IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2023, accepted 28 June 2023, date of publication 10 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3293659

RESEARCH ARTICLE

Architecture and Decision-Making for Autonomous Tram Development

KHANSA SALSABILA SUHAIMI[1], ABDURRAAFI' SYAUQY[1], MOHAMMAD SALMAN SUBKI[1], BAMBANG RIYANTO TRILAKSONO[1,2], (Member, IEEE), ARIEF SYAICHU ROHMAN[1], (Member, IEEE), YULYAN WAHYU HADI[1], (Member, IEEE), HANDOKO SUPENO[1], DHIMAS BINTANG KUSUMAWARDHANA[1], AND DEWI NALA HUSNA[3]

[1]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia
[2]University Center of Excellence of Artificial Intelligence on Vision, NLP and Big Data Analytics (U-CoE AI-VLB), Institut Teknologi Bandung, Bandung 40132, Indonesia
[3]PT Industri Kereta Api (INKA), Madiun 63122, Indonesia

Corresponding author: Arief Syaichu Rohman (arief.rohman@itb.ac.id)

This work was supported by the Ministry of Finance of the Republic of Indonesia under the Lembaga Pengelola Dana Pendidikan (LPDP) Riset Inovatif Produktif (RISPRO) Research Funding and the Indonesian Ministry of Education and Culture under the World Class Research (WCR) Program.

ABSTRACT A decision-making system is an essential component of an autonomous tram. The decision-making system uses information on the surrounding area to recognize the traffic situation and determine appropriate actions to maintain safety and passenger comfort. However, guaranteeing reliable performance and safety remains challenging in all driving situations. The development of autonomous trams involves iterative engineering-related tasks. Therefore, proper architecture to facilitate the flexibility of engineering development is required. This paper proposes a modular architecture for a decision-making system for autonomous trams. The decision-making system can serve as a high-level controller for autonomous trams. The architecture consists of risk assessment and decision & planning modules. It also integrates several key functions, such as trajectory prediction, safety assessment, adaptive cruise control (ACC), collision avoidance (CA), and emergency braking system (EBS). In the decision and planning module, a finite-state machine is devised as part of the decision-making system. This module provides a speed reference for low-level speed controllers. In addition, the decision-making system architecture was implemented and its implementation was validated using a Carla simulator. Under a mixed-traffic scenario, simulation results showed that the decision-making system has a high percentage of mission successes. Out of 50 simulations in mixed-traffic scenarios, the tram reached its destination with an 80% success rate in which the success rates of ACC, CA, and EBS executions to avoid collisions were 96.94%, 100%, and 100%, respectively. In addition, the system works well in real-time to recognize the surrounding environment and determine actions. Although beyond the scope of the decision-making system, the simulation results also indicate that an improving performance of the tram's low-level speed controller may provide more reliable performance of the decision-making system.

INDEX TERMS Autonomous tram, Carla simulator, collision avoidance, decision-making system.

I. INTRODUCTION

Over the years, the growth of private transport has continued and traffic flow has become increasingly congested, impacting community productivity and risking safety. Congestion is an unresolved problem in large cities in Indonesia. The fear of total congestion (deadlock) has led to the implementation of mass public transportation, namely tram and light rail transit (LRT) [1]. Tram is a type of light rail vehicle (LRV) that operates on urban roads by obeying traffic signals [2] and interacting with other road users [3]. In Indonesia, the tram is an uncommon mode of mass transportation for the public. In the analysis issued in [4], it is pointed out that there are differences between public transport situations in

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

Indonesia and European cities. In Indonesia, most road users are less aware of safety and discipline [4]. Meanwhile, tram lines are more likely to mix with other vehicles and burdensome traffic conditions should be considered [4]. These conditions cause safety assurance problems because they affect the driver's fitness and cause human error [5], [6]. Data analysis [7] shows that accidents involving trams are caused more by negligent road users, such as misinterpreting tram line signs, miscalculating tram stop times, and passing without paying attention to tram lines.

Several studies on autonomous trams have been conducted. The study in [8] proposed a collision avoidance system for trams using Light Detection and Ranging (LiDAR). This proposal used the speed of the tram and the distance to an object to determine how fast the tram could brake and how long a braking distance would be. Thus, the necessity of braking is determined deterministically based on the braking distance and the distance to the object as a risk assessment. In addition, the proposed system provides an emergency braking signal if the distance to the object is minimal or if the object does not react to a given warning. A similar study was conducted in [9] using a Sick Ladar Digital-Multilayer Range Scanner (LD-MRS) laser scanner. The proposal includes a system architecture that comprises laser scanner sensors, object recognition, and risk assessment. The risk assessment estimates the time to collision based on the knowledge of relative velocities and the shortest distance. It then initiates a tram's stop if the tram driver does not react after being given a warning. Instead of using the deterministic approach, another study [10] utilized a probabilistic threat assessment that considers position, velocity, and classes for each of the road users. This study also did not implement active control of the tram but used a warning. However, the risk assessment method in [8], [9], and [10] would not properly handle the dynamic motion of objects as none of these studies have predicted the future movement of an object.

Unlike the studies in [8], [9], and [10], the framework proposed in [11] built a model that considered the status of trams and objects, restrictions, and safety qualification rules. The model was optimized for safety, energy cost, convenience, parking accuracy, and timeliness. Risk quantification was performed by quantifying four types of hazards: forward collision, rear-end collision, lateral collision, and overspeed. However, the proposed framework has not been studied further.

Another study proposed an autonomous tram with a signaling system, as in [2]. This proposal includes a system architecture comprising of localization, signal handling, obstacle handling, and vehicle control. The signal handling is used to detect tram signals and determine the movement authority limit (MAL) for compliance with these signals. Meanwhile, the obstacle handling detects and responds to dynamic objects by determining the MAL required to avoid dangerous objects. The proposed autonomous tram was successfully demonstrated on Germany's 6 km long Potsdam tram network in which crossing animals are frequently encountered in this network. In addition, the network included several signalized and non-signaled road crossings. However, this proposal did not consider the predicted future movement of an object. Instead, free-space detection is used to determine whether an area is free or occupied. The use of free-space detection is challenging owing to the growing vegetation on the track, and this requires a trade-off between the ability to detect objects close to the ground without detecting tall vegetation. It is noted that driving trams based on MAL require a signaling system.

Autonomous driving in the automotive and railway sectors takes a different approach. Autonomous driving in the train railway sector requires trains to run on dedicated tracks and continuously communicate with the train control room. By contrast, autonomous driving in the automotive sector must interact with a mixed traffic environment [7]. Even though a tram operates on a railway, the tram needs to interact with a mixed traffic environment, and an autonomous vehicle architecture is, therefore, more suitable for use in autonomous trams. From an architectural perspective, the work in [12] described an overall autonomous vehicle architecture that includes perception, understanding, decision-making, and actuation capabilities. A recent review of architecture for autonomous vehicles was conducted by [13], in which the architecture was divided into three types: sequential, behavior-aware, and end-to-end planning. An example of a sequential planning approach was used for urban driving, as in the DARPA urban challenge [14]. This approach decouples the perception, mission planning, motion planning, and vehicle control. Although the sequential planning approach has resulted in the successful implementation of DARPA's urban challenges, it does not consider the modeling of vehicle interactions. Unlike in [14], the work in [15] used a behavior-aware planning architecture that considers human-like driving behavior in real traffic scenarios. Meanwhile, advanced approaches, such as in [16], that use learning-based methods have been studied for several end-to-end autonomous vehicle architectures. These approaches make use of sensor data directly to generate control commands through a control model trained using machine learning. However, there have been no actual implementations or tests of these end-to-end approaches. Instead, most implemented prototypes use a sequential planning architecture [17], which has modular characteristics.

Decision-making is crucial for taking over the tram driver's complex tasks in an urban setting by providing an appropriate response to changes in traffic situations and an appropriate architecture for autonomous trams may be needed. Building an autonomous tram also engages in engineering work in which an agreed architecture of the decision-making system shall be available to refer to in the development process. Currently, there are many available choices and combinations of technology to implement the complex functions of an autonomous tram; hence, a modular architecture, such as sequential planning architecture, may be a suitable choice to facilitate implementation flexibility.

The autonomous tram reported in [2] was successfully demonstrated with the support of a signaling system, which was not the case in Indonesia with current infrastructure support. In this infrastructure context, the approaches in [8], [9], [10], and [11], which use risk assessment methods, are considered suitable for deployment in Indonesia. A risk assessment of the traffic environment using a new concept in the traffic safety field was proposed in [18]. However, the new concept is suitable for road vehicles within a straight road scenario, which is a different scenario from the case of the tram situation. In terms of the level of autonomous driving assistance (known as ADAS in the automotive industry [12]), the approaches in [8], [9], [10], and [11] achieved good results. However, it is clearly insufficient for an autonomous tram to rely on warnings, wait for the tram driver to react, or stop the tram if no action has been taken by the driver. In addition, in mass public transportation, making trams autonomous is not only about safety but also the convenience and quality of service (i.e. keeping to the timetable), as considered in [19] and [20]. Hence, more features should also be performed, including railway cruising, collision avoidance, and emergency braking. Predicting the future trajectory of an object is also essential for achieving those features, which have not been included in previous studies. Moreover, tram-restricted conditions in a railway provide common tram situations such as vehicle crossings, pedestrian crossings, and vehicles blocking a part of the railway. Therefore, cruising on a railway while maintaining a distance when an object partially or fully blocks the railway and anticipating object crossing must be ensured.

This paper presents a modular decision-making architecture of autonomous trams containing risk assessment and decision & planning modules in which a risk assessment method combined with a state machine to switch between predefined behaviors is employed. This architecture enables modular autonomous trams design to consider unknown situations as new states if it is necessary. This study enhances an autonomous tram architecture without the need of signaling infrastructure. The main contribution of this study is the development of a simple, safety and performance-achieving, and feasible decision-making architecture for the tram to cruise on its railway, maintain distance to in-front object, and anticipate crossing objects. Both the risk assessment and the decision & planning modules are designed to achieve safety and intended performance. In the risk assessment module, trajectory prediction for the surrounding objects, railway and the tram information are used to assess any risk. Meanwhile, the decision and planning module covers three actions: adaptive cruise control, emergency braking, and collision avoidance. Following to the action taken, an appropriate speed reference is planned based on perceived risk level. The proposed architecture is also implemented using a Carla simulator to validate its feasibility, performance, and reliability.

The remainder of this paper is organized as follows. Following an introduction in this section, section II presents the architecture of the proposed decision-making system. An in-depth explanation of the risk assessment module is provided in Section III. Section IV presents the detailed architecture of the decision and planning module, and Section V presents the actuation necessary to complete the implementation in Carla simulator. Section VI presents the simulation results of the implemented architecture and the decision-making system tested using Carla simulator. Finally, Section VII presents conclusions and potentials for future work.

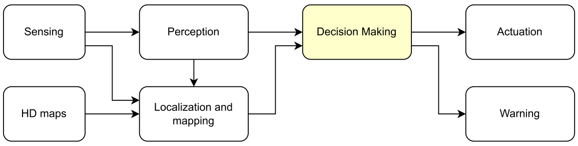

FIGURE 1. Autonomous tram architecture.

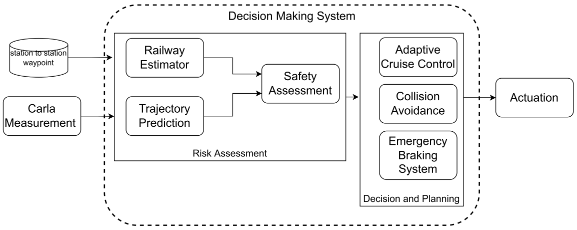

FIGURE 2. Architecture of decision-making system implemented in Carla simulator.

II. THE DECISION-MAKING ARCHITECTURE

In the autonomous tram architecture shown in Fig. 1, the decision-making system depends on the performance of other systems because it requires information regarding the object's state and the tram's state itself. Fig. 1 shows the position of the decision-making system in the autonomous tram architecture. It can be seen that the decision-making system obtains input from the perception, localization & mapping systems. The outputs generated by the decision-making system are then fed to the actuator and warning modules.

The architecture of the decision-making system is shown in Fig. 2. In the first layer, the architecture of the decision-making system is divided into risk assessment and decision & planning modules. In the second layer, the risk assessment module consists of a railway estimator, trajectory prediction, and safety assessment. Meanwhile, the decision and planning module comprises adaptive cruise control (ACC), collision avoidance (CA), and emergency braking system (EBS). The risk assessment module identifies traffic conditions and predicts hazards, whereas the decision and planning module handles action determination. Fig. 2 also shows the decision-making system implemented in Carla simulator. In Carla simulation scenarios, data from the perception and localization systems were obtained using Python API provided by Carla simulator, and the railway coordinates of the mapping were substituted by waypoints stored in a file. The actuation task was performed using a low-level controller. Notably,

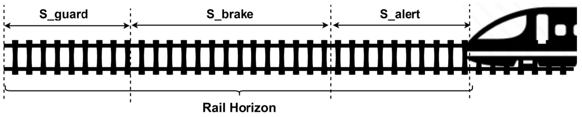

FIGURE 3. **Illustration of rail horizon.**

no warning task was included in the implementation due to limitations in the simulator.

A. RISK ASSESSMENT

This module aims to recognize traffic and predict possible collisions based on information obtained from the perception, localization & mapping systems. First, traffic situations are recognized by predicting the motion trajectory of all detected vehicles and pedestrians (referred to as objects) by trajectory prediction based on the historical information of the objects. The prediction of possible collisions for all detected objects is then performed using a safety assessment, which compares the predicted motion trajectories of these objects to a rail horizon, as depicted in Fig. 3. The rail horizon is the zone in front of the tram that must be monitored. Its length is calculated using a railway estimator. After calculating the collision probability for all the objects, the object with the most significant collision risk can be determined. This information is then forwarded to the decision and planning module to determine the required action.

B. DECISION AND PLANNING

The decision and planning module aims to plan actions based on changes in traffic situations identified and predicted by the risk assessment module. Information from the risk assessment module, that is, the position and speed of the tram, speed of the object, predicted location and time of the collision, is considered by the decision and planning module to yield effective and safe actions. In each frame, the decision and planning module selects one of the three states: emergency braking system (EBS), collision avoidance (CA), and adaptive cruise control (ACC). A tram speed profile was produced to accelerate, decelerate, or instantly stop the tram in each state. The speed profile is passed to a lower-level controller in the actuation system, which is beyond the scope of the decision-making system.

III. RISK ASSESSMENT MODULE

A. RAILWAY ESTIMATOR

The railway estimator uses the tram speed information to determine the current rail horizon. The railway estimator's basic concept was modified from traffic alerts and braking commands in a rail collision avoidance system (RCAS) [21].

$$S_H = S_{alert} + S_{brake} + S_{guard}. \tag{1}$$

The rail horizon S_H in (1) comprises three parameters: warning distance S_{alert}, maximum braking distance S_{brake}, and minimum braking distance S_{guard}. S_{alert} is the approximate distance to the warned objects, allowing objects that will cause collisions to get out of the way, and the tram does not have to slow down. Although a warning is not yet involved, it is still necessary to consider S_{alert} to describe natural conditions. S_{brake} is the farthest distance required to stop using minimum deceleration. However, S_{guard} estimates the shortest distance for a complete stop at the operating speed in mixed-traffic using maximum deceleration. Combining these three parameters can be used to determine the required viewing distance for each change in the tram speed.

B. TRAJECTORY PREDICTION

For each object perceived by the perception system, trajectory prediction predicts the object's attributes when moving or stopping based on its historical position and speed. The actual output of the trajectory prediction contains trajectory points that include time and speed information. Trajectory prediction constructs the object's motion trajectory by following a sequence of predicted paths. A simple method is to use physical rules with suitable assumptions.

In this study, a Kalman filter was used to predict the motion trajectories of objects. The Kalman filter uses the feedback control principle to estimate the state at a certain point in time and receives feedback in the form of noise measurements [22]. Kalman filter prediction is equivalent to filtering when the measurement data are unavailable or unreliable [23]. In this context, the Kalman gain is considered zero, which means that the calculated value is close to the actual value. The Kalman filter formula is divided into time-updated and measurement-value-updated formulas [22]. The time-updated formulas estimate the state and covariance for the next time step using the transition matrix and process noise. By contrast, the measured value-updated formula inserts a new measurement into the previous state estimate to obtain a new but better estimate.

C. SAFETY ASSESSMENT

In general, without knowing the exact position and direction of objects on a map, the safety assessment relies on attributes provided by the trajectory prediction, such as moving and stopping. The safety assessment has 2 (two) assessment conditions: moving objects and non-moving objects. The safety assessment checks whether the object's position is within the rail horizon for objects that are predicted to stop moving. As for moving objects, the safety assessment checks whether there is an intersection between the object's predicted motion trajectory and the rail horizon. The results of the assessment are time-to-region (TTR), as in (2) and (3), and the distance-to-collision (DTC), as in (5). Then, TTR becomes TTC if it satisfies (4), where α is the contention parameter [24].

$$TTR_1 = \frac{|cp - p_{tram}|}{v_{tram}} \tag{2}$$

$$TTR_2 = \frac{|cp - p_{OV}|}{v_{OV}} \tag{3}$$

$$|TTR_1 - TTR_2| \leq \alpha \tag{4}$$

$$DTC = v_{tram} TTC \tag{5}$$

where TTR_1 : time to tram region. TTR_2 : time to region for object. TTC : time to collision. cp : crossing point. p_{tram} : tram position vector. p_{ov} : object position vector. α : contention parameters.

A threat score (φ_{o_j}) is computed for each object, the j^{th} object. The object with the highest risk of collision was then determined by comparing the threat scores of all objects. In [25], the threat score was calculated using a conditional random field algorithm that considers the TTC and time-to-stop (TTS). In this study, the TTS was calculated based on the current speed of the tram using the maximum deceleration. In addition, a non-negative real number as a measure of the threat score presented in (6) was employed instead, where Δt_e is the time difference between TTC and TTS, σ is a tuning constant, and $sgn(.)$ is a signum function. The highest threat score (δ) is then computed using (7).

$$\varphi_{o_j} = e^{-0.5(\Delta t_e/\sigma)^2 sgn(\Delta t_e)} \tag{6}$$

$$\delta = \arg\max_{j}\{\varphi_{o_j}\} \tag{7}$$

IV. DECISION AND PLANNING MODULE

In this module, a finite-state machine (FSM) is used to determine actions of cruising on the railway, maintaining distance, decelerating, or stopping. The FSM design has three states, as shown in Fig. 4. The ACC state is related to cruising on railways and distance-keeping. The CA state is related to collision avoidance through gradual slowing, whereas the EBS state is related to collision avoidance through immediate stopping.

The state selection was performed based on the specified transition parameters. The first parameter is the safe distance to the object. The system is in the ACC state only if a safe distance from the object is achieved. This state is the initial state when the tram starts to operate. If the safe distance is not satisfied, then the ACC state moves to the EBS state. The second and third parameters represent TTC and DTC [25] with the highest collision risk. When TTC and DTC were below the specified safe limit, the state moved to the CA state. The fourth parameter was the tram speed. When the current state is a CA or an EBS state, a transition to the ACC state occurs when the tram speed is zero.

Once the current state is determined using the FSM, the associated reference speed profile is calculated within its associated state. This associated state acts as a high-level speed controller; thus, the ACC, CA, and EBS algorithms do not consider the tram dynamics when calculating the reference speed profile. Instead, the tram dynamics are considered in the low-level speed controller, which is discussed in the next section. The algorithms used by the EBS, CA, and ACC states to generate the speed profile are as follows.

A. EMERGENCY BRAKE SYSTEM

The speed profiling algorithm of the EBS state provides emergency braking to stop the tram within the shortest possible time to avoid highly close collisions. This state generates a zero-speed profile if the safe distance calculated using non-uniform linear motion (8) meets the emergency condition, as in (9). The notation v_{mt} is the operational speed in mixed traffic situations and the notation a_{max} is the maximum deceleration of the tram.

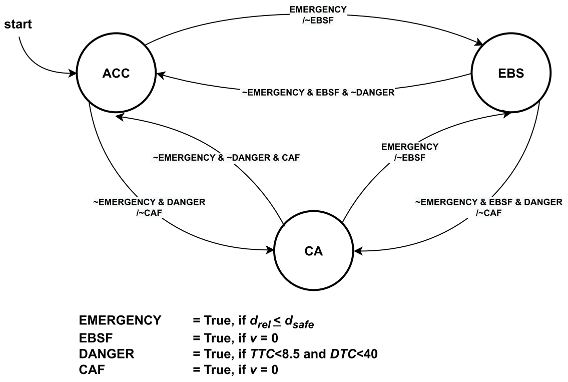
start
ACC
EBS
CA
EMERGENCY
/~EBSF
~EMERGENCY & EBSF & ~DANGER
~EMERGENCY & ~DANGER & CAF
EMERGENCY
/~EBSF
~EMERGENCY & DANGER
/~CAF
~EMERGENCY & EBSF & DANGER
/~CAF
EMERGENCY = True, if $d_{rel} \leq d_{safe}$
EBSF = True, if $v = 0$
DANGER = True, if TTC<8.5 and DTC<40
CAF = True, if $v = 0$

FIGURE 4. FSM of decision and planning.

$$d_{safe} = \frac{v_{mt}^2}{2a_{max}} \tag{8}$$

$$v_{ref} = \begin{cases} 0 & ; d_{rel} \leq d_{safe}, \\ otherwise & \end{cases} \tag{9}$$

In Fig. 4, a transition from either the ACC or CA state to the EBS state occurs when the 'Emergency' condition is True; that is, the object's relative distance to the tram is less than or equal to the safe distance, and output of this transition is 'EBSF' (EBS Finished) with a value of False. Thus, the EBS state remains a priority until the tram stops and no other object is too close to the tram.

B. COLLISION AVOIDANCE

In Fig. 4, a transition from ACC to CA state occurs when the 'Danger' condition is True, i.e., $TTC < 8.5\ s$ and $DTC < 40\ m$, and the transition output is 'CAF' (CA Finished) condition with a False value. Meanwhile, the EBS state transition to the CA state occurs when the following three conditions are met, i.e. the 'Danger' condition is True, the 'Emergency' condition is False, and the 'EBSF' condition is True. This transition responds to a predicted collision when a safe distance is met, and the tram has stopped successfully. The output of this transition is 'CAF' with a false value.

As the current state moves to the CA, the speed profiling algorithm of the CA state provides a zero-speed profile in which the low-level speed controller (i.e. smooth brake actuation; see the next section) must incrementally execute. This algorithm uses fuzzy logic to calculate the required speed reduction. The fuzzy logic is adapted to provide a smooth transition in speed reduction as a train-braking system in [26]. The adapted algorithm uses the Mamdani inference method based on the DTC and TTC [27]. The smaller the DTC and TTC are, the more likely that the tram will hit an object in front of it. In this fuzzy controller, DTC and TTC act as inputs, whereas the amount of braking is the output.

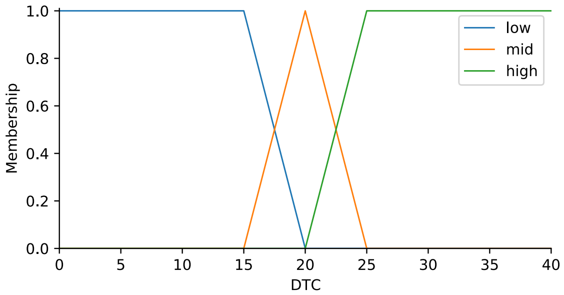

FIGURE 5. Input DTC membership function.

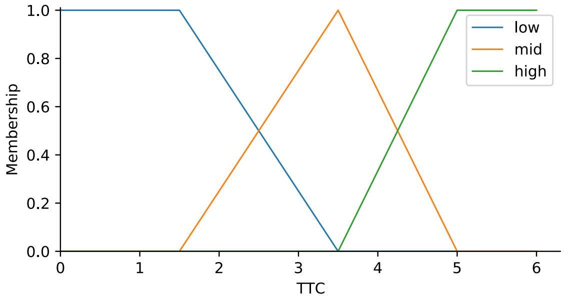

FIGURE 6. Input TTC membership function.

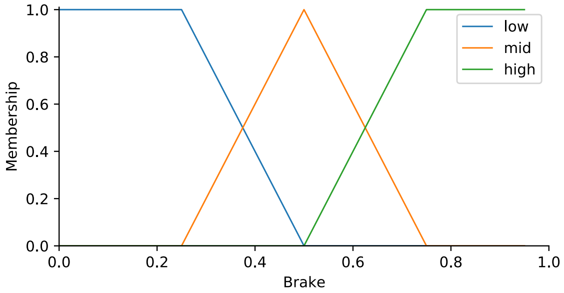

FIGURE 7. Output brake membership function.

The Mamdani fuzzy inference method has 3 (three) processes for producing an output: fuzzification, inference, and defuzzification. In the fuzzification process, the degree of membership functions of the inputs and outputs are determined based on the trapezoidal and triangular functions shown in Fig. 5, 6, and 7 for DTC, TTC, and output, respectively.

The degree of membership function is on the graph's y-axis and ranges from 0 to 1. The degree of membership was used in the inference process. When DTC is in meters (m) and TTC in seconds (s), the brake ranges from 0 to 1. In addition, the rules depend on DTC and TTC inputs in the inference process. In turn, the rules determine the braking level of the output. In particular, the number of rules is obtained by multiplying with the number of inputs, as shown in (10).

$$\begin{aligned} TotalRule &= numberofDTC \times numberofTTC \\ TotalRule &= 3 \times 3 = 9 \end{aligned} \tag{10}$$

The implication function and composition of the rules were applied during the inference. The implication function is beneficial for determining the cause-and-effect relationship between input and output. The implication function is defined as in (11) [28].

$$\begin{aligned} \alpha - predicate_i &= \mu_{A1}[x_1] \cap \mu_{A2}[x_2] \\ &= \min(\mu_{A1}[x_1], \mu_{A2}[x_2]) \end{aligned} \tag{11}$$

where i is the i^{th} fuzzy rule, A_1 is the first input set, A_2 is the second input set, x_1 is the first input value, x_2 is the second input value, and μ is the degree of membership. Rule composition combines the degree of membership of the rule implication function with the maximum value of the rule. The merging of membership degrees using the maximum method is defined in (12).

$$\mu_{sf}(x_i) = \max(\mu_{kf}(x_i), \mu_{kf}(x_i)) \tag{12}$$

where $\mu_{sf}(x_i)$ is the degree of membership of the fuzzy solution up to the i^{th} rule, and $\mu_{kf}(x_i)$ is the fuzzy consequence of the i^{th} rule [27]. The results of the inference process form a new fuzzy set based on the mergers. In the defuzzification process, the area of the new fuzzy set is determined using the centroid method. The calculation of the braking reference using the centroid method is shown in (13).

$$brake_{CA} = \frac{\int \mu(z)zdz}{\int \mu(z)dz} \tag{13}$$

where $brake_{CA}$ is the result of the defuzzification or the middle value of the fuzzy area, $\mu(z)$ is the degree of membership, and $\int \mu(z)zdz$ is the moment of the rule area [28].

C. ADAPTIVE CRUISE CONTROL

The speed profiling algorithm of the ACC state provides a speed profile for cruising on the railway and adapting to the presence of another tram (or object) ahead. This state is divided into two substates, 'normal' ACC and 'keep distance' ACC. The 'normal' ACC generates a speed profile for cruising on the railway based on operating speed rules if the situation is safe. The 'keep distance' ACC generates a speed profile that is lower than the current speed of the tram. This is to ensure that the tram can maintain a safe distance when approaching another object in front of it.

In Fig. 4, the 'start' arrow indicates that the initial condition was the ACC state. The transition from the EBS or CA state to the ACC state occurs when the 'Emergency' and 'Danger' conditions are false. This transition implies that the following three parameters-safe distance, TTC limit, and DTC limit-are not violated, and the 'EBSF' or 'CAF' condition is true, which corresponds to an event in which the tram stopped successfully after executing EBS or CA state.

Model predictive control (MPC) algorithm is used to control the distance and longitudinal speed of the tram relative to another object in front, as in [29] and [30]. MPC may be considered a suitable approach because it can deal with both input and output constraints and has a smooth output [30].

Safe distance and passenger comfort were assigned as constraints to ensure that the distance was sufficient for a sudden stop and that the acceleration or deceleration did not exceed the comfort limit. Furthermore, MPC minimizes cost function based on predictions, which is the square of the error between the desired output and actual output. The algorithm is shown in (14)-(18).

$$v_{i+1} = v_i + a_i t \tag{14}$$

$$d_{rel,i+1} = d_{rel,i} - v_{rel,i} t - \frac{1}{2} a_i t^2 \tag{15}$$

v_i, $d_{rel,i}$ and $v_{rel,i}$ are the tram's speed, the distance and speed relative to another tram, respectively, at the i^{th} time step. a_i is the given acceleration/deceleration value of the tram. t is the change in time between time steps.

$$\begin{aligned} C = \sum_{i=0}^{N} & ((w_d(d_{rel,i+1} - d_{safe}))^2 \\ & + (w_v(v_{rel,i+1} - v_{mt}))^2 \\ & + (w_{rate}(a_{i+1} - a_i))^2 \\ & + (w_a(a_{i+1}))^2) \end{aligned} \tag{16}$$

The predicted future conditions by MPC may consist of several steps as needed; however, only the first step's acceleration or deceleration value is used for this purpose. The cost function used to make predictions is given by (16) [27], where d_{safe} indicates the desired safe distance between the tram and another object, and v_{mt} is the tram operating speed in mixed-traffic situations. a_i is the tram acceleration at the i_{th} time step, and N represents the number of predictive steps. w_d and w_v are the weights representing the safe distance control and speed control. w_{rate} and w_a are the weights representing the change of acceleration/deceleration and acceleration/deceleration value.

The w_d, w_v, w_{rate}, and w_a can be set to obtain the desired control behavior. The greater the weight, the more the related variables are prioritized to approach the desired value. The cost function is minimized so that the relative distance and velocity values have less error for the given reference at each predicted step. Thus, the acceleration or deceleration values can be obtained.

$$d_{safe} = \frac{{v_{mt}}^2}{2a_{max}} + v_{rel} t_{gap} \tag{17}$$

The safe distance value (d_{safe}) in (17) is the minimum safe distance that must be satisfied [31]. Note that t_{gap} is the time required for the tram to begin realizing the need to maintain distance. Where a_{max} is the maximum deceleration value that can be achieved, and v_{mt} is the operational speed in mixed-traffic situations.

$$v_{ref} = v_{actual} + at \tag{18}$$

The acceleration or deceleration value obtained from the MPC algorithm a is used as a reference for speed changes. Therefore, the low-level controller receives the calculated speed profile in (18).

V. ACTUATION

The actuation system is a low-level controller for an autonomous tram. The task of this system is to make the actual speed of the tram following the speed profile generated by the decision-making system. Although this system is beyond the decision-making system, it completes the implementation of the autonomous tram architecture (Fig. 1) in Carla Simulator.

As the simulator does not provide a tram model, the firetruck model is chosen as the tram model because its dynamics exhibit similar behavior. The predefined firetruck model of Carla simulator uses NVIDIA PhysX vehicle model to simulate its vehicle dynamics. In Carla simulator, the actuation system drives the tram by providing throttle, brake, and steering control signals to its dynamic model. This simulation requires a steering wheel signal to track waypoints assumed to be tram railways.

There are three actuation methods in this system, i.e. maximum brake, smooth brake, and proportional-integral-derivative (PID). Each of these functions is a low-level controller of a particular FSM state. The maximum brake receives only a simple zero-speed profile from the EBS state to stop as quickly as possible with maximum braking. Meanwhile, the smooth brake receives the speed profile from the CA state, and the PID receives the speed profile from the ACC state, as described below.

A. SMOOTH BRAKE

The smooth brake method adjusts the control signal so that a speed decrease does not interfere with passenger comfort. This method was intended to track the speed profile generated by the CA state. The initial value of the brake control signal is zero and then increases, such that the tram stops before the collision point. The change in the brake control signal (*brake*) is calculated using a linear equation, as shown in (19).

$$brake = brake_{CA} \frac{d}{d_{DTC}} \tag{19}$$

$brake_{CA}$ is the brake reference calculated using (13), which is the value of the intended brake, and d is the distance traveled since the brake was applied. If the brake control signal reaches $brake_{CA}$ and d reaches d_{DTC} in a given situation, the smooth brake applies the maximum brake control signal to prevent a collision.

B. PID CONTROL

The PID method provides a throttle control signal based on the speed profile given by the ACC state. The PID calculates the error between the speed profile (18) and the tram's actual speed, as in (20), and then computes the throttle control signal (*throttle*), as in (21).

$$e = v_{ref} - v_{actual} \tag{20}$$

$$throttle = K_p e + K_i \int edt + K_d \frac{de}{dt} \tag{21}$$

The PID gains K_p, K_i, and K_d were obtained by manual tuning. Initial gains were randomly set to a small value, then they were changed based on observing the speed response of the tram. Several simulations were run specifically for tuning by comparing desired speed with the actual speed until the desired steady-state speed was achieved.

VI. SIMULATION RESULTS AND DISCUSSION

A. SIMULATION SETUP

Carla simulator is an autonomous driving simulator that provides maps with various characteristics, types of actors, and accessible information about the actors and their environments [32]. Town02 map of Carla simulator was selected to model the urban mixed-traffic environment because it is the closest simulated city environment to the city environment in Indonesia. Actors such as vehicles and walkers (or agents) can spawn and move in autopilot mode using the Carla simulator traffic manager module. By default, this module sets agents to follow a dynamically produced trajectory and randomly chooses a path as they approach a junction with 70% of their current speed limit. Other actors provided by the Carla simulator are sensors, i.e. LiDAR, radar, IMU, GNSS, and others, that can be attached to a vehicle to gather information about its surroundings. These sensors produce raw data that can be utilized by localization and perception systems; however, these sensors are not used in this simulation. Instead, decision-making in this simulation uses the Python API class from Carla simulator, which defines information on the location and rotation of agents.

Although the environment provided by the simulator was similar to that in real urban driving, some differences were observed. The road network in Town02 of Carla simulator is a single lane in each direction, where one of them is the lane in which the tram drives. Therefore, no other vehicle drives in the same direction on the tram side, and no other vehicle cuts the lane from the side of the tram. In consequence , the behavior of other vehicles remains in their lane. Moreover, unlike real urban driving in which human drivers of the other vehicles may cause unpredictable situational awareness, other vehicles move freely as long as there are no other agents in front of them in a simulation environment. A warning horn for the tram is also not supported, and agents cannot respond to the warnings by default.

B. PARAMETER SETTINGS

Specific parameters must be defined to simulate the decision-making system in Carla simulator. In a simulation environment, Carla simulator used throttle and brakes with a scalar value of 0 to 1 for each vehicle type. This scalar value is a relative value among all vehicles in Carla simulator and represents the minimum and maximum values of throttle and braking. The inertial characteristic of a tram is quite different from the other objects. Therefore, the maximum throttle value was assumed to be half of the maximum value.

There is a comfort limit, and an assumed operational speed limit on the algorithm side. The comfort limit is $-1\ m/s^2$ to $1\ m/s^2$, and the operational speed in the station and urban areas is 2.78 m/s and 5.56 m/s. Therefore, the tram speed will not exceed 5.56 m/s in this simulation. Moreover, the tram is assumed to have a minimum deceleration of 0.5 m/s^2 and a maximum of 1 m/s^2. The operational speed, minimum deceleration, and maximum deceleration are used to compute Equations (1), (8), (16), and (17). In addition, the cost function of the MPC algorithm in (16) has weights that a tuning process must select to achieve desired control behavior. The weights used are listed in Table 1. Meanwhile, Fuzzy logic was designed using the membership function shown in Table 2.

TABLE 1. MPC algorithm weights.

Parameter	Quantity
w_d	100
w_v	10
w_{rate}	5
w_a	3

TABLE 2. Set of membership functions of the input and output for the fuzzy controller.

Membership Function	Quantity
DTC high	$20m < s < 40m$
DTC medium	$15m < s < 25m$
DTC small	$0 < s < 20m$
TTC high	$5.5s < t < 8.5s$
TTC medium	$3.5s < t < 7s$
TTC small	$0 < t < 5.5s$
Brake high	$0.5 < b < 1$
Brake medium	$2.5 < b < 7.5$
Brake small	$0 < b < 0.5$

C. SIMPLE SCENARIOS SIMULATION

We simulated inter-station travel through Town02 using the Carla simulator, as shown in Fig. 8. A trip consists of several scenarios that a tram typically encounters. In this simulation, after the tram leaves the station, the tram encounters object A, which is moving at a varying speed. Then, objects B and C cut across the tram's lane at the first and second T-junction. After passing the second T-junction, object C moves in front of the tram and suddenly stops at two different points. Finally, the tram encounters object D, which cuts across the tram's lane.

In this simulation, the decision-making system was implemented in Carla simulator to observe the process of recognizing traffic situations, predicting collisions, and determining actions to be carried out. The tram must consider these three scenarios:

1) A scenario where an object cuts through the railway.
2) A scenario where some other trams or objects move at lower speeds in front of the tram.
3) A scenario in which an object in front suddenly stops.

At the beginning of the simulation, the tram received waypoints to reach its destination and perceived the surrounding environment. The railway estimator then determines the rail horizon distance based on the tram's current speed; the minimum distance is 15 m at 0 m/s, and the maximum distance is 57 m at 5.56 m/s. These distances are determined by S_{alert},

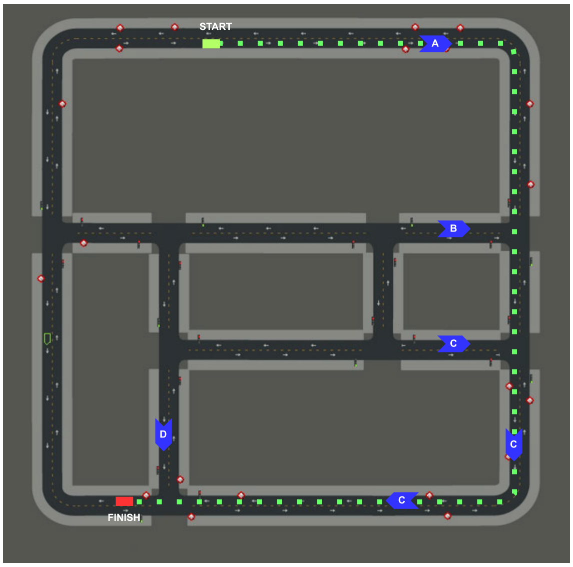

FIGURE 8. Simulation path: trams (*red*), railways (*green dots*), and objects (*A, B, C, and D*).

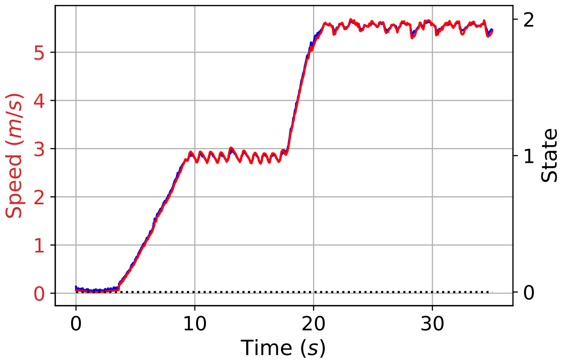

FIGURE 9. Speed profile of the 'normal' ACC state, i.e. State 0 (*dotted line*): reference speed (*blue*) and actual speed (*red*).

S_{brake}, and S_{guard} from (1). In this state, there are no objects within the perception radius, so the safety assessment will give a safe signal, and the FSM state is in 'normal' ACC. As shown in Fig. 9, when the tram is in the station area, the MPC algorithm will generate a speed profile for the recommended speed in the station area and a speed profile for the operational speed in mixed-traffic when entering an urban area.

After entering an urban area, the tram meets object A on the same lane, moving at varying speeds and stops. In this scenario, a state transition occurs between the 'normal' ACC state and the 'keep distance' ACC state, as shown in Fig. 10. When the relative distance between the tram and the object widens above 30 m, the FSM switches the state to the 'normal' ACC state. The given speed profile is, therefore, towards operational speed. However, when the object decreases its speed and the distance is sufficiently small, the FSM moves the state to 'keep distance' ACC states that provide a speed profile to reduce the speed. This behavior is shown in Fig. 10 and 11. The blue line in Fig. 10 shows

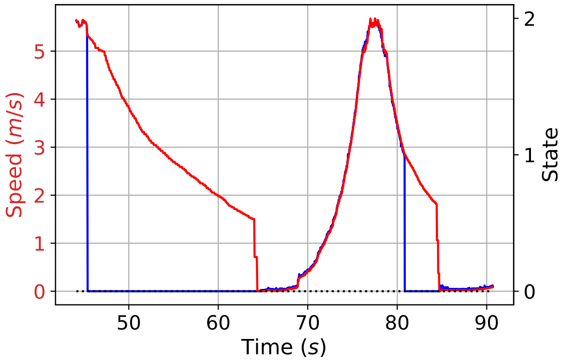

FIGURE 10. Speed profile of the scenario involving an in-front object with a lower speed: speed profile reference from MPC (*blue*), tram actual speed (*red*), and current state (*dotted line*), i.e. 'normal'/'keep distance' ACC (State 0).

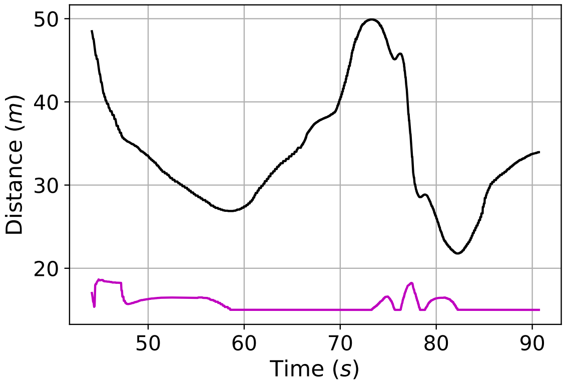

FIGURE 11. Relative distance profile of the scenario involving an in-front object with a lower speed: relative distance (*black*) and safe distance (*magenta*).

the speed profile generated by the MPC algorithm, and the red line shows the actual speed that tracks the speed profile. In Fig. 11, the black line shows the relative distance and the magenta line shows the safe distance that must be maintained. From these two figures, it can be observed that the tram managed to maintain its distance without crossing the safe distance limit, which varied with its relative speed.

The following scenario occurs when the tram crosses the T-junction and the perception system detects object B. The trajectory prediction receives the data, which then predicts the motion trajectory traversed by object B. Trajectory prediction using the Kalman filter method depends on the amount of historical data obtained from the observed object. Hence, trajectory prediction is relatively accurate when new historical data are obtained. In trials on Carla simulator, traffic situation s and dangers can be determined before the tram reaches the T-junction. The visualization shown in Fig. 12 depicts the intersection between the rail horizon and the predicted trajectory of the object that will cause a collision. This situation causes the state to move to the CA state, which constructs a speed profile to reduce speed using fuzzy logic, as shown in Fig. 13. Once the situation is safe, the state returns to 'normal' ACC.

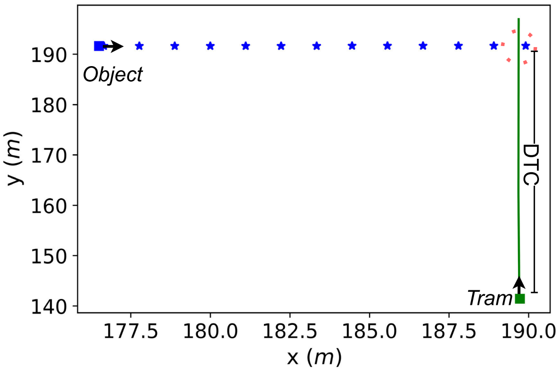

FIGURE 12. Visualization of rail horizon (*green line*), predicted trajectory (*blue star*), and predicted collision point (*red dotted line*) of scenarios with object crossing railways.

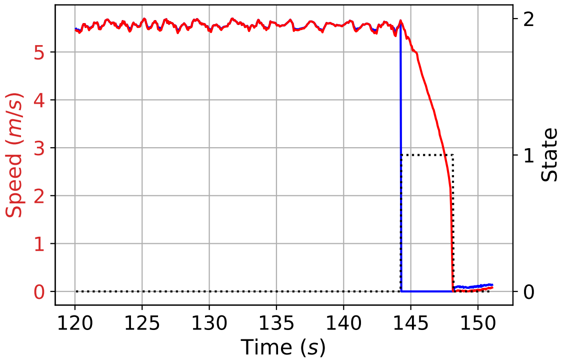

FIGURE 13. Speed profile of scenarios with object crossing the railways: reference speed (*blue*), actual speed (*red*), and current state (*dotted line*), i.e. 'normal' ACC state (State 0) or CA state (State 1).

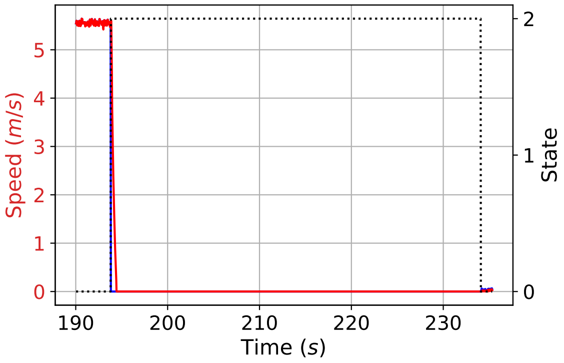

FIGURE 14. Speed profile of scenario with suddenly-stopped object: reference speed (*blue*), actual speed (*red*), and current state (*dotted line*), i.e. 'normal' ACC state (State 0) or EBS state (State 2).

In the following scenario, object C unexpectedly stops twice. This situation results in a shorter distance than the safe distance that must be maintained. In this situation, the FSM moves to the EBS state and provides a zero-speed profile to apply the maximum brake. The behavior of the tram in this scenario is shown in Fig. 14.

At the end of the trip, the tram perceives object D, which cuts the tram's lane. However, the decision-making system does not predict an impending collision, and the tram maintains its current speed. In this scenario, object D moves at a lower speed than the tram and will not meet at the T-junction. Therefore, the decision to stay in the 'normal' ACC state is valid.

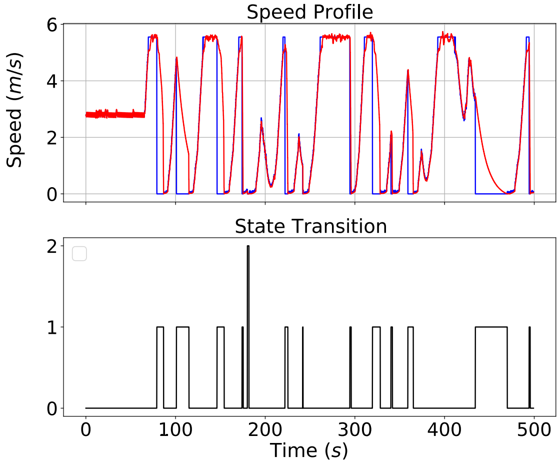

FIGURE 15. Speed profiles (*upper sub-figure*) and state transitions (*lower sub-figure*) in mixed-traffic simulation: reference speed (*blue*), actual speed (*red*), and state (*black*: 0 = ACC state, 1 = CA state, 2 = EBS state).

D. MIXED-TRAFFIC SCENARIO SIMULATION

We also simulated the decision-making system in mixed-traffic consisting of vehicles and pedestrians driven automatically and randomly using Carla simulator server. This test aims to evaluate the performance and reliability of the decision-making system in dense and diverse environments. In this test, the tram ran around Town02 and passed through four T-junctions. The number of vehicles and pedestrians in urban areas was set to 40, but this number could change if another object occupied the spawn point determined by the simulator.

Fig. 15 shows a plot of the speed profile generated in the ACC, CA, and EBS states in a 5-minute mixed-traffic simulation. The upper sub-figure shows reference and actual speeds, and the lower sub-figure shows the corresponding state. The figure shows that the tram is in the CA state 12 times, in the EBS state once, and in the ACC state 13 times. Transition to the CA state often occurs when a vehicle is about to turn at T-junctions, and several times when a pedestrian crosses. Meanwhile, the transition to the EBS state occurred at the T-junction when the vehicle turned near the tram. At 175 s in the simulation, the tram recognized that vehicle had stopped and maintained its speed. However, when the tram reached T-junctions, the vehicle moved suddenly and turned. In this case, a transition to the CA state occurs to prevent collisions. The transition is followed by an immediate transition to the EBS state (see the lower sub-figure of Fig. 15 at 175 s) because the distance between the vehicle and tram is sufficiently small, less than the safe distance.

The corresponding control signal for the speed profile in mixed-traffic simulation is presented in Fig. 16. It can be

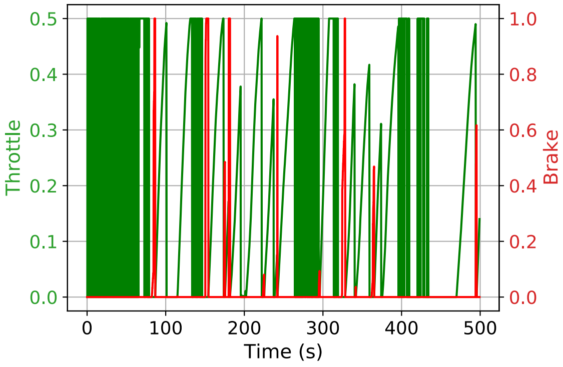

FIGURE 16. Control signals in mixed-traffic simulation: throttle (*green*) and brake (*red*).

observed that the control signal was saturated when tracking the given speed profile. Saturation occurs because the control signal is limited to a specific value, i.e. throttle and braking have a minimum value of 0, while the maximum value is 0.5 for throttle and 1 for braking. Signal saturation causes chattering phenomena as shown in the first 60 seconds of Fig. 15 when the actual speed oscillates. The other chattering phenomena are shown in Fig. 9, 13, and 14. In addition, the control signal at approximately 80 seconds is gradually braking to maximum (see Fig. 16) because collision points were quite close. Meanwhile, at 440 s, no braking signal was provided in the CA state, and only the throttle was reduced.

The mixed-traffic scenario simulation was run 50 times. The simulation was conducted to test the implementation of the decision-making system architecture in mixed-traffic scenarios. In the 50 simulations that involved a tram, vehicles (cars, motorcycles, trucks, buses, and bicycles), and pedestrians, the following situations emerged: (1) vehicles traveling on the railway at a lower or higher speed than the tram, (2) vehicles turning and traveling on the railway, (3) vehicles crossing the railway, (4) vehicles stopping suddenly, (5) traffic jams at the intersection, (6) pedestrians walking on the railway, and (7) pedestrians crossing the railway.

The simulation results showed that the tram can maintain safety by avoiding potential collisions in 40 simulations; therefore, the success rate of the decision-making system was 80%. It was observed that 14 collisions occurred when the FSM was in the ACC state, as shown in Table 3. Among these 14 collisions, three were caused by objects stopping on the railway and 11 were caused by objects that were about to cross the railway. In the three collisions caused by objects stopping on the railway, the ACC state cannot transition to the EBS state. In this situation, the objects had different directions from that of the tram. Noting that the distance of the object on the railway to the tram is calculated only when the object is in the same direction as the tram, the condition for the state transition is not met, and collisions consequently occur. Meanwhile, in the 11 collisions caused by objects that were about to cross the railway, the CA state was initially active and successfully stopped the tram. The CA state immediately transitioned to the ACC state. However, the ACC state could not transition back to the CA state to avoid collision because the calculated TTC was practically very large (the tram's zero speed was detected). From this perspective, in 50 simulations, the success rates of the ACC, EBS, and CA states in avoiding collisions were 96.94%, 100%, and 100%, respectively. These success rates shall not difficult to improve using the necessary settings or adjustments in the decision-making algorithm.

TABLE 3. Simulation results on the mixed-traffic scenario.

FSM State	Not Collision	Collision	Percentage of not collision
ACC	444	14	96.94
EBS	23	0	100
CA	411	0	100

VII. CONCLUSION

A modular architecture for the decision-making system of an autonomous tram was proposed herein. The proposed decision-making system has two modules: risk assessment and decision & planning. In the risk assessment module, the motion trajectory of an object was predicted using the trajectory prediction submodule. Then, the trajectory was compared with the rail horizon determined using the railway estimator submodule. The safety assessment submodule predicts whether an object may pose a hazard to the tram. The second module of the decision-making system is the decision and planning module, which consists of an adaptive cruise control (ACC) state, collision avoidance (CA) state, and emergency braking system (EBS) state. The module determines the actions required to maintain the safety of the tram based on a designed finite-state machine.

The decision-making system is simulated in Carla simulator with two different settings: (a) specified scenarios to observe the state transitions and speed profile of each state and (b) mixed-traffic scenarios to determine the system performance and reliability. Based on 50 simulation results of mixed-traffic scenarios, the success rate of the tram in reaching its destination was 80%. From another perspective, the success rates of ACC, CA, and EBS execution in avoiding collisions were 96.94%, 100%, and 100%, respectively. Collisions that occur in the ACC state are caused by errors in calculating the TTC (time to collision) when an object crosses the railway just after the tram stops and by errors in recognizing the situation as safe if an object on the railway is not in the same direction as the tram. In addition, the proposed decision-making system has considered trajectory predictions of objects but with constant-velocity assumptions. This may limit the performance of the proposed decision-making system. However, the performance of trajectory predictions may be enhanced by considering maneuvering objects with dynamic velocities.

The simulation results indicated that the proposed decision-making system architecture could perform collision avoidance and autonomous operations in real-time. In our future work, it will be necessary to develop a strategy and fine-tune the system to improve the performance of the decision-making system. The improvement is considering the size of the object, devising trajectory prediction for maneuvered objects, extending safety assessments to recognize and predict various hazardous situations, and improving speed profiling. With many choices of available technology for engineering realization of the decision-making system on a target hardware or platform, having a modular system architecture as proposed will make its implementation easier.

REFERENCES

[1] M. A. Farizi, "Pros and cons of the tram reactivation policy plan for the development of mass public transportation in Surabaya," M.S. thesis, Dept. Political, Airlangga Univ., Surabaya, Indonesia, 2019.

[2] A. W. Palmer, A. Sema, W. Martens, P. Rudolph, and W. Waizenegger, "The autonomous Siemens tram," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, Sep. 2020, pp. 1–6, doi: 10.1109/ITSC45102.2020.9294699.

[3] J. Zhang, Q. Lu, and S. Zao, "TCAS: Train collision avoidance system," in *Proc. 7th Int. Conf. Energy, Environ. Sustain. Develop. (ICEESD)*, 2018, pp. 1079–1085, doi: 10.2991/iceesd-18.2018.197.

[4] M. Damayanti, S. Malkhamah, and K. Walker, "Tramway management system in Indonesia," *J. Civil Eng. Forum*, vol. 1, no. 1, pp. 23–28, Jan. 2015, doi: 10.22146/jcef.22727.

[5] P. N. D. Putro, C. S. Wahyuning, and Yuniar, "Analysis of the level of stress and sleepiness of the operational area II machinist in Bandung," *Reka Integr.*, vol. 3, no. 1, pp. 1–12, 2015.

[6] A. L. Pramasari, B. Widjasena, B. Kurniawan, and S. Suroto, "Analysis of fatigue risk level for commuter line machinists on the Bogor-Jakarta city route," *Jurnal Kesehatan Masyarakat*, vol. 5, no. 2, pp. 93–99, Apr. 2017, doi: 10.14710/jkm.v5i2.16377.

[7] C. Di Palma, V. Galdi, V. Calderaro, and F. De Luca, "Driver assistance system for trams: Smart tram in smart cities," in *Proc. IEEE Int. Conf. Environ. Electr. Eng. IEEE Ind. Commercial Power Syst. Eur. (EEEIC/I&CPS Europe)*, Madrid, Spain, Jun. 2020, pp. 1–6, doi: 10.1109/EEEIC/ICPSEurope49358.2020.9160780.

[8] A. Patlins, N. Kunicina, A. Zhiravecka, and S. Shukaeva, "LiDAR sensing technology using in transport systems for tram motion control," *Elektronika Ir Elektrotechnika*, vol. 101, no. 5, pp. 13–16, 2010.

[9] M. Lüy, E. Çam, F. Ulamiş, Ì. Uzun, and S. Ì. Akın, "Initial results of testing a multilayer laser scanner in a collision avoidance system for light rail vehicles," *Appl. Sci.*, vol. 8, no. 4, p. 475, Mar. 2018, doi: 10.3390/app8040475.

[10] R. Katz and R. Schulz, "Towards the development of a laserscanner-based collision avoidance system for trams," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Gold Coast, QLD, Australia, Jun. 2013, pp. 725–729, doi: 10.1109/IVS.2013.6629553.

[11] K. Li, "The challenges and potential of risk assessment for active safety of unmanned tram," in *Proc. Int. Conf. Control, Autom. Inf. Sci. (ICCAIS)*, Hangzhou, China, Oct. 2018, pp. 22–27, doi: 10.1109/ICCAIS.2018.8570696.

[12] J. Ibañez-Guzmán, C. Laugier, J.-D. Yoder, and S. Thrun, "Autonomous driving: Context and state-of-the-art," in *Handbook of Intelligent Vehicles*, New York, NY, USA: Springer, pp. 1271–1310, 2012, doi: 10.1007/978-0-85729-085-4_50.

[13] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018, doi: 10.1146/annurev-control-060117-105157.

[14] M. Buehler, K. Iagnemma, and S. Singh, "The DARPA urban challenge: Autonomous vehicles in city traffic," in *Series Springer Tracts in Advanced Robotics*, vol. 56. New York, NY, USA: Springer-Verlag, 2009.

[15] Q. Sun, X. Huang, J. Gu, B. C. Williams, and H. Zhao, "M2I: From factored marginal trajectory prediction to interactive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 6533–6542, doi: 10.1109/CVPR52688.2022.00643.

[16] Y. Cai, S. Yang, H. Wang, C. Teng, and L. Chen, "A decision control method for autonomous driving based on multi-task reinforcement learning," *IEEE Access*, vol. 9, pp. 154553–154562, 2021, doi: 10.1109/ACCESS.2021.3126796.

[17] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6469–6486, Apr. 2021, doi: 10.1109/JIOT.2020.3043716.

[18] X. Zheng, B. Huang, D. Ni, and Q. Xu, "A novel intelligent vehicle risk assessment method combined with multi-sensor fusion in dense traffic environment," *J. Intell. Connected Vehicles*, vol. 1, no. 2, pp. 1–14, 2018, doi: 10.1108/JICV-02-2018-0004.

[19] A. Naweed and J. Rose, " 'It's a frightful scenario': A study of tram collisions on a mixed-traffic environment in an Australian metropolitan setting," *Proc. Manuf.*, vol. 3, pp. 2706–2713, Jan. 2015, doi: 10.1016/j.promfg.2015.07.666.

[20] I. Bae, J. Moon, and J. Seo, "Toward a comfortable driving experience for a self-driving shuttle bus," *Electronics*, vol. 8, no. 9, p. 943, Aug. 2019, doi: 10.3390/electronics8090943.

[21] A. Lehner, T. Strang, and C. Rico-García, "A reliable surveillance strategy for an autonomous rail collision avoidance system," in *Proc. ITS World Congr.*, 2008, p. 1–11.

[22] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *Proc. SIGGRAPH Course*, 2008, pp. 1–16.

[23] M. Grewal and A. P. Andrews, "Linear optimal filters and predictors," in *Kalman Filtering: Theory and Practice Using MATLAB*, 4th ed. New York, NY, USA: Wiley, 2015, ch. 5, sec. 5.4, p. 200.

[24] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Proc. Veh. Technol. Conf. IEEE 55th Veh. Technol. Conf. (VTC Spring)*, May 2002, pp. 317–321, doi: 10.1109/VTC.2002.1002718.

[25] G. Li, Y. Yang, T. Zhang, X. Qu, D. Cao, B. Cheng, and K. Li, "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 122, Jan. 2021, Art. no. 102820, doi: 10.1016/j.trc.2020.102820.

[26] B. Tomar and N. Kumar, "Fuzzy logic based braking system for Metro Train," in *Proc. Int. Conf. Intell. Technol. (CONIT)*, Jun. 2021, pp. 1–4, doi: 10.1109/CONIT51480.2021.9498360.

[27] L. Gonzalez, J. A. Matute-Peaspan, J. P. Rastelli, and I. Calvo, "Longitudinal collision avoidance based on model predictive controllers and fuzzy inference systems," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6, doi: 10.1109/ITSC45102.2020.9294584.

[28] N. Febriany, "Application of the fuzzy Mamdani method in determining the nutritional status and daily calorie needs of toddlers using MATLAB software," M.S. thesis, Dept. Mathematics Education, Indonesian Educ. Univ., Bandung, Indonesia, 2016.

[29] H. Chen, L. Liu, Y. Zhang, and Y. Tian, "Adaptive speed control of autonomous vehicle under changing operation conditions," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 3522–3526, doi: 10.23919/ChiCC.2017.8027903.

[30] S. Dani, D. Sonawane, D. Ingole, and S. Patil, "Performance evaluation of PID, LQR and MPC for DC motor speed control," in *Proc. 2nd Int. Conf. Converg. Technol. (I2CT)*, Apr. 2017, pp. 348–354, doi: 10.1109/I2CT.2017.8226149.

[31] D. L. Luu, C. Lupu, L. S. Ismail, and H. Alshareefi, "Spacing control of cooperative adaptive cruise control vehicle platoon," in *Proc. IEEE Int. Conf. Autom., Quality Test., Robot. (AQTR)*, May 2020, pp. 1–6, doi: 10.1109/AQTR49680.2020.9129997.

[32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

KHANSA SALSABILA SUHAIMI received the bachelor's degree in electrical engineering from Institut Teknologi Sumatera, Lampung. She is currently pursuing the master's degree in electrical engineering with Institut Teknologi Bandung. Her research interest includes the control of autonomous driving which is also a core topic of the master's research project.

ABDURRAAFI' SYAUQY is currently pursuing the bachelor's degree in electrical engineering with Institut Teknologi Bandung. His final project is about the autonomous control of a tram. His research interests include the IoT, robotics, and automation.

MOHAMMAD SALMAN SUBKI is currently pursuing the bachelor's degree in electrical engineering with Institut Teknologi Bandung. He is also working on his final project about autonomous control of a tram.

BAMBANG RIYANTO TRILAKSONO (Member, IEEE) received the degree in electrical engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia, and the master's and Ph.D. degrees in electrical engineering from Waseda University, Tokyo, Japan. He is currently a Professor with the Control and Computer System Research Group, School of Electrical Engineering and Informatics, ITB. His research interests include control systems, robotics, and artificial intelligence.

ARIEF SYAICHU ROHMAN (Member, IEEE) was born in Malang, East Java, Indonesia. He received the bachelor's (Ir.) degree in electrical engineering from Institut Teknologi Bandung (ITB), Indonesia, in 1990, the M.Eng.Sc. degree in systems and control from the University of New South Wales, Sydney, Australia, in 1998, and the Ph.D. degree in systems and control from The University of Newcastle, Newcastle, Australia, in 2005. He was with the Research and Development Division, PT IPTN, the Indonesian aircraft industry, from 1990 to 1992. He joined the Control and Computer System Research Group, School of Electrical Engineering and Informatics, ITB, in 1992. He is currently an Associate Professor. His research interests include anti-windup systems, model predictive control, sliding-mode control, engineering education, and control applications in distillation columns, electric vehicles, and auto-drive systems.

YULYAN WAHYU HADI (Member, IEEE) was born in Tangerang, Indonesia, in June 1991. He received the B.Sc. and M.Sc. degrees in electrical engineering from Institut Teknologi Bandung, Indonesia, in 2012 and 2015, respectively. He is currently a Lecturer with the School of Electrical Engineering and Informatics, Institut Teknologi Bandung. His research interests include control systems, autonomous systems, and robotics.

HANDOKO SUPENO received the bachelor's degree in informatics from Pasundan University and the master's degree in electrical engineering from Institut Teknologi Bandung (ITB), Bandung, Indonesia, where he is currently pursuing the Ph.D. degree in electrical engineering and informatics. His research interests include deep reinforcement learning and autonomous driving.

DHIMAS BINTANG KUSUMAWARDHANA received the bachelor's degree in electrical engineering from Institut Teknologi Bandung, where he is currently pursuing the master's degree in electrical engineering. He was a member of TFRIC-19, in which his team developed an AI-based tool to accelerate doctors in diagnosing Covid-19 patients through medical imaging. He is a Project Manager on autonomous vehicles with Riset.ai.

DEWI NALA HUSNA received the bachelor's degree in electrical engineering from Institut Teknologi Bandung, Bandung, Indonesia, in 2017. Since 2018, she has been a Junior Engineer with PT INKA, Madiun, Indonesia, and engaged in train control and monitoring system development under the Transportation System Development Unit.