Journal of Student Research Volume 12 Issue 3 (2023)

Model-Based Design Approach for Validation of Vehicle Sensor Fusion Algorithm

Ashley Phan[1], Brandon Stevens[1], Benjamin Fitzpatrick[1], Jordan Olson[1], Hwan-Sik Yoon[1#] and Paul Puzinauskas[1#]

[1]The University of Alabama
[#]Advisor

ABSTRACT

In recent years, the development of connected and autonomous vehicle (CAV) systems has accelerated dramatically as these systems can provide considerable benefits for vehicle performance and passenger safety. However, CAV systems can prove hazardous if not properly tested before deployment. A need for testing methods that are economical, time-effective, and low-risk has created the concept of model-based design, whereby systems are modeled and tested in various simulation environments before being fully deployed to the real system. This paper describes a model-based testing approach that has been developed to verify and validate a sensor fusion algorithm for CAV systems to enhance autonomous controls of a 2019 Chevrolet Blazer as part of the EcoCAR Mobility Challenge (EMC). Teams of undergraduate and graduate students with automotive interests laid out testing procedures to assess model fidelity and to identify and resolve issues with the algorithm before deployment to a student-adapted prototype vehicle. Students learned and applied complex engineering concepts rapidly to develop a sensor fusion and tracking algorithm.

1. Introduction

Ground transportation is rapidly changing today with advanced propulsion systems and autonomous features pushing towards a future of “zero crashes, zero emissions, and zero congestion” [1]. Along with new technologies, Mobility as a Service (MaaS) is now becoming more convenient and affordable for consumers due to its integration with the connected world [2]. Personal MaaS is currently a niche market but will likely dominate the urban transportation space in the future with projected compounded annual growth rates of 31.5% over the next few decades [3].

For over 30 years, Argonne National Laboratory (ANL) in the United State has managed multi-year advanced vehicle technology competitions [4]. They currently manage the EcoCAR Mobility Challenge (EMC) with support from General Motors (GM), MathWorks, the U.S. Department of Energy, and over 25 other industry sponsors. The current competition focuses on electrification, connectivity, and active driver assistance. The University of Alabama (UA) competes alongside ten other North American universities in the program, which challenges them to integrate and optimize an advanced propulsion system for a 2019 Chevrolet Blazer. EMC also tasks the participating universities with developing and implementing Level 2 autonomous features. The Society of Automotive Engineers (SAE) defines Level 2 autonomous features as those that enable steering and braking/acceleration to support the driver [5]. As part of the competition, ANL hosts student trainings led by ANL engineers or one of the various industry sponsors.

The UA team is made up of 65 students ranging in academic year from Freshman to PhD candidates. Five graduate students are fully funded through the program to lead student learning activities and engineering exercises to accomplish competition goals. Most students participate without receiving formal course credit, but several academic courses have been developed and adapted from ECM activities at the university. Students are recruited to join the engineering team at the start of the academic year across all majors within the College of Engineering. Figure 1 shows the demographic makeup of the team.

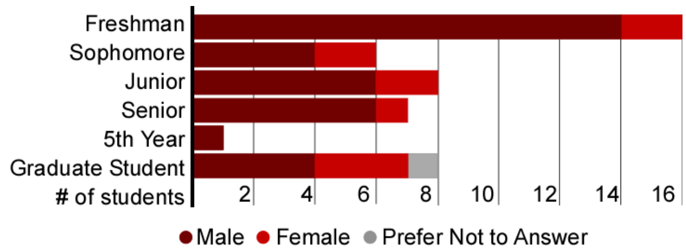

Figure 1. Team Demographics.

The four-year program follows GM's global vehicle development plan and began in the fall of 2018 when each team identified a geographic region with high MaaS growth potential. The teams defined vehicle technical specifications (VTS) driven by the needs of their unique market for a fleet-owned and customer-driven vehicle in the MaaS economy. In order to support the VTS goals, new hybrid propulsion systems were designed, and sensors and computational hardware were selected and integrated which enable the connected and autonomous vehicle (CAV) features.

One CAV feature implemented by the UA EMC student team is the adaptive cruise control (ACC) feature. ACC acts to maintain a set velocity of a vehicle, similarly to conventional cruise control, but adds the ability to deviate from the commanded set point to maintain a safe distance and relative velocity from other vehicles in its path [6, 7].

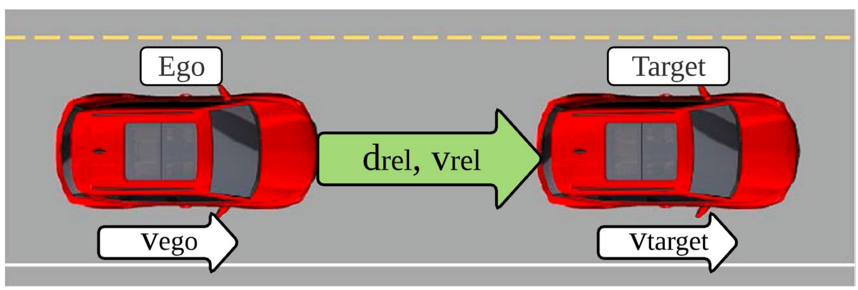

Figure 2. Definition of Ego and Target Vehicles.

As shown in Figure 2, the controlled vehicle is known as the ego vehicle, and other vehicles in its path are known as target vehicles. In the figure, v_{ego} and v_{target} are the velocities of the ego and target vehicles, respectively, and d_{rel} and v_{rel} are the distance and velocity, respectively, of the target vehicle relative to the ego vehicle. Functionality of the ACC feature relies on having an accurate mapping of the environment. This is achieved by using a sensor fusion algorithm as an input to the system. Sensor fusion is the process of aggregating object detections from multiple sensors into a fused detection for each object [8, 9]. The sensor fusion algorithm is thus able to attain a more accurate mapping of the environment than if the sensors were used separately.

This paper details the development and testing processes of the sensor fusion algorithm by the students along with the educational outcomes. These processes, elicited from model-based design, evaluate the performance of the algorithm across various testing environments. Test results are analyzed to improve the algorithm, forming a closed-loop, iterative development process that spans from initial requirements through vehicle deployment.

Although the basic concepts and simple examples of the different levels of testing environments can be found in the literature, their application details are scarce and hard to find. By showing how the concepts can be applied to the validation of an autonomous feature in a vehicle, the paper will serve as a good reference for the students and engineers interested in this field.

2. Description of System and Approach

2.1 Model-Based Design Development Process

Model-based design (MBD) is a model-centric approach to the development of dynamic systems including vehicles [10, 11]. A core concept of MBD is continuous testing at all stages of development, which can help identify issues early and increase overall understanding of the system. This concept can be represented by a diagram known as the V-cycle. UA's implementation of the V-cycle is shown in Figure 3.

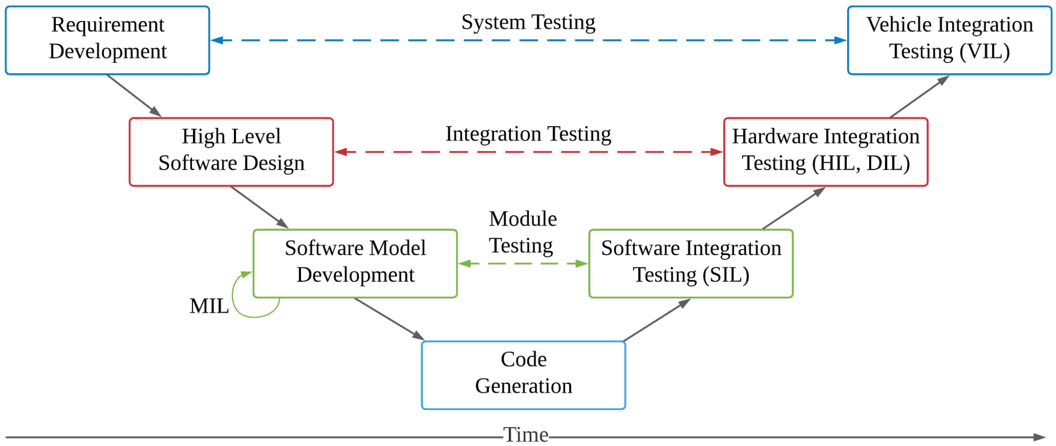

Figure 3. V-Cycle for the Development of UA CAV System.

A new feature begins as a set of requirements, and is progressively developed until it is ready to be tested in a simulation environment via model-in-the-loop (MIL) testing [12]. Once functionality has been confirmed in the simulation environment, the feature is incrementally introduced into hardware through some combination of the software-in-the-loop (SIL), hardware-in-the-loop (HIL), driver-in-the-loop (DIL), and finally vehicle-in-the-loop (VIL) environments [13, 14]. These environments and their testing capabilities are shown in Table 1.

Table 1. Comparison of Different Testing Environments.

Testing capabilities	**MIL**	**SIL**	**HIL**	**DIL**	**VIL**
Faster than real time	✓	✓			
Automated/iterative testing	✓	✓	✓		
Testing of unsafe situations	✓	✓	✓	✓	
Real-time execution		✓	✓	✓	✓
Generated code		✓	✓		✓
Communication interface			✓		✓
True vehicle response					✓
Driver acceptance				✓	✓

The V-cycle demonstrates a practical progression of new features from development to deployment. A version of this V-cycle is used by the UA student team for the development of the sensor fusion algorithm as discussed in later sections.

2.2 Hardware Architecture

The UA team's CAV perception system, shown in Figure 4, consists of Intel's Mobileye 6 as the front camera-based vision system, one Bosch mid-range radar (MRR) in the front, and two Bosch rear MRRs placed at the rear corners. The communication-enabling hardware includes Cohda Wireless's MK5 on-board unit (OBU) and the MobileMark MGWG-303 antenna. This hardware was donated by EMC sponsors and is required for use by EMC competition rules. Table 2 lists the detection range and field of view (FOV) specifications for the Intel Mobileye camera and Bosch MRRs. Figure 5 shows the front perception system's FOV for the Mobileye and the front radar's two antennas: a main antenna and an elevation antenna.

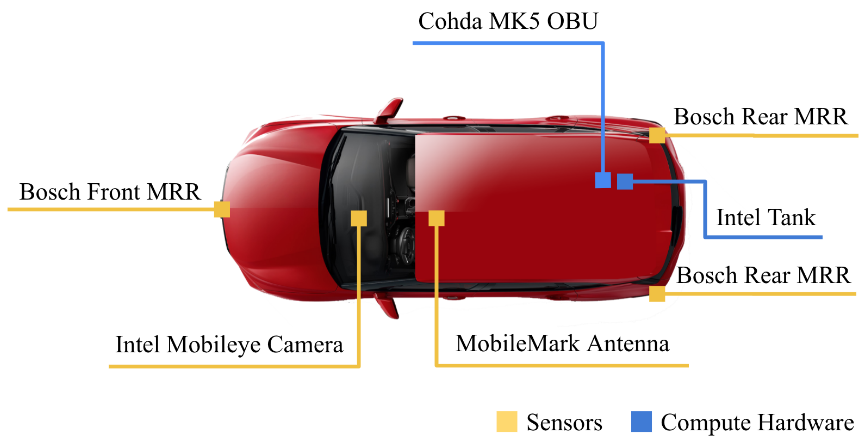

Figure 4. UA CAV System Hardware.

Table 2. Front Perception System Sensor Specifications.

Sensor	**Field of View (FOV)**	**Detection Range**
Intel Mobileye 6	Width: 38° Height: 30°	150 m
Bosch Front MRR	Main antenna: 12° Elevation antenna: 84°	160 m 12 m
Bosch Rear MRR	150°	80 m

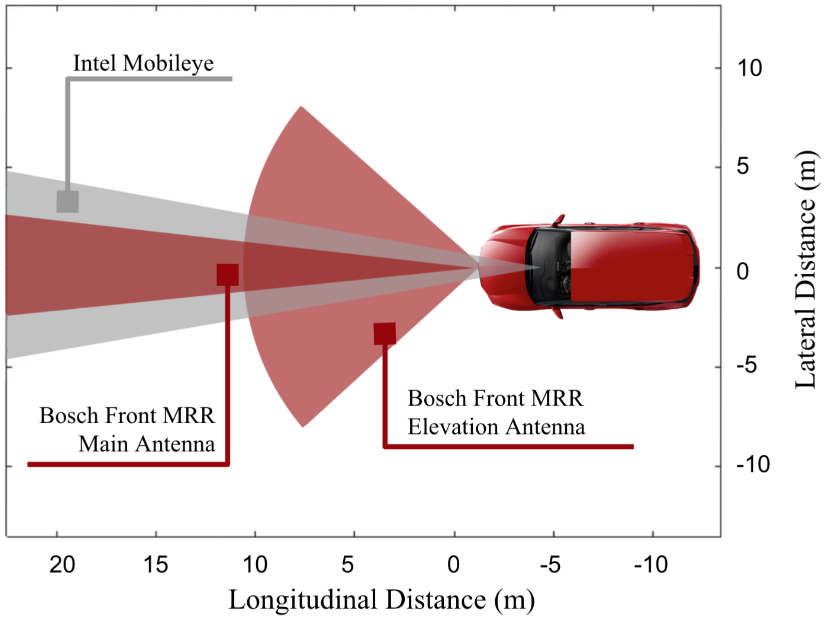

Figure 5. FOV of the Front Perception System.

The perception and communication hardware interface with the CAV system controller, Intel's IEI Tank 870-Q170. The Tank is a high-performance embedded computer running a Linux operating system. As the CAV controller, the Tank performs data logging and parsing and executes the sensor fusion and tracking algorithm.

3. Sensor Fusion and Tracking Algorithm

3.1 Algorithm Architecture

The longitudinal dynamics control system takes inputs from the sensor fusion algorithm as shown in Figure 6. The sensor fusion algorithm aggregates raw detections from multiple sensors to attain a more accurate mapping of the environment than if the sensors were used independently. Figure 7 shows a general sensor fusion process. The vehicles in the driving scenario are detected by the sensors and the raw sensor detections are used as inputs to the sensor fusion algorithm.

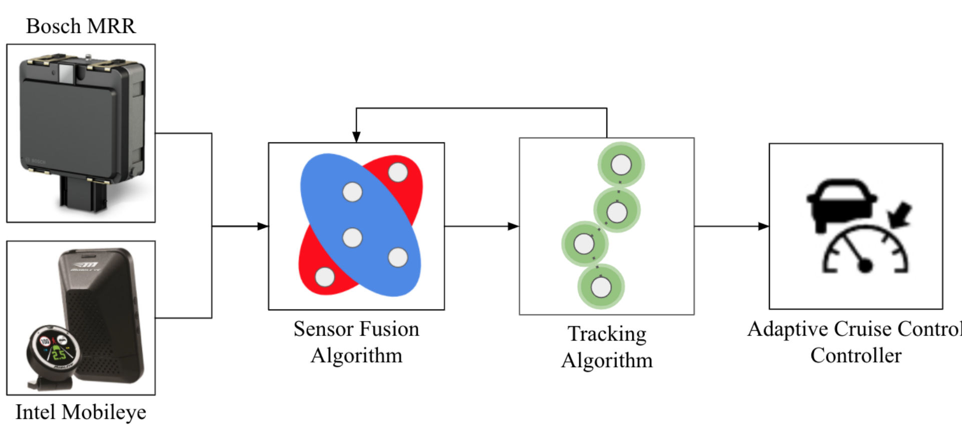

Figure 6. Top-Level Architecture of the UA CAV System.

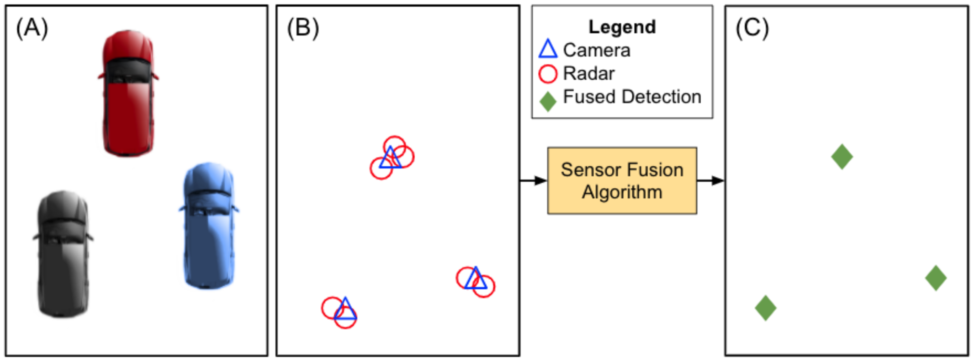

Figure 7. General Sensor Fusion Process: (A) Driving Scenario, (B) Raw Sensor Detections, and (C) Fused Output of Sensor Fusion Algorithm.

The sensor fusion algorithm outputs fused detections which provide accurate position estimations for the surrounding objects. Most sensor fusion algorithms designate primary and secondary sensors, where the detections from primary sensors have precedence over secondary sensor detections [15]. The Mobileye camera was selected as the primary sensor due to its accurate pre-processed detections and object classification capabilities. Bosch's front radar was designated as the secondary sensor as it is highly sensitive and detects insignificant objects. The fused detections are used as inputs to the tracking algorithm, which assigns the fused detections to object tracks. An object track maintains a history of the detected object and is used to predict the object's trajectory. These tracks are used as inputs to the ACC controller and provide feedback to the sensor fusion algorithm.

3.2 Development of Algorithm

The developed sensor fusion algorithm is based on a hierarchical agglomerative clustering algorithm with the Euclidean distance as the distance metric [16]. The algorithm begins by specifying the number of clusters that will be identified. Since the Mobileye was designated as the primary sensor, the number of clusters is set to equal the number of detections reported by the Mobileye. Figure 8 illustrates an example case where three vehicles are detected by Mobileye, indicated by the three triangles, thus the algorithm identified three clusters.

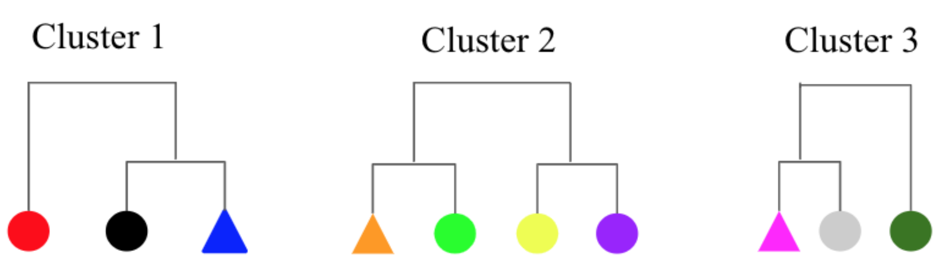

Figure 8. Example Vehicle Detection Scenario for Fusion Algorithm.

Each detection begins as an individual "cluster" and the distance metric is used to find the closest pairs of clusters. Each of the two closest clusters are merged into one cluster. For example, the black radar detection and blue camera detection are clustered together in Figure 8. This process is repeated until the remaining number of clusters equals the specified number of target clusters, which is three in this example. Figure 8 and Figure 9 show the hierarchical clustering results for the example scenario.

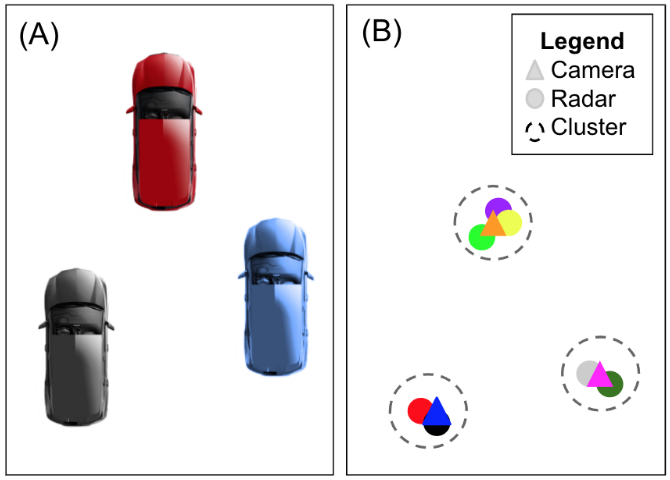

Figure 9. Clustering Result for (A) Driving Scenario and (B) Raw Sensor Detections.

The tracking algorithm uses a linear regression approach, known as the least-squares fit, to assign the fused objects to object tracks and predict the objects' trajectories. The tracking algorithm also determines when to initiate new tracks and discard unused tracks. The algorithm's prediction capabilities are extremely useful when the sensor fusion algorithm fails to detect a tracked object. The sensor fusion and tracking algorithms were developed using MathWorks' MATLAB and Simulink programming environments. Simulink is a graphical programming environment that integrates with the MATLAB language to model and simulate controllers as well as physical systems. MATLAB and Simulink are two tools widely used by students in academic classes as they allow for quicker development. It was also advantageous to develop the sensor fusion and tracking algorithms in Simulink since the ACC controller was developed in Simulink and uses the object tracks as inputs to the controller.

3.3 Testing of Algorithm

Functionality of the sensor fusion and tracking algorithm was verified and validated using three testing methods: MIL, HIL, and VIL. The iterative testing sequence is described in Figure 10. MIL offers the benefits of rapid prototyping and automated testing of the sensor fusion and tracking algorithm. HIL testing is essential for validating the interface and functionality of the hardware that is used by the algorithm. Lastly, VIL enables the verification and validation of the algorithm in real-time and in a complete system.

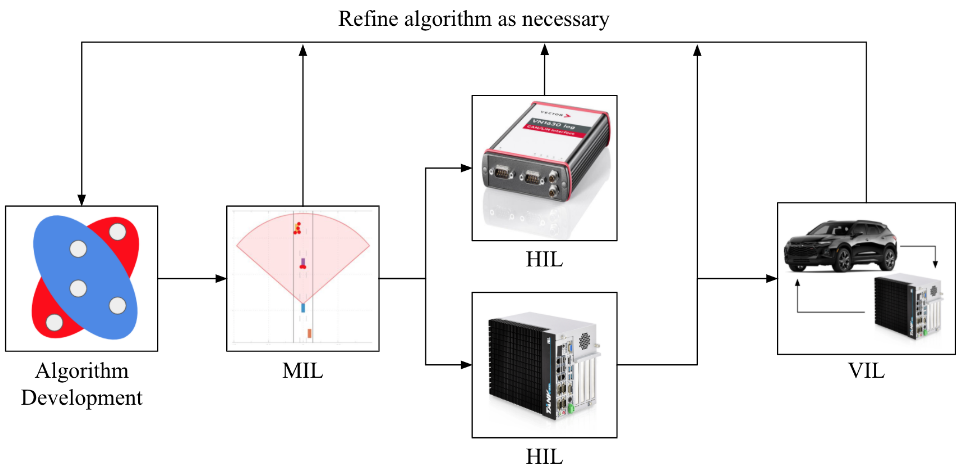

Figure 10. Testing Sequence of the Sensor Fusion Algorithm.

In each of the three environments, the sensor fusion and tracking algorithm was evaluated using the metrics defined in Table 3. Each metric can be used to evaluate the performance of the fusion algorithm independently or the combined sensor fusion and tracking algorithm. When evaluating the fusion algorithm independently, the hit-rate (HR) and miss-rate (MR) evaluate the rate at which the algorithm successfully detects or fails to detect an existing target that is within the primary sensor's detection range. The root mean squared error (RMSE) is used to measure the average error in the algorithm's range estimation for each target. Likewise, for the tracking algorithm, the hit-rate and miss-rate also evaluate the rate at which the algorithm correctly and incorrectly assigns the fused objects to tracks. The RMSE measures the average error in the algorithm's trajectory predictions for the tracked objects.

Table 3. Performance Metrics for Evaluating the Sensor Fusion Algorithm.

Metric	Formula	Description
Hit-Rate	$HR = \frac{TP}{TP + FN}$	Evaluates how often the algorithm detects targets that exist
Miss-Rate	$MR = \frac{FN}{TP + FN}$	Evaluates how often the algorithm fails to detect targets that exist
Root Mean Squared Error	$RMSE=\sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$	Measures the average error in the algorithm's predictions compared to the actual outcome
TP = True Positive, FN = False Negative, N = Total number of predictions		

3.3.1 Model-In-the-Loop (MIL) testing

The MIL testing environment enables rapid prototype development of the sensor fusion and tracking algorithms and can be performed on any computer system. Furthermore, it allows for automated tests to be performed faster than real time. The MIL environment uses simulated driving scenarios and synthetic sensor detections as inputs to the sensor fusion and tracking algorithms. The driving scenarios were developed and executed within MathWorks' Driving Scenario Designer application. The simulated sensors were programmed to include noisy and false detections to imitate the performance of the real-life sensors. Table 4 shows the hit-rate and miss-rate for various detection probability parameters. Since the actual sensors perform with high detection probabilities, the simulated sensors were programmed with detection probabilities of 0.9.

Table 4. Hit-Rate and Miss-Rate for Various Detection Probabilities.

Detection Probability	Hit-rate (%)	Miss-rate (%)
0.9	85	15
0.7	76	24
0.6	66	34

The Driving Scenario Designer application was used to simulate five driving scenarios. The scenario names and descriptions are defined in Table 5. Scenarios A – D include the ego vehicle and a single target vehicle while Scenario E includes the ego vehicle and two target vehicles.

Table 5. Testing Scenarios for Sensor Fusion and Tracking.

ID	Scenario Name	Description
A	Direct Approach	Ego vehicle approaches a stationary target vehicle from 200 meters away at a maximum speed of 25 mph.
B	Right Lane Change	Ego vehicle approaches a stationary target vehicle from 160 meters away, switches into the right lane at 65 meters, and passes the target vehicle at a maximum speed of 25 mph.
C	Left Lane Change	Ego vehicle approaches a stationary target vehicle from 160 meters away, switches into the left lane at 65 meters, and passes the target vehicle at a maximum speed of 25 mph.
D	Back Away	Ego vehicle begins 2 meters behind the stationary target vehicle and reverses at 5 mph until the ego vehicle is 80 meters away.
E	Direct Following	Ego vehicle follows at least 10 meters behind two non-stationary vehicles that are traveling in adjacent lanes at a maximum speed of 25 mph.

Figure 11 shows the bird's-eye plot and the corresponding three-dimensional simulation view for driving scenario A, direct approach. The blue ego vehicle approaches the stationary target, represented in orange, from 200 meters away and at a constant speed of 25 mph.

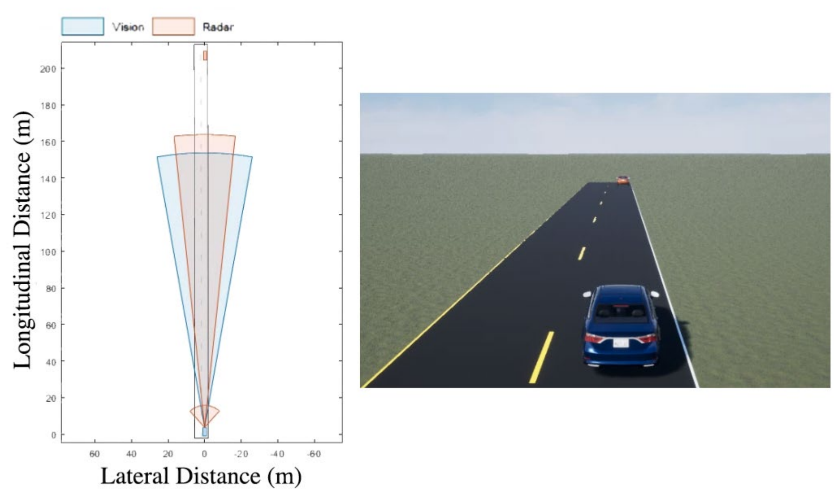

Figure 11. Testing Scenario A (Direct Approach).

Figure 12 shows the bird's-eye plot and the corresponding three-dimensional simulation view for driving scenario E, direct following. Scenario E includes two non-stationary targets displayed in orange and yellow. The blue ego vehicle follows at least 10 meters behind both target vehicles while traveling at a constant speed of 25 mph.

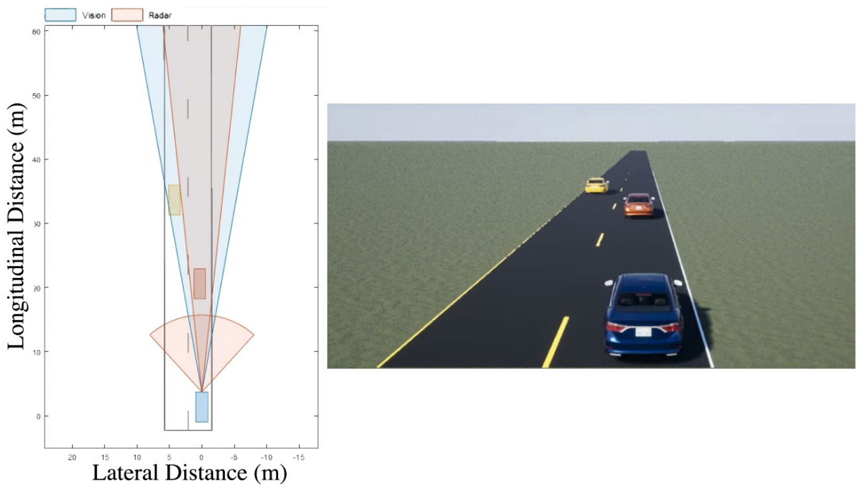

Figure 12. Testing Scenario E (Direct Following).

Figure 13 compares the actual longitudinal distance between the stationary target and the ego vehicle to the estimated distance by the sensor fusion algorithm. The sensor fusion algorithm does not detect the target vehicle until the vehicles are approximately 150 meters apart. This is a result of the algorithm designed to use the vision system as the primary sensor, which has a detection range limited to 150 meters. When the target is within the detection range of both the camera and radar, the algorithm has a miss rate of 8.33%. This behavior is due to either one or both of the sensors failing to accurately detect the vehicle. Although the sensor fusion algorithm fails to detect the vehicle, the tracking algorithm is able to accurately estimate the vehicle's position as shown in Figure 14.

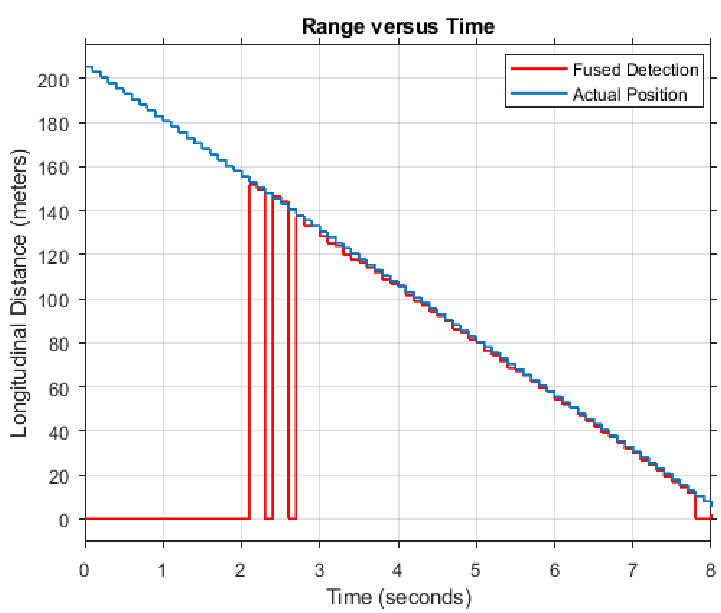

Figure 13. Actual Distance and Estimated Distance by the Sensor Fusion Algorithm without Tracking for Scenario A.

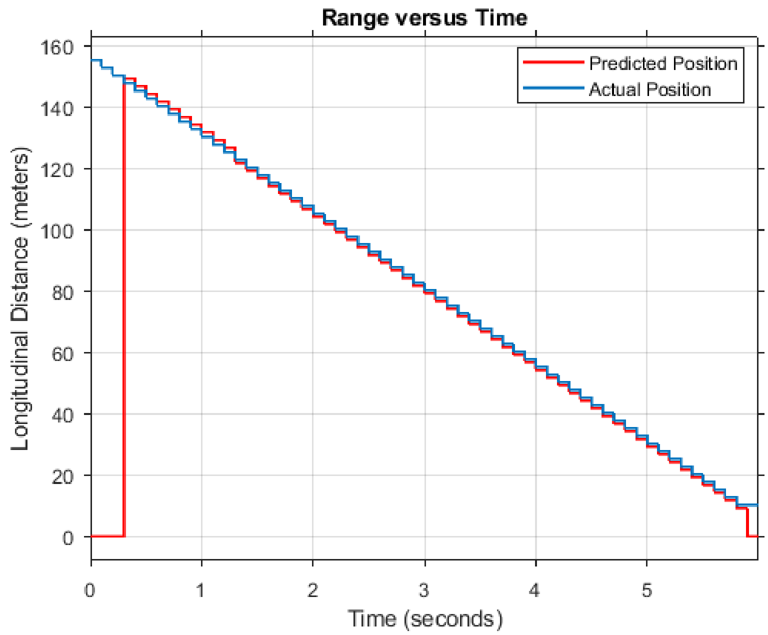

Figure 14. Actual Distance and Estimated Distance by the Sensor Fusion Algorithm with Tracking for Scenario A.

The fusion algorithm and the sensor fusion and tracking algorithm were also evaluated on scenario E, direct following. Figure 15 shows the sensor fusion results for the two targets and their actual longitudinal distances from the ego vehicle. The algorithm had an average hit-rate of 88.89%, average miss-rate of 11.11%, and average range RMSE of 10.08 meters. For the same scenario, a perfect hit-rate and range RMSE of one meter were achieved by combining the fusion and tracking algorithms as shown in Figure 16.

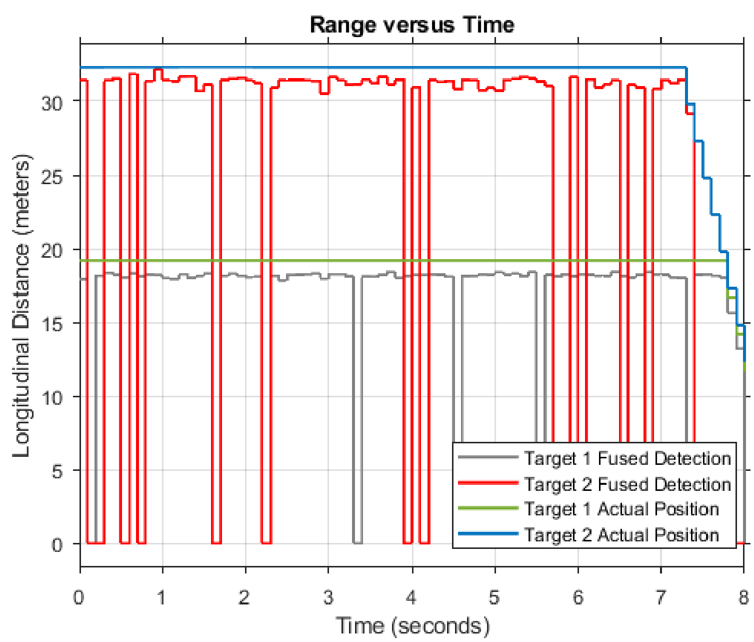

Figure 15. Actual Distance and Estimated Distance by the Sensor Fusion Algorithm without Tracking for Scenario E.

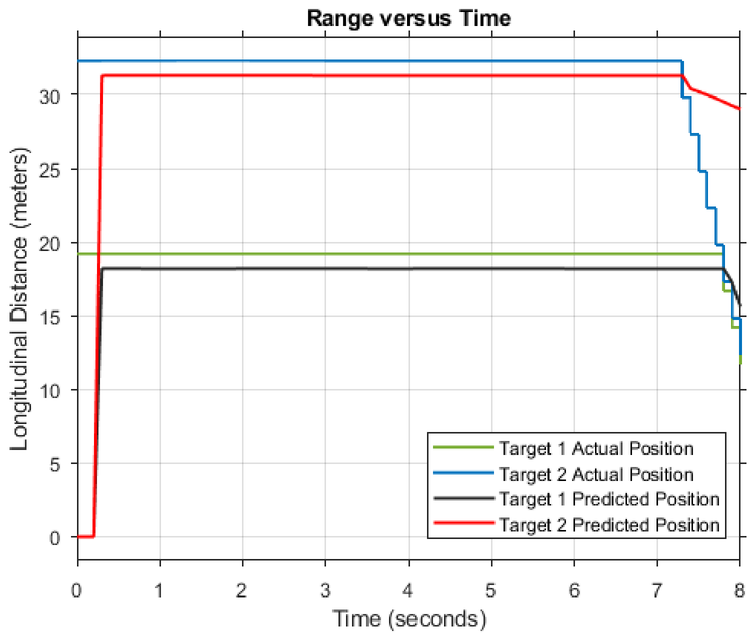

Figure 16. Actual Distance and Estimated Distance by the Sensor Fusion Algorithm with Tracking for Scenario E.

Table 6 summarizes the performance metrics for the sensor fusion algorithm on the five driving scenarios. The sensor fusion algorithm showed the lowest hit-rate for scenario B and scenario C, which are both lane change scenarios. As the ego vehicle is switching lanes, the target vehicle temporarily falls outside of the sensors' FOV. Since the tracking algorithm maintains a history of the target's position, the algorithm is able to predict the target's position even if the target goes undetected. Table 7 shows that for all five driving scenarios, the sensor fusion and tracking algorithm achieved a perfect hit-rate and a range RMSE of one meter.

Table 6. Performance Results for Different Driving Scenarios by the Sensor Fusion Algorithm.

ID	Hit-Rate (%)		Miss-Rate (%)	Range RMSE (m)
A	91.67		8.33	40.79
B	82.76		17.24	35.67
C	78.05		21.95	52.33
D	91.23		8.77	13.43
E	Target 1	93.82	6.18	5.87
	Target 2	83.95	16.05	14.29

Table 7. Performance Results for Different Driving Scenarios by the Sensor Fusion and Tracking Algorithm.

ID	Hit-Rate (%)		Miss-Rate (%)	Range RMSE (m)
A	100		0	1.00
B	100		0	1.00
C	100		0	1.00
D	100		0	1.00
E	Target 1	100	0	1.00
	Target 2	100	0	1.00

MIL testing was used to perform rapid prototyping and iterative testing of the sensor fusion and tracking algorithms. The addition of the tracking algorithm provided a performance improvement of 13.1% to the average hit-rate and an improvement of 96.3% for the average range RMSE. Additionally, multi-target scenarios exposed the limitations of the original track assignment strategy. Thus, a shorter distance threshold was added to the assignment strategy in order to differentiate targets that are closely adjacent to each other. In the future, the detection probabilities of the simulated sensors will be decreased to further assess the prediction capabilities of the tracking algorithm.

3.3.2 Hardware-In-the-Loop (HIL) testing

The HIL framework, described in Figure 17, was used to validate two elements:

1. Sensors' input/output functionalities
2. Algorithm performance on previously recorded data

The first validation goal includes confirming sensor functionalities and receiving accurate raw detections.

These are prerequisites to achieving the second validation goal of evaluating algorithm performance. During a series of drives, the real-world sensor detections were recorded using two individual hardware units: Vector's VN1630 and the CAV controller.

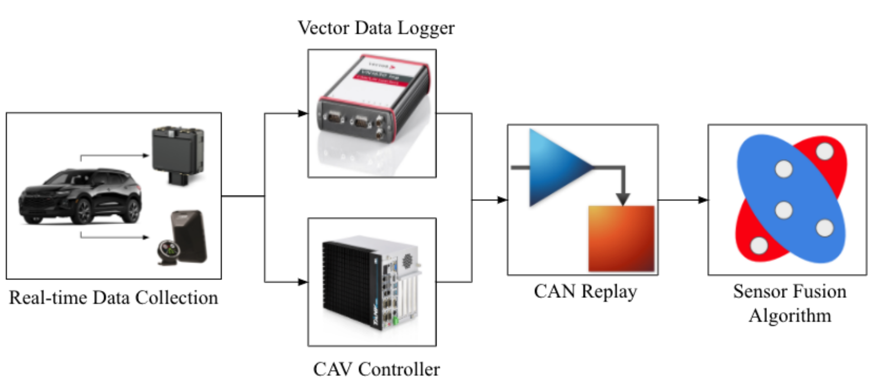

Figure 17. HIL Testing Configuration for the Sensor Fusion.

Initially, the HIL setup included the VN1630 to evaluate the sensors' functionalities and record detections as the device provided ease of data collection in real-time. The VN1630 was then replaced with the CAV controller, which also performs as the dedicated CAV data logging hardware. The recorded sensor detections were replayed in real-time over the Control Area Network (CAN) bus communication interface to the CAV controller. This process was achieved by using MathWorks' Vehicle Network Toolbox (VNT), specifically the "CAN Replay" block. Figure 18 displays the bird's-eye plot for the replayed real-world detections and a raw image frame from the closed course testing of Scenario A, direct approach. The bird's-eye plot shows the Mobileye FOV and detections in blue and the front MRR FOV and detections in red.

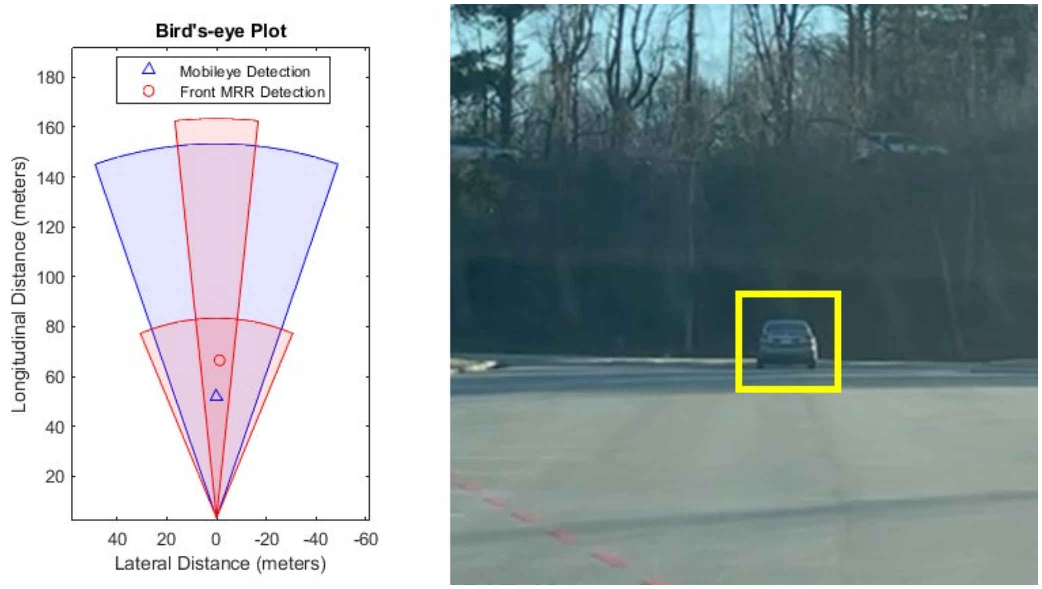

Figure 18. Birds-Eye Plot of Replayed Scenario A, Direct Approach.

The HIL setup revealed that the Mobileye and radar detections had a significant offset when detecting the same target. The fusion algorithm's performance suffered greatly due to this offset. The sensor fusion algorithm was temporarily modified to account for this offset during the clustering process, and the sensors will be recalibrated in future to eliminate the offset. A real-time visualization feature was added to the data collection process so that these calibration issues can be addressed earlier in future.

3.3.3 Vehicle-In-the-Loop (VIL) testing

By transitioning to VIL testing, the sensor fusion and tracking algorithm can be verified and validated in real-time and in the complete system. The algorithm was executed on the CAV controller which was mounted in the Blazer. The algorithm was evaluated in a real-world closed-course testing of scenario A, direct approach. The target vehicle's actual position and relative acceleration were measured using the OxTS RT3000 v2. This Global Navigation Satellite System (GNSS) receiver provides high-precision position and relative acceleration measurements that can be used to validate the sensor outputs. Due to the limitations of the closed-course that was used for testing, the Blazer approached the target vehicle from 130 meters and reached a maximum speed of 20 mph. Figure 19 compares the algorithm's estimated range and actual range of the target vehicle. Additionally, Figure 20 compares the sensor fusion algorithm's estimated range rate, or relative velocity, and the actual range rate.

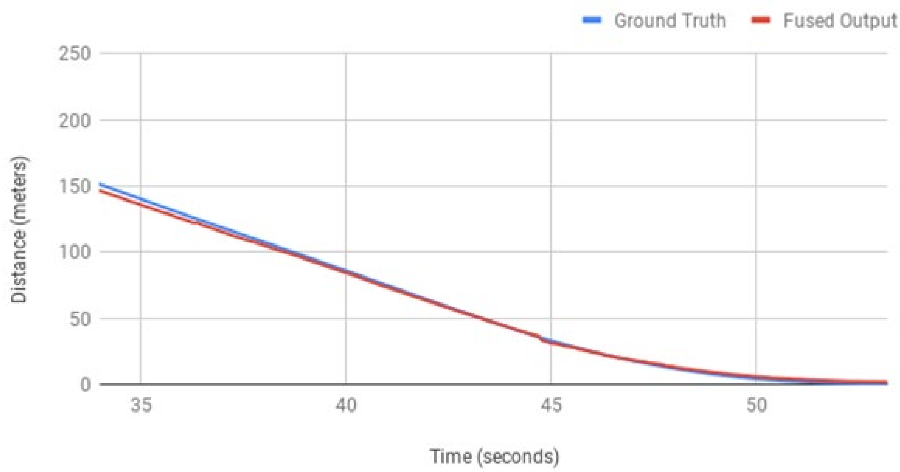

Figure 19. Estimated and Actual Longitudinal Range for Direct Approach.

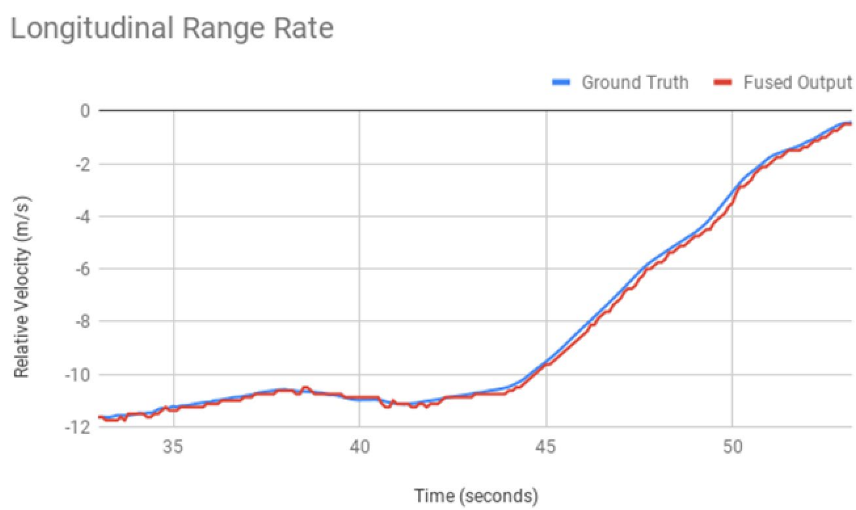

Figure 20. Estimated and Actual Range Rate for Direct Approach.

The algorithm's overall performance is summarized in Table 8. The algorithm showed a 100% hit-rate and 0% miss-rate but overestimated the target vehicle's position with an RMSE of 1.15 meters. The estimated range-rate also had an RMSE of 0.19 m/s^2.

Table 8. Evaluation Metrics for Direct Approach.

ID	Hit-Rate	Miss-Rate	Range RMSE (m)	Range Rate RMSE (m/s^2)
A	100	0	1.15	0.19

The results from the VIL testing revealed that the algorithm's range and range rate estimation require significant improvements. This is a result of the prediction models being trained on a small dataset of real-world data and the radar requiring further calibration.

4. Discussion

4.1 Comparison of Simulation and Real-world Testing

Simulation testing environments, such as MIL, provide the advantage of rapid and iterative testing in ideal and safe environments. For example, both the sensor fusion and sensor fusion and tracking algorithms were evaluated on identical driving scenarios. This revealed that the sensor fusion and tracking algorithm has an improvement of 13.1% for the average hit-rate and an improvement of 96.3% for the average range RMSE. When performing real-world testing, each driving scenario may be slightly different and thus the algorithm's performance differs. The simulated scenarios, however, can be programmed to perform similarly to the realistic, non-ideal driving scenarios. For example, the sensors were programmed to have a detection probability of 0.9 and thus had an average miss rate of 15%.

Although the simulated scenarios can be modified to perform closely to the real-world scenarios, it is not possible that the simulations perfectly imitate the real-world scenarios. Real-world testing, such as VIL testing, evaluates the algorithm's performance in a complete system while executing in real-time. VIL testing revealed that the algorithm's performance is highly affected by the sensors' sensitivity to different lighting and weather conditions. When testing in the rain, the Mobileye camera does not have a perfect hit-rate as the sensor is often hindered by rain and the windshield wipers. These scenarios justified the importance of accurate predictions from the tracking algorithm. Since the range and range rate estimations resulted in RMSEs of 1.15 meters and 0.19 m/s^2 respectively, the tracking algorithm requires further improvements.

4.2 Future work

The MIL, HIL, and VIL environments revealed that the algorithm's performance can be improved by further calibrating the front radar. Since the tracking algorithm is based on a linear regression model, the algorithm does not perform well in complex, non-linear scenarios. In order to overcome this limitation, the team will develop a tracking algorithm that combines Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNN) with deep reinforcement learning (DRL). The RNNs are responsible for predicting the target vehicle's trajectory while the DRL performs the track association process. Lastly, a Software-In-the-Loop (SIL) environment will be added to the iterative testing procedure. SIL testing will compile generated source code and verify that the output matches that of the model subsystem.

4.3 Educational Outcomes

One of the goals of the EcoCAR Mobility Challenge was to supply unique industry-led training opportunities to all EcoCAR students to provide the technical skills needed for students to complete competition objectives and deliverables, especially those related to CAV systems [4]. These industry-led training topics include Hardware-In-The-Loop, hardware debugging, technical requirement development, vehicle testing and test plans, and data collection. In addition, the graduate students leading the team have built technical onboarding modules and hosted bi-weekly internal lecture training sessions to provide knowledge for the operation of cross-functional student teams. Topics include automotive standards, modeling and simulation of automotive systems, operation of hybrid- and battery- electric vehicles, model-based design, software verification and validation, agile process methodology, and overviews of various software tools.

The development of the sensor fusion and tracking algorithm primarily occurred during the COVID-19 pandemic, so all student development and training took place virtually. Training was conducted and recorded over online tools that allowed any interested student on the team to participate. Recordings of these training sessions will allow for their re-use in later years of the competition for new students who join the team.

An entire CAV sub-team of the project assisted in the development of the sensor fusion and tracking algorithms and their realization on the student-adapted prototype vehicle. Students gained understanding of complex topics such as automotive system modeling, algorithm development, and testing in simulation environments. The participating students also had numerous chances to interact with various industry sponsors through both training and networking opportunities organized by ANL.

These extensive hands-on activities have also been incorporated in course material at the university. For example, several relevant courses have been developed covering topics including modeling and simulation of hybrid electric vehicles and introductory and advanced machine learning. In these courses, students use many of the same tools and concepts that they use as a part of their work on the competition team.

Feedback for internal team training was generally positive, with students specifically citing an increased comfort level learning the related topics, with the virtual platform making it easier to attend in a busy student's schedule. Anonymous feedback was also gathered through students by ANL which showed that 10% of the students rated the industry training "Not at all" or Slightly" valuable, 24% rated the training "somewhat" valuable, and 66% rated the trainings "moderately" or "extremely" valuable [4].

5. Conclusion

This paper detailed how the UA EcoCAR Mobility Challenge team is using model-based design methodology to develop and verify its sensor fusion and tracking algorithms. The developed sensor fusion and tracking algorithm was evaluated using a MIL, HIL, and VIL testing environments. The MIL environment used simulated driving scenarios and generated synthetic sensor detections to rapidly and iteratively test the algorithm. The sensor fusion and tracking algorithm demonstrated an average improvement of 13.1% in the hit-rate and an average improvement of 96.3% in the range RMSE over the pure sensor fusion algorithm. The HIL environment was used to replay recorded sensor detections in real-time over the CAN bus communication interface to the CAV controller. This environment revealed that the algorithm's performance suffered from a sensor calibration issue. This inspired the team to integrate a real-time visualization feature to the data-collection process such that these issues can be addressed sooner. Lastly, the algorithm was verified and validated in real-time and with a complete vehicle system by using a VIL environment. VIL testing revealed that the algorithm achieves a perfect hit-rate, but the range and range rate prediction models require additional training.

This project was successfully completed by the student team, and the UA team earned an overall first-place finish for the EMC Year 3 competition as well as various awards from industry sponsors. The feedback from students was very positive in regard to the ability to do the specific hands-on activities and collaborative work that would not be available to them otherwise. Students gained skills in automotive engineering, modeling and simulation, requirement development and testing, sensor fusion and tracking, ACC, and software engineering to complete the development and testing of these vehicle features.

References

1. General Motors. 2021. "General Motors forges ahead toward a future of Zero Crashes, Zero Emissions and Zero Congestion at CES 2021." Accessed January 31, 2022. https://media.gm.com/media/me/en/gm/news.detail.html/content/Pages/news/me/en/2021/gm/01-19-General-Motors-forges-ahead-toward-a-future-of-Zero-Crashes-Zero-Emissions-and-Zero-Congestion-at-CES-2021.html.
2. Gindrat, R. 2018. "How MaaS Public Transit Is Changing the World." Accessed January 31, 2022. https://www.forbes.com/sites/forbestechcouncil/2018/11/06/how-maas-public-transit-is-changing-the-world/?sh=509683497518.

3. MarketsandMarkets. 2021. “Mobility as a Service Market.” Accessed February 1, 2022. https://www.marketsandmarkets.com/Market-Reports/mobility-as-a-service-market-78519888.html.
4. Britt J, Shoults LW. How COVID-19 led to improvements and adaptations to experiential learning opportunities for an increasingly remote environment. In2021 ASEE Virtual Annual Conference Content Access 2021 Jul 26.
5. SAE International. 2018. "SAE International Releases Updated Visual Chart For Its ‘Levels Of Driving Automation’ Standard For Self-Driving Vehicles.” Accessed February 1, 2022. https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles.
6. Marsden G, McDonald M, Brackstone M. Towards an understanding of adaptive cruise control. Transportation Research Part C: Emerging Technologies. 2001 Feb 1;9(1):33-51.
7. Xiao L, Gao F. A comprehensive review of the development of adaptive cruise control systems. Vehicle system dynamics. 2010 Oct 1;48(10):1167-92.
8. Yeong DJ, Velasco-Hernandez G, Barry J, Walsh J. Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors. 2021 Mar 18;21(6):2140.
9. Kim J, Han DS, Senouci B. Radar and vision sensor fusion for object detection in autonomous vehicle surroundings. In2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN) 2018 Jul 3 (pp. 76-78). IEEE.
10. Aarenstrup, R. 2015. *Managing Model-Based Design*. Natick: The MathWorks.
11. Mahapatra S, Egel T, Hassan R, Shenoy R, Carone M. Model-based design for hybrid electric vehicle systems. SAE International; 2008 Apr 14.
12. Plummer AR. Model-in-the-loop testing. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2006 May 1;220(3):183-99.
13. Pietruch M, Mlyniec A, Wetula A. An overview and review of testing methods for the verification and validation of ADAS, active safety systems, and autonomous driving. Mining–Informatics, Automation and Electrical Engineering. 2020;58.
14. Olson J, Stevens B, Phan A, Yoon HS, Puzinauskas P. Model-based design approach for validation of vehicle longitudinal control algorithm. Journal of student research. 2022 Jun 12;11(2).
15. Park YS, Choi WS, Han SI, Lee JM. Navigation system of UUV using multi-sensor fusion-based EKF. Journal of institute of control, robotics and systems. 2016;22(7):562-9.
16. Yeong DJ, Velasco-Hernandez G, Barry J, Walsh J. Sensor and sensor fusion technology in autonomous vehicles: A review. Sensors. 2021 Mar 18;21(6):2140.