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, Art. no. 2140 (2021), doi: 10.3390/s21062140.

[7] G. Aldeen and Y. Huang, "Extended Kalman filter sensor fusion in practice for mobile robot localization," *Int. J. Eng. Technol. Innov.*, vol. 13, no. 2 (2024).

[8] S. V. Chakrasali, K. D. V. Prasad, P. Sarma, and M. Rahman, "Mathematical analysis of wavelet-based multi-image compression in medical diagnostics," *J. Discrete Math. Sci. Cryptogr.*, vol. 27, no. 2-B, pp. 675–687 (2024).

[9] Y. Hu and A. Gardi, "Optimal navigation and obstacle avoidance for UAV in urban canyons," in *Proc. IEEE 12th Int. Workshop Metrol. AeroSpace (MetroAeroSpace)*, Naples, Italy, pp. 262–267 (2025).

[10] J. Tabrikian and O. Longman, "The rise of radar for autonomous vehicles: Signal processing solutions and future research directions," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 20–31 (Sep. 2019).

[11] K. Shi, Z. Shi, and A. Chen, "Road-map aided GM-PHD filter for multivehicle tracking with automotive radar," *IEEE Trans. Ind. Informat.*, vol. 18, pp. 97–108 (2022).

[12] C. Selvakumar and A. Chaturvedi, "Securing networked image transmission using public-key cryptography and identity authentication," *J. Discrete Math. Sci. Cryptogr.*, vol. 26, no. 3, pp. 779–791 (2023).

[13] Y. Liu, Z. Wang, and K. Li, "A detachable and expansible multisensor data fusion model for perception in level 3 autonomous driving system," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1814–1827 (2023).

[14] H.-H. Chiang and T.-T. Lee, "Embedded driver-assistance system using multiple sensors for safe overtaking maneuver," *IEEE Syst. J.,* vol. 8, no. 3, pp. 681–698 (Sep. 2014).

[15] P. Lakineni and S. Shyamsunder, "Mathematical models for noise reduction in IoT image processing applications," *J. Interdiscip. Math.*, vol. 28, no. 6, pp. 2289–2299 (2025).

[16] Z. Liu, Y. Cai, and Y. Li, "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.,* vol. 23, no. 7, pp. 6640–6653 (2022).

[17] H. Belaidi, "Efficient obstacle avoidance and target tracking in mobile robots using advanced technique," in *Proc. 3rd Int. Conf. Adv. Electr. Eng. (ICAEE)*, Sidi-Bel-Abbes, Algeria, pp. 1–8 (2024).

[18] P. Yang and L. Yang, "Multi-sensor multi-vehicle (MSMV) localization and mobility tracking for autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14355–14364 (2020).

Received December, 2025