Simulation of Advanced Driving Assistance Systems for a Dynamic Vehicle Model

Tevfik Ataman
Vocational School, Cankiri Karatekin University, Turkiye
tevfikataman@karatekin.edu.tr (corresponding author)

Mehmet Ali Biberci
Faculty of Engineering, Cankiri Karatekin University, Turkiye
m.alibiberci@karatekin.edu.tr

Mustafa Bahattin Celik
Faculty of Engineering, Karabuk University, Turkiye
mcelik@karabuk.edu.tr

Received: 3 July 2024 | Revised: 20 July 2024 and 25 July 2024 | Accepted: 26 July 2024

ABSTRACT

Advanced Driving Assistance Systems (ADAS), such as collision avoidance systems and adaptive cruise control, are important features of autonomous driving and are gaining importance day by day in terms of increasing road safety. To increase the reliability of the system, virtual simulation environments are used during the design and development stages. This study examines the effect of ADAS features on energy parameters during the driving cycle in a virtual simulation environment. The discussion focuses on the simulation of an electric vehicle and the relationship between energy use and ADAS functions. For ADAS modeling and simulation, a dynamic model was developed in MATLAB and examined throughout the NEDC drive cycle.

Keywords-ADAS; MATLAB; driving scenario; dynamic modeling

I. INTRODUCTION

To increase safety, many leading automakers develop and evaluate Advanced Driving Assistance Systems (ADAS). One of the ADAS modules that is typically available as an add-on in contemporary cars is the collision avoidance system. ADAS modules use a variety of sensors to sense the surroundings of the vehicle. Sophisticated software can predict risky situations based on sensor input and provide control signals to prevent an accident [1, 2]. A highly realistic simulation environment is required to save time and reduce risks throughout the early phases of ADAS design, development, and evaluation. Various simulation tools are used for environments, sensors, and vehicles to reduce development time and expenses and ensure the proper operation of complex software [3]. Software-In-the-Loop simulation (SIL) is a useful technique for testing complicated software systems before putting them to use in real-world scenarios. To provide an efficient platform for software testing, the SIL simulation concept incorporates real production software (or algorithms) into a virtual environment (mathematical simulation) that contains models of a physical system. In this case, a simulation of the car and its interactions with the real environment is required [4, 5].

The Autonomous Emergency Braking (AEB) system aims to prevent an imminent collision if the driver is unaware of the incident or does not respond quickly enough. This system uses a mix of sensors, including vision, radar, ultrasonic, camera, and GPS, to successfully detect obstructions in the vehicle's path, including pedestrians and other cars [6]. The AEB system receives real-time data from these sensors about the vehicle's surroundings. The AEB system uses three main systems, brake, acceleration, and steering control, to keep the vehicle under control [7]. AEBs are designed to alert drivers to approaching collisions, recognize potentially dangerous situations, and help prevent collisions or reduce their impact. Combining many sensors has become necessary to handle large amounts of data due to the growing number and variety of sensors. To gain more accurate information about the surroundings, the system may integrate and merge data from several sensors, including cameras, LIDARs, and RADARs, thanks to sensor fusion technology [8]. This technique offers many benefits, including increased image resolution, increased dependability, and expanded parameter coverage. The system can operate more effectively when multiple inputs from many sensors are used, as opposed to only one sensor. Since AEB systems rely on a variety of sensors, integrating sensor fusion technology improves vehicle safety [1].

Passenger vehicles with Adaptive Cruise Control (ACC) can adjust their speed to match the traffic flow [9]. This system focuses on driving in congested traffic while maintaining a safe distance from the vehicle in front. By keeping a preset minimum distance from the vehicle ahead and regulating the ego vehicle's speed automatically, ACC can help drivers feel less stressed. As a result, the driver is more comfortable and can focus on the road slightly better. ACC is often based on a radar sensor [10]. This equipment is mounted in front of the car and continuously scans the road ahead. ACC maintains the driver's selected speed as long as the road ahead is clear. If the system detects a slower vehicle within its range, it softly reduces speed by depressing the accelerator or aggressively activating the brake control system. ACC automatically increases speed to the driver's selected level if the car in front accelerates or changes lanes. When traveling at speeds greater than 30 km/h, standard ACC can be activated to assist the driver, especially when traveling cross-country or on highways. When traveling at less than 30 km/h, the ACC stop-and-go version is in operation. Even at very low speeds, it can keep the predetermined distance from the leading vehicle or even stop the ego vehicle.

II. SYSTEM DESCRIPTION

Since real testing and experimentation are costly, simulation programs are widely used. Designing and analyzing systems in this way is called model-based system design. This study analyzes a system using a mathematical model with the help of Matlab/Simulink, focusing on measuring the charge rate of a battery at certain voltage values and its relationship with ADAS. Figure 1 shows the simulation framework.

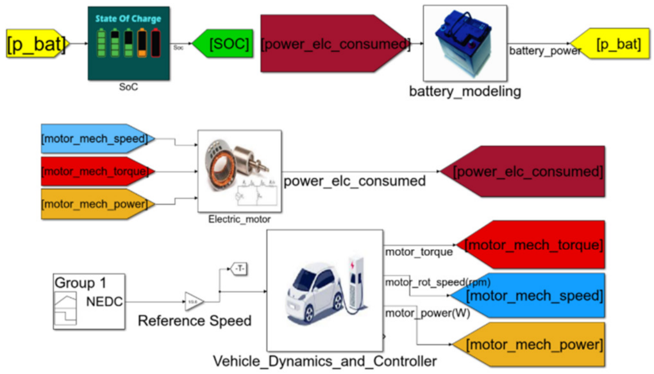

Fig. 1. General system model.

A. Overall System Modeling

MATLAB Simulink 2020a was used for the entire modeling process. Utilizing the sensor fusion algorithm, AEB was implemented using an Automated System Driving Toolbox. Using Driving Scenario Designer software, test scenarios were developed after the system was designed. Test scenarios recommended by EURO NCAP were developed. Different fusion algorithms were developed based on the level of fusion involved to accomplish sensor fusion. This study used a multi-object tracking framework, motion models, assignment algorithms, and Kalman filtering to realize sensor fusion.

In the modeling phase, the driving cycle model was first created. Subsequently, the vehicle dynamics model and the transmission model were created. A controller was added to follow the vehicle cycle. In the next stage, vehicle and environment parameters, such as the torque, speed, and power outputs that the wheels need to achieve the desired speed, were entered into the vehicle dynamics model. After creating the transmission model and entering the parameters, the torque, speed, and power outputs that the engine should provide were calculated. Then, an electric motor model was created, printouts were taken, electric motor parameters were added, and the electric power that should be provided was determined. By creating a battery model, parameters and values from the previous model were defined as input. At this stage, the electric power that needs to be drawn from the battery was determined. The energy analysis aimed to determine the State Of Charge (SOC) modeling.

B. Driving Cycle

In the EU, two types of driving cycles are typically used to evaluate vehicles. The first is the 1999 ECE-15, also known as the urban driving cycle. This was designed to simulate urban driving conditions in Rome or Paris. It replicates a 4052 m urban ride with a maximum speed of 50 km/h and an average speed of 18.7 km/h, as shown in Figure 2. This is especially helpful for evaluating an EV's performance when driving in cities [11].

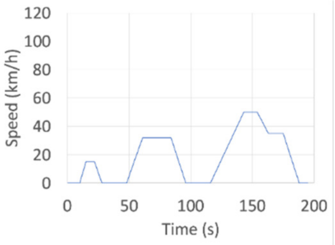

Fig. 2. ECE-15 driving cycle.

C. Simulation Software

The simulation was performed in Matlab, including the vehicle dynamics, SOC, battery, electric motor, and ADAS models. Improvements can be made by adding new components to the model. The equations used for modeling were analytical or semi-empirical, as described below. The theoretical results were validated against bulk data, and the ECE-15 driving cycle produced by this model matches the corresponding bulk data very well. The presented model can be used for any ADAS model by changing the parameters detailed below [11].

III. ADAS MODEL AND BASIC EQUATIONS

A. Development of Vehicle Model

Lead vehicle and ego vehicle dynamics were modeled in Simulink. Figure 3 shows the leading vehicle dynamics model used for ACC modeling. This modeling is important for distance and speed control.

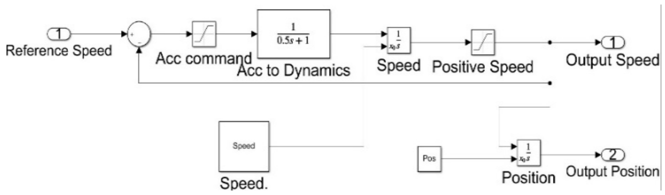

Fig. 3. Leading vehicle dynamic model.

To approximate a realistic driving environment, the acceleration of the front vehicle varies according to a sine wave during the simulation. The ACC system block provides an acceleration control signal for the car. The sampling time T_s and the simulation time T are defined in s. Figure 3 shows the basic vehicle tracking graph obtained with these sampling and simulation times, showing the positions of vehicles starting from different points with the same speeds.

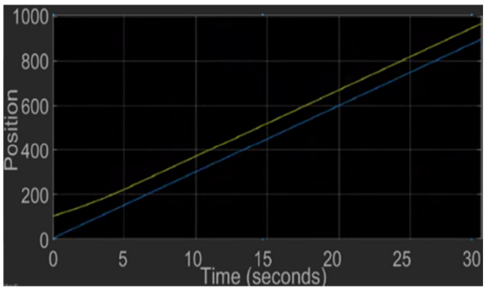

Fig. 4. Vehicle tracking chart with the same speeds from different locations.

B. Configuration of Adaptive Cruise Control (ACC) Block

The ACC system is used in addition to the AEB system. When using this ADAS feature, the ego vehicle slows down as it approaches a vehicle in front and accelerates to cruising speed if the front vehicle moves a safe distance away. To initiate the development of ACC, the initial stage involves collecting data from sensors that are installed on the ego vehicle. Two essential components are required to implement ACC: a camera and a radar sensor. The camera identifies various items within the frame, such as vehicles, pedestrians, and trees. Meanwhile, the radar determines the precise distance between the ego vehicle and each detected object [12]. The output of the ACC system is the desired acceleration of the ego vehicle. The safe distance between the vehicles is a function of their speed. Given the physical limitations of vehicle dynamics, acceleration is limited to the range of -3 to 2 m/s^2. This situation is shown in the block parameters in Table I. Figure 5 shows the addition of the ACC block to the system [13].

TABLE I. PHYSICAL LIMITATIONS OF VEHICLE DYNAMICS

Parameters	Values
Initial condition for longitudinal velocity (m/s)	30
Default spacing (m)	10
Maximum velocity (m/s)	50
Minimum longitudinal acceleration (m/s^2)	-3
Maximum longitudinal acceleration (m/s^2)	2
Sample time (s)	0.1

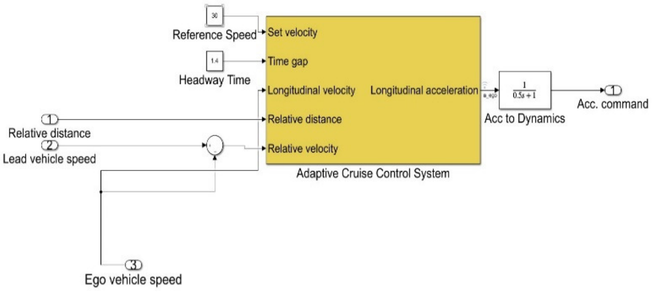

Fig. 5. Integrating the ACC block into the system.

C. Calculating the Stopping Time

The braking controller subsystem utilizes the Forward Collision Warning (FCW) and AEB control algorithm, which is based on a calculation approach that determines the stopping time [13]. T_{stop} is the duration from when the ego vehicle initially engages its brakes, with α_{brake} deceleration, until it stops. The following equation determines the stopping time:

$$T_{stop} = V_{ego}/\alpha_{brake} \qquad (1)$$

where V_{ego} is the velocity of the ego vehicle, and T_{stop} is the duration it takes to stop. The FCW system notifies the driver of an imminent collision with a vehicle ahead. The driver is expected to respond to the alert and use the brake with a T_{react} delay time. The total travel time of the ego vehicle before colliding with the vehicle in front can be expressed as:

$$T_{FCW} = T_{react} + T_{stop} = T_{react} + V_{ego}/\alpha_{brake} \qquad (2)$$

A comparison is made to determine whether the AEB system will be active or not. With data from sensors, the Total Collision Time (TTC), which refers to the time before the ego vehicle hits any obstacle, is evaluated. If the T_{FCW} value is higher than the TTC value, the FCW warning is activated. If the driver does not react under these conditions, the AEB system is activated, and autonomous braking is applied to the ego vehicle. In this way, even if the accident is inevitable, the impact of the collision will be minimized by reducing the speed of the ego vehicle as much as possible. Gradual braking can also be applied at this stage.

IV. RESULTS AND DISCUSSION

The simulation model was analyzed using the above formulations and modeling systems. An ADAS system was simulated in MATLAB, and the results were evaluated. The blocks used to make the above modeling are necessary to analyze the ADAS system. Figure 6 shows the speed graph for the leading and the ego vehicle. The results show that the velocity of the ego vehicle fluctuates to reach the desired speed value. This will have negative consequences in terms of driving comfort, safety, and energy consumption. Therefore, the block parameters were rearranged to reduce this fluctuation with MPC. Table II shows the edited parameters. By adjusting the block parameters, the ego vehicle reached the desired speed more efficiently. The fluctuation was less and therefore, the desired speed was achieved in a shorter time, as shown in Figure 7 [12].

While the vehicle moves based on the ECE-15 cycle, the electric motor speed and the driving cycle are similar. This is because there is no need for a gearbox for using the electric motor. The torque required by the vehicle during acceleration can be met even if the electric motor efficiency is low, due to the high differential ratio. The electric motor operates with 74-90% efficiency during the ECE-15 cycle. Since the ECE-15 cycle has an idle rate of almost 25%, the average efficiency of the electric motor is decreased. Figure 8 shows the acceleration and speed graph of the vehicle while following the cycle. The vehicle acceleration reaches 1.1 m/s^2 in acceleration and -1.4 m/s^2 in deceleration.

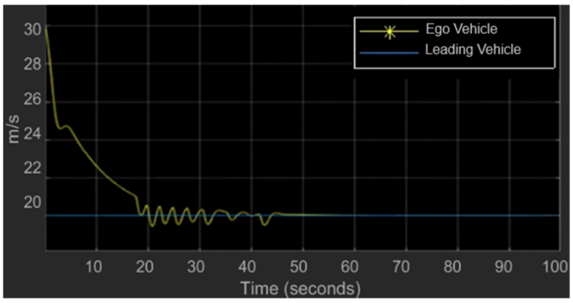

Fig. 6. Speed graph of the leading and the ego vehicles.

TABLE II. REGULATED PARAMETERS

Parameters	Values
Initial condition for longitudinal velocity (m/s)	30
Default spacing (m)	5
Maximum velocity (m/s)	50
Minimum longitudinal acceleration (m/s^2)	-3
Maximum longitudinal acceleration (m/s^2)	2
Sample time (s)	0.5

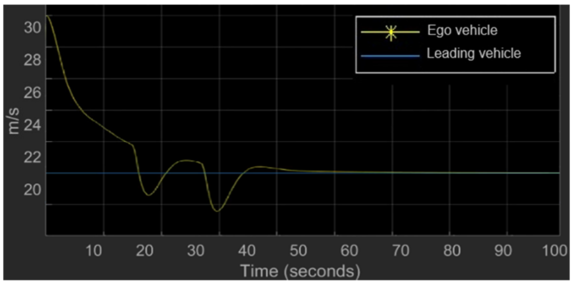

Fig. 7. Speed graph of the leading and ego vehicles after regulated parameters.

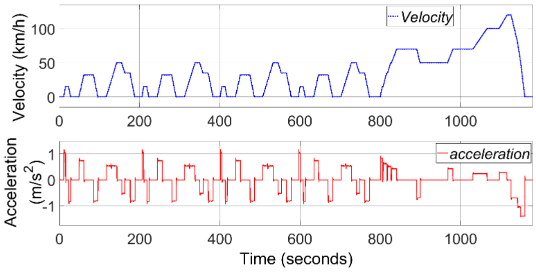

Fig. 8. Vehicle acceleration and velocity graph.

The electric energy produced is directly proportional to the energy consumed. Figure 9 shows the remaining battery charge status and total distance data after performing the ADAS modeling with the driving cycle mentioned above. Since the power consumed by low- and high-voltage accessories is included in the model, energy consumption continues even when the vehicle is stationary [14]. The charge rate of the battery drops to 92% in one cycle, that is, in 1180 s. The battery SOC is decreasing, but not linearly. While it decreases at some points, it increases at other points. On examination of the places it rises, it is observed that it is related to regenerative energy. This situation is examined separately in the graph regarding battery power [15].

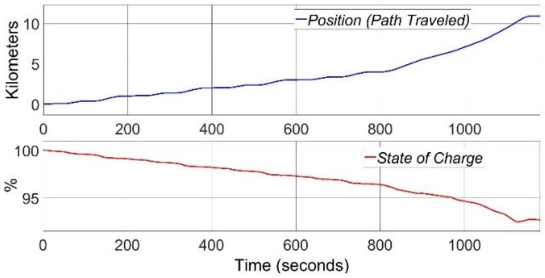

Fig. 9. Battery charge status and total distance after ADAS model.

Figure 10 shows the data obtained by applying ADAS modeling to the general system model, including the driving cycle, vehicle dynamic model, battery model, and controller. During the first 3 s, the ego vehicle accelerates at maximum throttle to achieve the desired speed chosen by the driver [16]. Between 3.2 and 12.9 s, the leading vehicle experiences a gradual increase in acceleration. Consequently, to ensure a safe distance from the vehicle in front, the ego vehicle increases its speed at a reduced pace. Between 13 and 25 s, the ego vehicle maintains the speed selected by the driver, as indicated in the velocity plot. However, when the leading vehicle decreases its speed, the discrepancy in spacing gradually diminishes and eventually reaches zero after 20 s. Between 25-45 s, the leading vehicle decelerates and then accelerates once more. During the simulation, the controller ensures that the real distance between the two cars is always larger than the predetermined safe distance [17].

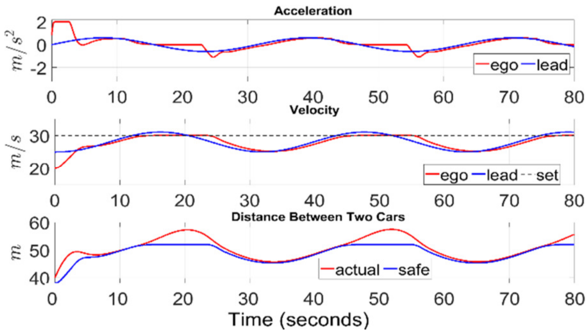

Fig. 10. Data obtained through ADAS modeling.

Figure 11 shows the engine power, battery power, and electric power data that must be given to the engine for the system to which ADAS modeling is applied.

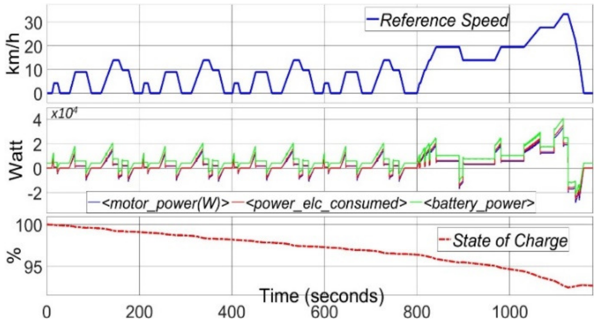

Fig. 11. Speed, power, and charge status data after ADAS modeling.

Examining from the beginning of the cycle, when the speed is zero, the engine and mechanical power are also zero. Then, when the vehicle starts to accelerate, the mechanical-electrical power required for the engine also increases. In this case, the electric power coming out of the battery is more. When the vehicle slows down, that is, during braking, this braking power is seen to be negative. It is possible to store this power in the battery. While the motor's electrical and mechanical power is zero, the battery power is around 0.4 kW. This is because the resistor and accessory forces are included in the modeling. Figure 12 shows the most prominent points of the cycle in detail [18].

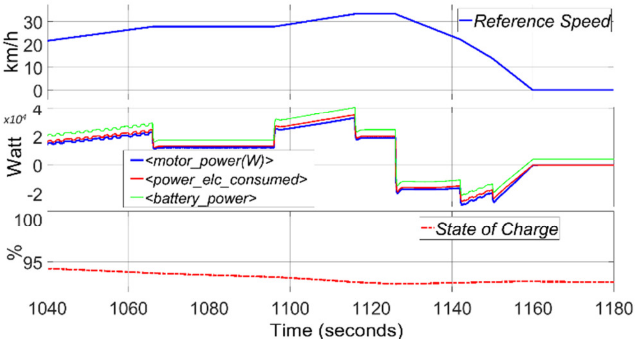

Fig. 12. Salient points of the cycle.

Distinct points of the cycle provide more detailed information about modeling. These points are the points where the vehicle is fastest and slows down with great acceleration [4, 19]. Depending on the ADAS model, a negative power appears on the engine power graph during braking, and this power is recovered by regenerative braking and stored in the battery as seen in the SOC graph. Figure 13 shows the energy that must be drawn from the battery to ensure the driving cycle if the advanced driving assistance systems are active, the energy provided by regenerative braking, and the energy consumed from the battery.

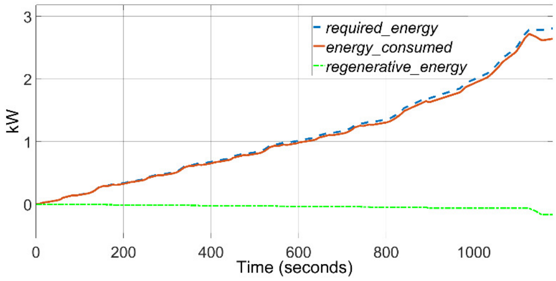

Fig. 13. Required energy, consumed energy, and regenerative energy.

V. CONCLUSION

This study performed ADAS modeling to examine the energy consumption effects of a system that includes dynamic vehicle, battery, electric motor, and SOC models. When the ADAS model is active, reducing the distance between vehicles has a positive effect on the battery charge status. In this, the effect of regenerative braking can be seen. Values below zero in the power curve graphs support this. Toward the end of the driving cycle, as the vehicle speed increases, there is a sharp decrease in the battery charge state, while the intense braking effect and the effects of regenerative power increase the charge state again. While the energy required to complete the driving cycle was 2.8 kW, the energy consumed was found to be 2.64 kW. The difference, 0.16 kW, is provided by regenerative braking. With the effect of the ADAS model, 5.71% of the energy is recovered. Various improvements can be made to the ADAS or vehicle dynamics models. With a more developed dynamic model, the energy consumption of the ADAS model can be evaluated during different driving cycles.

REFERENCES

[1] M. J. Kim, S. H. Yu, T. H. Kim, J. U. Kim, and Y. M. Kim, "On the Development of Autonomous Vehicle Safety Distance by an RSS Model Based on a Variable Focus Function Camera," *Sensors*, vol. 21, no. 20, Jan. 2021, Art. no. 6733, https://doi.org/10.3390/s21206733.

[2] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Vehicle System Dynamics*, vol. 44, no. 7, pp. 569–590, Jul. 2006, https://doi.org/10.1080/00423110600563338.

[3] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a Formal Model of Safe and Scalable Self-driving Cars." arXiv, Oct. 27, 2018, https://doi.org/10.48550/arXiv.1708.06374.

[4] D. Gruyer, S. Glaser, S. Pechberti, R. Gallen, and N. Hautiere, "Distributed Simulation Architecture for the Design of Cooperative ADAS," presented at the First International Symposium on Future Active Safety Technology toward zero-traffic-accident, Tokyo, Japan, Sep. 2011.

[5] V. Ili, S. Popi, and M. Kovači, "Data flow in automated testing of the complex automotive electronic control units," in *2016 Zooming Innovation in Consumer Electronics International Conference (ZINC)*, Novi Sad, Serbia, Jun. 2016, pp. 1–3, https://doi.org/10.1109/ZINC.2016.7513639.

[6] K. Abdelgawad, J. Gausemeier, J. Stöcklein, M. Grafe, J. Berssenbrügge, and R. Dumitrescu, "A Platform with Multiple Head-Mounted Displays for Advanced Training in Modern Driving Schools," *Designs*, vol. 1, no. 2, Dec. 2017, Art. no. 8, https://doi.org/10.3390/designs1020008.

[7] T. Helmer, "Development of a Methodology for the Evaluation of Active Safety using the Example of Preventive Pedestrian Protection," Ph.D. dissertation, Technische Universität Berlin, 2014.

[8] Y. Rachakonda and D. S. Pawar, "Evaluation of intersection conflict warning system at unsignalized intersections: A review," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 10, no. 4, pp. 530–547, Aug. 2023, https://doi.org/10.1016/j.jtte.2023.04.003.

[9] "Beyond the Headlights: ADAS and Autonomous Sensing," Woodside Capital Partners, Sep. 2016.

[10] W. Koch, "Sequential Track Extraction," in *Tracking and Sensor Data Fusion: Methodological Framework and Selected Applications*, W. Koch, Ed. Springer, 2014, pp. 83–88.

[11] M. A. Biberci and M. B. Celik, "Dynamic Modeling and Simulation of a PEM Fuel Cell (PEMFC) during an Automotive Vehicle's Driving Cycle," *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5796–5802, Jun. 2020, https://doi.org/10.48084/etasr.3352.

[12] "Adaptive Cruise Control System Using Model Predictive Control - MATLAB & Simulink," *Mathworks*. https://www.mathworks.com/help/mpc/ug/adaptive-cruise-control-using-model-predictive-controller.html.

[13] W. Zhou, X. Wang, Y. Glaser, X. Wu, and X. Xu, "Developing an improved automatic preventive braking system based on safety-critical car-following events from naturalistic driving study data," *Accident Analysis & Prevention*, vol. 178, Dec. 2022, Art. no. 106834, https://doi.org/10.1016/j.aap.2022.106834.

[14] "Test Protocol - AEB VRU Systems v. 3.0.3," European New Car Assessment Programme (Euro NCAP), 2020.

[15] "AEB Car-to-Car systems Test Protocol v. 3.0.3," European New Car Assessment Programme (Euro NCAP), 2021.

[16] "Test Protocol - AEB VRU systems v. 2.0.2," European New Car Assessment Programme (Euro NCAP), 2020.

[17] "Test Protocol - AEB systems, v. 2.0.1," European New Car Assessment Programme (Euro NCAP), 2017.

[18] L. Xia, T. D. Chung, and K. A. A. Kassim, "An Automobile Detection Algorithm Development for Automated Emergency Braking System," in *Proceedings of the 51st Annual Design Automation Conference*, San Francisco, CA, USA, Jun. 2014, https://doi.org/10.1145/2593069.2593083.

[19] S. Anand and S. S. Ohol, "Modelling and simulation of adaptive cruise control and overtake assist system," *Materials Today: Proceedings*, vol. 72, pp. 1353–1360, 2023, https://doi.org/10.1016/j.matpr.2022.09.330.