IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 June 2023, accepted 5 July 2023, date of publication 17 July 2023, date of current version 26 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296317

RESEARCH ARTICLE

Adaptive Cruise Control Based on Real-Time DDS Middleware

BASEM ALMADANI, NAJOOD ALSHAMMARI, AND ANAS AL-ROUBAIEY
Department of Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
Corresponding author: Najood Alshammari (g202009140@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals.

ABSTRACT This paper discusses the Adaptive Cruise Control (ACC) system, which is a solution that allows drivers to minimize the amount of time spent driving. It supports four different driving modes and regulates the acceleration and deceleration of the car in order to maintain a fixed speed or avoid a collision with another vehicle. Real-Time Publish-Subscribe (RTPS) middleware has emerged as one of the most efficient and practical options for the real solutions to the difficulties listed above. This paper proposes to develop a real-time system for integrating the various components of ACC, such as Information Cluster, Radar, Brake Switches, Cruise Switches, ACC Controller, Engine/Throttle Controller, Brake Controller, Brake Actuators, Speed Sensors, and Back Brake Lights. The exchange of data is through a RTPS Data Distribution Service (DDS) middleware. The design of the publish/subscribe model was explained in detail in this paper along with the proper Quality of Service (QoS) policies suggested to govern the behavior of the model.

INDEX TERMS Adaptive cruise control, DDS middleware, leading vehicle, quality of service policies, ACC controller, engine controller, brake controller.

I. INTRODUCTION

As traffic on highways has steadily risen, greater congestion has resulted, resulting in vastly increased traffic oscillations and increased accidents [1]. Furthermore, according to the National Highway Traffic Safety Administration (NHTSA), an estimated 31,720 individuals have died in motor vehicle traffic accidents between January and September 2021. The prediction represents the greatest number of fatalities in the first nine months of any year since 2006 and the largest percentage rise in the Fatality Analysis Reporting System's history [2], [3]. The need for Adaptive Cruise Control (ACC) arises. ACC will not only fulfil safety requirements, it will fulfil driver's comfort and provide ease to the passengers even in perilous traffic maneuvers, as it may alleviate driver stress by automatically adjusting the vehicle's speed and maintaining a predetermined minimum distance to the preceding vehicle [4]. As a result, the driver is more comfortable and can focus better on the traffic. ACC is an extension of the conventional cruise control system that is widely used in many commercial vehicles. ACC system main objective is to replace the driver in terms of operating the throttle pedal and the brake pedal in order to control the vehicle's speed, acceleration, deceleration and braking, in many different situations, whether it is in traffic jams, in a cut-in situation, cut-out situation or in a situation where a vehicle ahead brakes to avoid an accident [2]. The ACC system controls the vehicle to maintain a safe desired distance in relation to the leading vehicle.

The ACC system can be based on either an end-to-end controller or a hierarchical controller. The end-to- end controller is one main controller where the input is taken from sensors processed in the controller producing the desired torque [3]. On the other hand, the hierarchical structure is built on an upper controller and a lower controller. Each controller plays a significant role in the control structure of ACC. The lower controller is a control layer while the upper control works as a decision layer. When the road is clear and there are not any leading vehicles ahead of the vehicle with ACC, the vehicle would maintain a speed that was initially pre-set by the driver and that is called a speed control mode. On the other hand, if there is a leading vehicle ahead of the vehicle with ACC then the vehicle with ACC is in a distance control mode. In the case of the distance control mode, the upper controller

The associate editor coordinating the review of this manuscript and approving it for publication was Jie Gao.

would contain a longitudinal car-following model along with an optimization algorithm.

When it comes to the acceleration, an optimal control instruction and command will be calculated by the upper controller, depending on the motion state of both vehicles, the leading vehicle and the vehicle with ACC. On the other hand, the switching logic and the braking force are contained within the lower controller. The switching logic usually contains the vehicle's drive control strategy for the Electrical Vehicle's (EV) motor in addition to a brake control strategy. The braking force is a distribution strategy as it includes the distribution strategy of braking force for the vehicle's axles (front and rear) [4]. In other words, whenever the driver activates the ACC, the upper controller starts taking information from the equipped sensors and calculates the optimal trajectory, acceleration, and speed. After that, the lower controller takes action and begins to execute the trajectory calculated by the upper controller and sending low-level instructions to the vehicle control interface. As mentioned earlier in this section, the controllers hold major significance in ACC and that drawn researchers' attention to the controllers' design as there are many different algorithms that have been proposed and employed in controllers' design. Many of these algorithms will be in the literature review of this paper such as fuzzy control, Model Predictive Control (MPC), deep reinforcement learning, Robot Operating System 2 (ROS2), and Action Dependent Heuristic Dynamic Programming (ADHDP). Furthermore, the hierarchical structure is common in all these papers. However, there are not any researches that have proposed a messaging oriented real-time middleware for adaptive cruise control to integrate the subsystems and the different controllers of the ACC. Moreover, real-time middleware has been implemented successfully in different areas increasing the reliability and flexibility of the systems integrated. Targeting the above issue, the objective and contribution of this study is to propose a real-time middleware platform for ACC in vehicles and introduce the publish/subscribe model to this area of research. By introducing a real-time middleware to vehicles as a whole, the automation of EVs can be more feasible in term of integrating all the heterogenous systems that exist in an EV. The overall system this paper proposes contains the following components that will all be integrating using the DDS; radar, instrument cluster, brake switch, cruise switches, engine controller, brake controller, ACC controller, brake actuators, speed sensors, and back brake lights.

More specifically, the main contribution of this paper can be concluded as:

1- Adapt a messaging oriented real-time middleware with a real-time publish-subscribe model to the ACC system.

2- Adapt the settings of the Quality-of-Service (QoS) policies to the ACC system.

3- Evaluate the performance of the proposed solution via TOSSIM emulator.

After stating the contribution, the merits of this work are as follows:

TABLE 1. Abbreviation list.

Abbreviation	Meaning
ACC	Adaptive Cruise Control
DDS	Distributed Data Services
QoS	Quality of Services
RTPS	Real-time Publish-subscribe
PDR	Packet-Delivery-Ratio
EED	End-to-End Delay
IPI	Inter-Packet Interval
PID	Proportional Integral Derivative
MPC	Model Predictive Control

1- Interoperability: Ensure that applications based on DDS implementations from various vendors may communicate with one another.

2- QoS Policies: DDS has a rich set of QoS policies that enable users to control the behavior of communications. These QoS rules may be used alone or in combination to influence a range of aspects of communication, including reliability, performance, data persistence, and the amount of system resources used. These QoS policies differentiate DDS from other communication middleware technologies. DDS is a superior choice for communications in a wide range of sectors and architectures due to the breadth and depth of configuration options provided by these QoS policies.

3-The architecture utilizes RTPS protocol which is a multicast. where a single message sent by a sender may be received by a large number of recipients. Reliability is an important goal of RTPs, which is aimed to ensure that the communication method is dependable.

The present article is structured in the following manner: Section II pertains to Related Work and is bifurcated into two parts. The first part provides a review of articles that center around ACC, whereas the second part is dedicated to studies based on DDS. In Section III, the proposed solution, which entails the amalgamation of the controllers and other constituents of the ACC system. Additionally, expound on the design of the real-time publish/subscribe model of the ACC, which encompasses the Quality of Services policies. In section IV, the mathematical model of an ACC system is explicated. Moreover, Section V delves into the design models and thoroughly examines all conceivable scenarios. Section VI comprises design diagrams of the domain in a publish/subscribe manner. The final segment of this document comprises three sections, namely Section VII Experimental Work and simulation results, Section VIII which outlines the directions for future work, and Section IX which provides the concluding remarks.

Table 1 shows a list of abbreviations that have been used more than once in this paper.

II. RELATED WORK

In this section, we describe the published studies that are most related to our work. To the best of our knowledge, this study is the first trail to leverage the DDS middleware capabilities in controlling and improving ACC systems. Therefore, we classify the related work into two sub-sections: ACC and DDS middleware.

A. ADAPTIVE CRUISE CONTROL BASED STUDIES

Yang et al. [5] proposed an optimal design for an ACC system based on MPC and Active disturbance rejection control (ADRC) compensatory control. The MPC approach was implemented as the top controller of the hierarchical design in order to improve safety, tracking capabilities, fuel economy, and ride enjoyment. MPC may provide a suitable command to the lower controller during each sample time period based on all available information. However, if the prediction model is inaccurate, it is hard to obtain a suitable response; as a result, a predictive acceleration estimator based on the least square approach was developed. Utilizing this Acceleration Predictive estimation (APE) approach in the MPC framework when the front target vehicle accelerates or decelerates can increase control accuracy. Once the desired acceleration has been attained, the throttle or brake actuator is modulated to monitor the planned acceleration. As a result, the lower-level controller made use of acceleration feedback and compensatory control techniques such as ADRC and vehicle dynamic model (VDM). When the host vehicle is subjected to internal or external disturbances, this enables the host vehicle to follow the front target vehicle accurately and securely. The suggested ACC–APE–ADRC controller and the ACC–APE controller were validated using road trials. Three distinct experimental settings were examined.

Zhang and Zhuan [7] examined the control strategy for an ACC system on a Battery Electric Vehicle (BEV) during the car-following method, with the primary focus being on the incorporation of regenerative braking in a BEV during the car-following operation. The ACC system is structured in a hierarchical manner. Additionally, the structure is controlled by an upper and lower controller. The higher controller improves various objectives, including safety, monitoring, comfort, and energy consumption, by utilizing the MPC approach. Energy is recovered in the lower controller during braking. They evaluated their suggested ACC technique using simulation, and they were able to provide safe tracking for the leading car, while also greatly improving comfort and energy efficiency.

Wei et al. [8] presented double deep Q-networks as a deep reinforcement learning technique for building an end-to-end vision-based ACC system. To create a simulation environment resembling a highway situation, a gaming engine was used that included both real-world automotive models and feature data for learning and testing. A reward mechanism was integrated into the reinforcement learning model for both Internal Combustion Engine (ICE) vehicles and EVs to conduct ACC. Additionally, the gap statistics and total energy consumption for a variety of vehicle types were analyzed in order to determine the link between incentive systems and engine parameters. Compared to existing radar-based ACC systems or human-in-the-loop simulations, their proposed vision-based ACC system may give a more gap-controlled or shorter velocity route, depending on the optimization strategy used. The suggested technology operates in real-time and is capable of adapting to the changing speed trajectories of the vehicle ahead.

Chen et al. [9] proposed a hybrid MPC controller based on a simplified dual neural network (SDNN) and a proportional integral derivative (PID) focusing on a single neuron (SN) for ACC control of autonomous electric vehicles, with the objective of balancing comfort, tracking, safety, and energy economy while taking spatiotemporal constraints between the leading and following vehicles into account. Following and cruising modes were specified and implemented using the MPC algorithm based on SDNN and the PID controller based on SN, respectively. Typically, conventional ACC systems neglect lateral dynamics control; but, in this case, it was considered, resulting in the ACC system being able to work on the curved road. The braking approach was also demonstrated to be successful in simulations.

Reke et al. [10] presented a ROS2-based architecture for a self-driving vehicle in this study. Self-driving cars must make real-time judgments based on sensory information, necessitating both high reliability and a high level of functional safety. Additionally, this article explained how ROS, in particular, may not always meet these standards. On the other hand, the successor ROS2 has revolutionized the landscape, since it has been offered as a solution for autonomous driving. Additionally, the current existing ROS-based robotic software has been shown to be insufficient for safety- critical applications such as self-driving cars. This paper presented an architecture for a self-driving vehicle based on ROS2 that enables safe and predictable real-time behavior while keeping ROS's distributed design and standardized message formats. Their initial testing using an automated real-world passenger vehicle at both high and low speeds suggest that their method is viable for autonomous driving under the requisite real-time conditions.

Adiththan and Ravindran [11] discussed methods for evaluating the QoS capabilities of networked embedded software systems. Given a target system S, its internal algorithmic processes and component subsystems can be adjusted to enable it to cope efficiently with hostile environment conditions that may emerge. The research gave a case study of cruise control systems in automobiles to evaluate their assessment and reconfiguration methodologies. They have also demonstrated the problems inherent in properly coordinating a large number of vehicles in the face of adverse environmental conditions (such as slipperiness, altitude, signaling message loss, and so on), while striving to meet per-vehicle QoS standards. Their model-based engineering methodologies enable them to assess several networked embedded systems with identical functional objectives against certain non- functional attributes. It enables the planning of bigger network installations with predictable behaviour using a system-of-systems composition.

Reschka et al. [12] suggested that the critical factors to consider are the work required to test a software update and the vast diversity of possible settings accessible in distinct automotive models. They examined the needs for software that enables such updates, monitors for new software versions, and offers reconfiguration procedures for individual control units and dispersed groups of control units in this study. They studied the total vehicular environment, including space, electric power, processing power, and cost limits, as well as four example road car automation systems and a complete x-by-wire target vehicle for performing these applications, in order to identify the requirements. The examination of these three distinct demand sources elucidates important middleware features and needs, particularly in terms of runtime and update lengths. The criteria include capability for updating with built-in authentication, application exchange on single control units, and capability loss and take-to-control unit relocation.

Bai et al. [13] argues that the rapid use of self-driving autos has spawned a new study area in vehicle management system real-time scheduling. After examining classic open-loop scheduling techniques used in autos and reactive real-time scheduling strategies advocated for general Distributed real-time embedded (DRE) technologies, they decided that a novel real-time scheduling solution should be developed based on a unique characteristic of driving management. They proposed AutoE2E, a two-tier software system for automobile operating systems, in that work. It managed to overcome the shortcomings of existing solutions by leveraging a second- tier controller to dynamically reduce runtime within reasonable parameters in order to reclaim efficient CPU usage control for End-to-End (E2E) real-time promises. AutoE2E has been assessed on a physical test platform using miniature automobiles as well as in a larger-scale simulation. Their final results reveal that AutoE2E beats a cutting-edge system solely based on rate adaptation by 35.4 percent in terms of target miss ratio over the same time period, while also minimizing route tracking error.

Chen et al. [14] proposed a model-free ADHDP-based Eco- ACC approach for EVs operating in a car-following scenario in order to improve vehicle safety, battery life, ride comfort, and energy efficiency. The simulated results indicate that the Eco-ACC technique may help keep the EV on track with the automobile ahead of it, greatly minimize battery capacity loss, and somewhat cut energy consumption. Additionally, the proposed Eco-ACC is model-independent, real-time, and robust to a variety of car-following scenarios.

Zhao et al. [15] successfully developed a novel spacing control law that improves fuel efficiency and ride comfort, resulting in increased driving safety. This control rule is intended to be used as the upper-level controller in an ACC system based on model predictive control and a real-time weight tuning technique for calculating the optimal desired acceleration. During the transitional maneuvers (TMs), the proposed spacing control law might simultaneously achieve control objectives such as high fuel efficiency and riding comfort. To improve fuel efficiency across TMs, the proposed spacing control regulation based on MPC employs a new compromise control strategy that takes into account creating a safe inter-vehicle gap with zero relative velocity behind a vehicle ahead. Due to the complexity of the inter- vehicle statements used in TMs, a real-time weight tuning approach is also developed and applied to the spacing control rule, resulting in an optimal control command. So that the host vehicle may smoothly generate a safe inter-vehicle distance while boosting ride happiness and fuel economy, but previous research does not account for vehicle safety, energy consumption, passenger comfort, and time-varying problems.

Luu et al. [16] used Matlab/Simulink to construct a simulation engine for platoons of ACC autos. Because the driver in a platoon establishes the velocity reference, it has been governed using the usual CC method. A control legislation for ACC vehicles has been created in accordance with the constant time gap (CTG) spacing strategy. This is to pursue the leader vehicle at a preferred range that is dependent on the vehicle, the exact distance being determined using a sensor such as radar range and range rate sensors, in order to accomplish the mission of maintaining the desired spacing and relative velocity of the vehicle ahead that does not require the use of current traffic infrastructure.

Chen et al. [17] provided an intelligent fuzzy control system for combining throttle and brake control in order to perform ACC and Stop & Go operations inside a single control framework. A simulation module for automobile intelligent cruise control is developed using MATLAB/Simulink and is based on vehicle dynamics modelling, with test data from a 1.6 L car equipped with four-speed automatic gearboxes serving as an example. The results of the testing cases demonstrate that the proposed fuzzy control approach is effective and feasible, capable of performing not only high- velocity ACC and low-velocity Stop & Go control duties, but also normal cruise control functions.

Patel et al. [18] discussed a philosophical and model-based technique for changing the behavior of the ACC system in autonomous automobiles. The main notion is to employ adaptive systems inside safety monitoring in the event of a system fault and to automatically trigger a new safe reconfiguration to counteract the dangerous configuration. Numerous situations and environmental conditions were used to perform a complete case study of the ACC system's adaptive activity during runtime. Matlab/Simulink was used to develop the modelling environment and to determine the feasibility of the ACC system requirements and a related system component.

The primary objective of this segment of the literature review is to elucidate and draw attention to the recent trends in ACC research. Notably, these studies share a commonality of the hierarchical structure of the ACC system, irrespective of their respective foci. However, it is worth noting that this paper deviates from this trend by eliminating the hierarchical structure. Moreover, certain investigations have endeavored to enhance the simulation through the integration of regenerative braking [7], creating simulation environment resembling a highway situation [8], enabling the ACC system to work on curved roads [9], developing a modelling environment for the adaptive behavior of the ACC system during runtime, utilizing a variety of scenarios and environmental factors [18], and not only frameworks that are capable of performing high-velocity ACC but also low-velocity Stop & Go control duties [17]. Several other studies have demonstrated an interest in strategies for controlling spacing [15], [16]. In the realm of this particular field of interest, only a limited number of studies have incorporated diverse forms of middleware, including AutoE2E [13] and ROS2 [10].Other scholars have conducted a thorough analysis of the necessary conditions for a middleware that enables the streamlining of updates, verification of new software versions, and incorporation of reconfiguration mechanisms for both singular control units and distributed control unit collections [12]. Furthermore, QoS capabilities of networked embedded software systems [11]. The aforementioned gateways serve as useful entry points to the notion of basing ACC on real-time DDS middleware, while leveraging DDS QoS policies to regulate DDS behavior within said system.

B. DATA DISTRIBUTED SERVICES MIDDLEWARE BASED STUDIES

Al-Madani and Ali [19] suggested that as a result of inefficient peak-load management, concentrated power generation, restricted information flow, and inadequate distribution assistance, today's electric grid faces several difficulties. Several groups are working on Smart grid as a result of these restrictions. The proliferation of heterogeneous devices in a smart grid only serves to exacerbate the problem of increased complexity and inefficiency. Middleware is regarded the best solution for dealing with the heterogeneity of various devices and ensuring interoperability. Data Distribution Service (DDS) middleware delivers high levels of dependability and efficiency by addressing additional performance indicators and numerous QoS criteria, particularly in real-time and mission-critical operations, despite the fact that many middleware have been presented. An ANSI C12.19-based DDS implementation has been studied for transmission and consumption. To facilitate connection and carry out experimental investigation for the assessment of interoperability and other performance metrics, data structures are obtained for topics formation over RTI Connext in order to demonstrate that DDS is an ideal option for Smart grid data interoperability and high reliability.

Llorens-Carrodeguas et al. [20], proposed a novel technique to distributing software-defined network (SDN) domains using DDS. This study argues that paradigm change in communication networks has occurred because of software-defined network (SDN) technology, which allows networks to be programmed by using centralized or decentralized controllers. New verticals, such as Industry 4.0, cooperative sensing, and virtual reality, have evolved as a result of the evolution of the industry. The usage of scattered domains is required to increase network scalability and durability in these sectors. DDS was used to distribute SDN subdomains in this article. The DDS facilitates network information interchange, controller synchronization, and self-discovery. Furthermore, it enhances the control plane's resilience, which is critical for 5G networks. Using SDN controllers, this paper has built a testbed to evaluate the DDS's efficacy, and then deployed it across several parts of Spain to see if it worked as expected. The latency and overhead of controller-to-controller communication was examined.

Madden and Ghalaab [21] determined that a distributed simulation architecture was required by NASA Langley Research Center in order to facilitate collaboration across real, virtual, and constructive nodes from across LAN, as well as extensibility to other NASA Facilities and external parties. In one configuration, the GovCloud cloud computing service was combined with the DDS middleware. Data was sent between the nodes through DDS. There is a potential approach to allow other Units and partners to access the distributed simulation over an established secure network, avoiding the requirement to negotiate individual interconnection security agreements. Piloted and unpiloted aircraft exchanged Auto-Dependent Surveillance Broadcast (ADS-B) signals using the prototype design. In order to evaluate the development in terms of upfront investment to enhance a node's design, connectivity and interoperability of nodes, performance, integration and security, various node configurations were run and reviewed.

El-Ferik et al. [22], discussed Unmanned Aerial Vehicle System (UAV) formation control in this work. DDS middleware is used to guide the multi-UAV agents through the L1 controller. The L1 adaptive controller is utilized to stabilize the general motion equations of each UAV, while the potential field approach is employed to formalize the following UAVs around the leader. DDS publisher/subscriber middleware is used to exchange data between both the leader and the followers. An adaption of the L1 controller has resulted in a high level of performance. The L1 controller's robustness was tested using Matlab Simulation. The analysis and stability of the framework of UAVs were supplied by the Lyapunov approach.

Almadani et al. [23] proposed a real-time Automatic Vehicle Location (AVL) and monitoring system for traffic control of pilgrims traveling towards the Saudi Arabian city of Makkah in this study. The system is built on the DDS. Many-to-many communication paradigm ideal for use in enormous traffic control applications is implemented using Real-Time Publish/Subscribe (RTPS) protocol, which is implemented by DDS-based middleware. This middleware technique allows us to find and monitor a large number of mobile cars while also identifying all passengers in real-time who are traveling to Mecca to complete the annual Hajj ritual. Various performance metrics are investigated via WLAN utilizing DDS in order to verify the validity of the proposed framework. The results demonstrate that DDS-based middleware is capable of meeting real-time requirements in a large-scale AVL setting.

Cantelli-Forti et al. [24] suggest that increasing efforts are needed to ensure the safety of essential infrastructure. Considering both cyber- and physical attack vectors when protecting against inexpensive unmanned vehicles, especially when it comes to Internet of Things (IoT) or hardware threats. The implementation of more robust and precise technologies is needed in these situations. Requirement analysis and demonstration system performance are discussed in this study. Anti-cyber-attack technologies and methods for detecting foreign physical items are examined. Examples and trials of the SCOUT Multitech Security system for interconnected space control ground stations are used in the study. The DDS ensures interoperability of the SCOUT systems. DDS standard is implemented in Vortex OpenSplice. All actors in the SCOUT system subscribe to the DDS subject in order to publish their data. Each subsystem has an agent installed that is able to recognize new data and respond appropriately. The data is analyzed, transferred to the central DDS system, and sent to the receiving system via server message block protocol (SMB). DDS subscribers may see the new instance and pick up a newly transmitted file from their system.

Ibarra-Junquera et al. [25] proposed a holistic framework for the purpose of bridging the gap between generic designs and physical implementations of industrial bioprocesses by proposing their framework for flexible and scalable automation. The framework is based on a holistic approach, in which a system and its attributes are studied comprehensively, as a whole, rather than as the sum of its components. To facilitate rapid design and implementation in the bioprocess industry, a software design pattern based on components, container technology, microservices concepts, and the publish/subscribe paradigm is defined. This pattern defines a set of components with offer and request services that can be easily connected via the plug-and-play technique and interconnected with a middleware based on the publish/subscribe pattern such as DDS. To show the suggested framework's applicability, two processes in the fruit juice drinks business were devised and executed at the *Punta Delicia S.A. de C.V.* juice manufacturing facility in Colima, Mexico. A method for manufacturing soursop soda was presented, with a fuzzy controller used to maintain a constant pasteurizer output flow (UHT) and an automated storage tank selection and filling procedure using actuated valves to route the fluid to the appropriate tank when the process is started. The results indicated that the platform provided a high-fidelity environment for designing, analyzing, and testing the flow of cyber information and its effect on physical operation in a beverage processing plant with a high demand for adaptability, flexibility, and efficiency of its processes, as demonstrated experimentally in a real production process for the production of 860 L of soursop. Each development was addressed individually until the procedures were optimized, resulting in cost savings associated with development and final application. Finally, in each of the case studies provided, both general and specific needs were addressed, demonstrating the framework's adaptability, scalability, and resilience.

Upon careful examination of the literature, it is our scholarly opinion that contemporary publications pertaining to ACC have yet to address the implementation of a message-oriented middleware that operates in real-time, nor have they presented an RTPS model that accurately represents any existing ACC system. Furthermore, an in-depth analysis of QoS policies within an ACC system has not been conducted. Conversely, the current scholarly articles that center around the exposition of DDS have yet to address any ACC systems. Our research endeavors to address a significant gap in the field by presenting a middleware technology for integrating an ACC system. Additionally, we have formulated an ACC system in an RTPS model, which offers exceptional flexibility and scalability. Furthermore, our research paper underscores the suitable QoS protocols for an ACC system and modifies the configurations of said QoS protocols to suitably align with the ACC system of a vehicle.

III. PROPOSED SOLUTION

ACC demands a high-speed information flow and exchange as it is considered mission critical, and any latency in an information arrival can lead to a deadly accident [25]. ACC system consists of radars and speed sensors, an engine controller, and ACC controller, a brake controller, brake actuators, brake switches, cruise switches, an instrument cluster, and back brake lights. And all these inputs and information are flowing in real-time and has to be analyzed immediately in order to make decisions.

Moreover, you have to sense what is happening in the environment then based on the information you have obtained make a decision. The proposed solution to overcome these issues and to meet the demands of the adaptive cruise control system is to introduce DDS real-time middleware as it aims to enable high-performance, real-time, interoperable, scalable data exchanges using the publish/subscribe pattern. DDS in this proposed solution integrates the components of ACC together in a way that gives different component access to all the inputs at the same time, which is real-time. Moreover, a DDS bus will integrate the information inside the ACC system, where the components of the system can be looking at the same information at the same time to make their own decision. Furthermore, introducing DDS to ACC in vehicles makes it much easier to add sensors and different instruments to the vehicle without having to reconfigure the flow of data in the system, it also would make it easier to add different features to the vehicle that use the same data obtained from the sensors without having to modify any hard coding, by simply adding the new feature to the DDS data-bus.

As a result of this novel addition, the instrument cluster, the ACC controller, the engine controller, the brake controller and the brake actuators can all have the chance to read the inputs

TABLE 2. The publisher/subscriber DDS middleware QoS policies.

QoS Policy	Value
Durability	TRANSIENT
Presentation	Ordered_access
Ownership	SHARED
Liveliness	MANUAL_BY_PARTICIPANT
Reliability	BEST_EFFORT
Transport_Priority	Transport_priority
Destination_Order	BY_SOURCE_TIMESTAMP

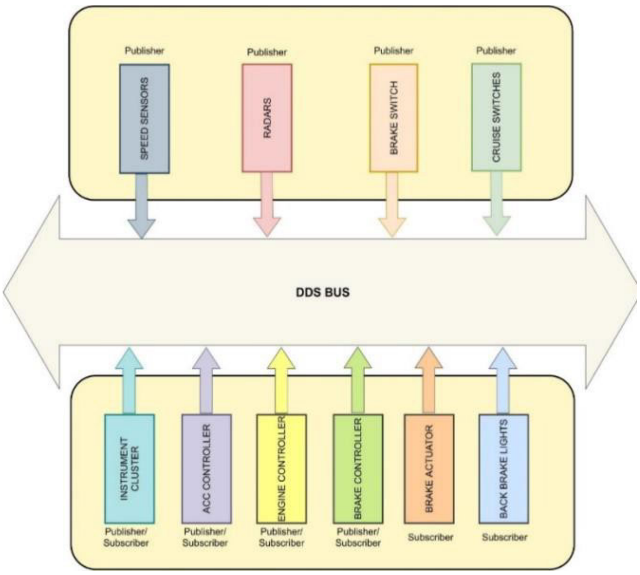

FIGURE 1. The DDS BUS containing all the components of the system.

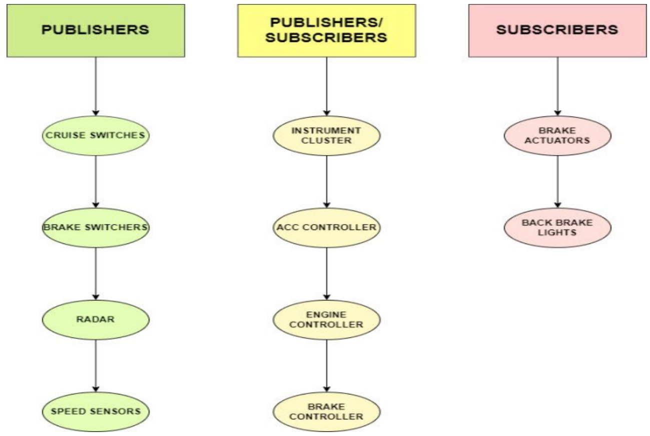

FIGURE 2. The publisher and subscribers of the system.

of the driver through the cruise switches and brake switches, and the inputs of the radar and the inputs of the speed sensors, at the same time.

The data that will be exchanged in the system is from now in this paper referred to as a Topic. In DDS, each topic contains first a name which must be unique, a type that has the definition of what it carries within, QoS policies that governs the behavior of the topics within the domain. Last but not least, the type of data is generic across all domains [26], [27]. In ACC, different topics exist and are listed in Table 3. Along with what is called Data Writer and Data Reader. The Data Writer is the creator of a topic that will publish it for subscribers to receive, the Data Reader on the other side is the receiver on the topics published and created by the Data Writer. Figure 2 shows the general relationship between the publisher, subscriber, data writer, data reader, topic, and QoS policies.

Lastly, there are important requirements for the ACC system that our proposed solution address and are listed below;

1- Interoperability between components by different vendors → DDS is a middleware standard that is a solution to integrate heterogeneous platforms.
2- Strict order of data arriving at each destination → Presentation QoS and Destination order QoS gives the user the liberty to custom the order of data arriving.
3- Ability to fetch back a value (i.e. resuming to initial set speed) → Durability QoS
4- Ability to allow redundant sensors to participate for safety purposes → Ownership QoS

A. DATA DISTRIBUTED SERVICES (DDS)

The OMG, which main objective is to standardize technology in a way that hundreds of different sectors can benefit from it and to integrate a wide range of technologies, have introduced in recent years the DDS [28].

The DDS is an application programming interface standard that is used to link data centrically. Moreover, DDS is a middleware platform that is considered a specific type of real-time publish/subscribe platform that utilizes a message passing middleware [29].

A middleware is an interoperability software that acts as a bridge connecting an application program and a network, concealing variations or incompatibilities in network transport protocols, physical infrastructure, system software, distributed databases, and connectionless calls. Furthermore, in heterogeneous computing middleware can be seen as a way to mask any complexity of the underlying heterogenous environment [31]. Last but not least, DDS consists of API, protocol and presentation. As when information is being exchanged between different applications the middleware ensures that information is in a context that can be understood by the receiving end regardless of the operating system that is being used in each application [30]. Furthermore, the presentation aspect of the middleware includes topics, types, filtering entities. Lastly, the protocol includes the reliability, discovery and QoS policies which will be discussed in details at the end of this section. A message-oriented middleware communicates in a ‘single shot’ rather than having to request and wait in order to communicate. Moreover, this specific type of middleware has a preference for applications that

TABLE 3. Topics of ACC model in RTPS model.

Topic name	Topic Description	Data Type	Data writer	Publisher	Data reader	Subscriber
Brake_Switch	Signal from the break switch that deactivates ACC and puts it intostandby mode.	Generic	Brake switch		1- Instrument cluster. 2- Engine controller 3- ACC controller	
ON	Puts the systems in standby mode.	Boolean	On button	Cruise switches	Instrument cluster	
OFF	Turns off theACC system.	Boolean	Off button	Cruise switches	Instrument cluster	
TimeGap+	Increment thetime gap by 1.	Generic	Time gap + button	Cruise switches	Instrument cluster	
TimeGap-	Decrement thetime gap by 1.	Generic	Time gap – button	Cruise switches	Instrument cluster	
Resume	To reset the speed back to theinitially set speed.	Generic	Resume button	Cruise switches	Instrument cluster	
-Speed	To decelerate	Generic	-speed button	Cruise switches	Instrument cluster	
Set+	Activate the ACC and set the speed/accelerates	Generic	Set+ button	Cruise switches	Instrument cluster	
CRZ_RQST	Cruise switch request wheneveractivated	Generic	Instrument cluster		ACC controller	
ACC_info_msg	ACC information messages from the controller to the driver	String	ACC controller		instrument cluster	
Distance	Between the ACC equippedvehicle and theleading vehicle	Floating-point number	Radar		ACC controller	
V_Lead	Speed of the leading vehicle	Floating-point number	Radar		ACC controller	
Target_v	Target speed tomaintain safe distance	Floating-point number	ACC controller		Engine controller	
BRK_DEC_RQST	Deceleration request from ACC controller	Generic	ACC controller		Brake controller	
V_speed	The ACC vehiclespeed	Floating-point number	Brake controller		1-ACC controller 2-Engine controller	
BRK_ACT_COM	The brake actuator command	Generic	Brake controller		Brake Actuator	
Wheel_Speed	Wheel speed bysensors	Floating-point number	Speed Sensors		Brake controller	
Light_COM	Back break lights	Generic	Brake controller		Back Lights	

require messages to be queued and held indefinitely [32]. Examples include workflow and communications apps.

As mentioned earlier in this section, middleware uses a publish/subscribe pattern, in a message-oriented middleware, it is possible for any program to publish data on the Internet, and interested programs should subscribe to a specific topic of interest. Subscribers look forward to receiving topics from publishers for a certain period of time and afterwards report an exception if the topic is not delivered within that period. As a result, the message passing method is extremely efficient in large distributed systems since neither the sender nor the recipient knows how many users are accessing the information. Additionally, neither party knows who originated the information.

Publish/subscribe is a communication model that works best when it comes to dynamic application that usually require reconfigurations frequently. Moreover, DDS ease adapting your solution based on the giving requirements and that leads to your solution being dynamic enough to adapt to a frequently changing environment. DDS has many different QoS specifications that can easily tailored and modified to fit a specific system and meet the requirements of any design. Furthermore, there are 22 different QoS policies that you can enable and give a certain value such as Latency Budget, Deadline, Resource Limits and History. Policies are set for the publisher and the subscriber and in order for the communication to be successful these policies need to be aligned and consistent with each other. Moreover, some policies can be seen as inter-related to each other and some on the other can be seen as conflicting. Last but not least, when you define every one of these things at the same time, at incredibly fast speeds, and in a rapidly evolving, challenging, and dynamic environments, the entire potential of DDS becomes apparent [27], [33].

B. DDS QUALITY OF SERIVE POLICIES FOR ADAPTIVE CRUISE CONTROL

QoS policies that are used by DDS can be used to improve the smoothness of transmission of data through the ACC system. In this section many QoS policies will be investigated in terms of how they can affect ACC and to show how these polices can be utilized to support ACC in terms of efficiency and time. Our system addresses the following QoS policies; Durability, Presentation, Ownership, Liveliness, Reliability, Transport_Priority, and Destination_Order. The QoS policies and the settings of each policy are summarized in Table 2.

1) DURABILITY

its main concern is whether the data should outlive their writing time, the writing time is the time the data writer takes writing the instance of a certain topic. DURABILILTY is important for the resume topic, as to reset the speed back to the initial set speed, the initial set speed need to be kept for a while until ACC is cancelled. Therefore, the value chosen for DURABILITY is TRANSIENT; for TRANSIENT the middleware is only required to keep the data in memory and not in permanent storage.

If the Set+ topic issues an instance, DURABILILTY policy is set to TRANSIENT immediately to hold in memory the very first issue for this topic. Moreover, the resume button is actually a Data Writer publishing through the cruise switches publisher, but in this only case, the resume button will be a subscriber to the Set+ topic and will be considered a late joiner that is interested in the very first issue.

2) PRESENTATION

it controls how the samples representing changes to data instances are presented to the subscribing application. Moreover, when it comes to ACC the order of changes in instances is extremely important, as a decision needs to be taken at the same exact time. In the case of the leading vehicle urgently braking, such event should be reported at the same time and to guarantee that, the order needs to be strictly monitored. Therefore, the value chosen for PRESENTATION is ordered_ access, as it gives the subscriber the ability to see changes in the same order, they have occurred in. Finally, access_scope is set to GROUP which makes changes that are made to instances by different DataWriters that share the same publisher are made available to subscribers on the same order they occur. This is important as the ACC controller needs to be updated by the Radars and Sensors continuously, and needs to receive changes and updates in order. Moreover, if the system uses a couple of redundant sensors for measuring the distance of the leading vehicle, then GROUP is the best fit to be able to monitor the changes made by all these sensors in the subscribing side.

3) OWNERSHIP

is a policy that can allow multiple DataWriter to update the same instance of the data. Choosing the value EXCLUSIVE gives the chance to specify one DataWriter that can modify an instance, while on the other hand, the value SHARED, indicates that there is no concept of an owner for the instance. Moreover, to turn off the ACC, the ''OFF'' topic, only two data writers are allowed to update the instances of the data which are the ''Off_Button'' data writer and the ''Brake_Switch'' data writer. In other words, ''Off_Button'' and ''Brake_Switch'' are the only two ways to turn off the ACC, therefore we are going to govern that by employing the OWNERSHIP policy. Last but not least, the OWNERSHIP_STRENGTH is usually a policy that applies the OWNERSHIP policy is SHARED, however, in this case it is unnecessary and that is due to the fact that the two data writers are equally important and should be of the same strength. Therefore, the default is zero and should remain this way.

4) LIVLINESS

is the mechanism used to determine if an entity is active or not. The value chosen for LIVELINESS is MAUAL_BY_PARTICIPANT, which declares all entities are active in the DomainParticipant, if at least one entity in the same domain is active. Moreover, if one entity in the ACC is active, all other entities must be active waiting for any update.

5) TRANSPORT_PRIORITY

is a policy that controls the importance of a topic or a topic's instances. Therefore, allowing the middleware to prioritize more important data. Moreover, this can be applied to the safe distance that needs to be left between the leading and following vehicles. it is more crucial if the distance is less than the safe distance than if the distance is over the safe distance.

6) DESTINATION_ORDER

its main concern is the logical order among changes that are made by the publisher entities to the same instance of data. The value chose for DESTINATION_ORDER is BY_SOURCE_TIMESTAMP, this indicates that data is ordered depending on the timestamp placed at the source. Moreover, this can guarantee a consistent final value of the data in all the subscribers.

IV. MATHEMATICAL MODEL

When the ACC vehicle is moving linearly on a horizontal plane, it is important to create an equilibrium of forces operating in the axis of the vehicle, this corresponds to the x coordinate according to norm. A behavior of this nature can be characterized as

$$\sum F_x = 0 \tag{1}$$

wherein Fx denotes forces operating in vector of the x-axis coordinate. The following is an example of an evolved form of the previous equation:

$$F_o = R_f + R_j + R_v. \tag{2}$$

F_o is the circumferential force to a point, R_f is the rolling resistance, R_j is the resistance of inertial forces while decelerating, and R_v is the resistance of air. It is required to manage the velocity of the ACC vehicle (v2) in order to obtain higher levels of quality in the job performed by the ACC vehicle. In this case, the car in front v 1 is used to regulate the speed of the vehicle ahead of it, including the distance between the two vehicles (d). Whenever the vehicles are at a certain distance from one another, the separation parameter is further adjusted and maintained constant. This is essential in case the distance between the leading vehicle and the ACC vehicle decreases while the ACC vehicle maintains a constant speed. If this occurs, it is important to activate the brakes of the ACC vehicle. As a result, the ACC vehicle's velocity is decreased to the same speed of vehicle 1, which is traveling in front of it. It is possible to prevent contact (crash) between the cars in this manner.

As long as all horizontal forces are equal on the vehicle, the vehicle's total braking force may be determined. As a result, the ACC vehicle's equation of motion begins to take shape as follow:

$$F_K = R_j - R_f + R_v \tag{3}$$

The braking force is denoted by the variable F_K. As the vehicle's speed lowers, the strength of the produced braking force is significantly greater than the rolling resistance, hence the air and rolling resistance are typically overlooked whenever the vehicle is braking (negligible). Because the braking force estimated in this study are so precise, we've included both resistances. When it comes to stopping force, linearity is defined as:

$$F = k \cdot t \tag{4}$$

K is the braking constant and t is the braking time. Because the vehicle's mass is not distributed evenly everywhere, the breaking force is transferred differentially on front and rear axles to maximize traction between both the tiers and the pavement. Breaking force is important to both axels of the vehicle in this article. In order to calculate the braking constant K, the total of the braking constants of the front and rear axles is used.

$$K = K_1 + K_2 \tag{5}$$

A vehicle's greatest deceleration is determined by the constant K, which is dependent on the speed of the car ahead of the ACC vehicle. Calculation of maximum deceleration is done as follows.

$$j_{max} = \frac{v_2^2 - v_1^2}{2.(D - d)} \tag{6}$$

D is the distance between vehicles at which ACC begins to work, and d is the minimum distance at which ACC ceases to function. v1 and v 2 velocities represent the velocity of the vehicle taking lead and the ACC equipped vehicle, respectively.

Assuming the vehicle has full grip, the braking force can only be as strong as the maximum braking force of the vehicle.

$$F_{max} = m \cdot j_{max} = m \cdot g \cdot \varphi_{max} \tag{7}$$

j_{max} is maximum vehicle deceleration, m is mass, g is gravity acceleration, and φ_{max} is maximum coefficient of adhesion. Forces that resist inertia are provided as an example.

$$R_j = m \cdot j \tag{8}$$

Vehicle deceleration is j and the vehicle's mass is m. The equation for calculating rolling resistance may be found here.

$$R_f = m \cdot g \cdot j \tag{9}$$

Here, f is the coefficient of roll-resistance coefficient. This is the empirical expression used to calculate the rolling resistance of radial tires.

$$f = f_0 + f_1 \cdot v + f_4 \cdot v^4 \tag{10}$$

The formula for calculating air resistance is followed:

$$R_v = \frac{1}{2} \cdot c_x \cdot A \cdot \rho \cdot v^2 \tag{11}$$

where c_x is the vehicle's drag coefficient, A is its frontal area, and ρ is the air density. The vehicle's slowing down is referred to as deceleration:

$$\mathrm{j} = \frac{F_K + R_f + R_v}{m} \tag{12}$$

$v = v_0 - \int Jdt$ is the formula for calculating both vehicles' velocities, where v 0 is the vehicle's beginning velocity? Calculating ACC vehicle travel distance is done using the following equation. To get the final distance, we use $S = S_0 - \int vdt$ as our starting point. A velocity integral must be calculated for both the ACC and preceding vehicles in order to establish their respective trip distances. ACC vehicles can also have variable or constant vehicle speeds in front of them. A distance separates them:

$$d = s_1 - s_2. \tag{13}$$

where the first vehicle's trip distance is s_1, and the ACC vehicle's journey distance is s_2.

V. DESIGN MODEL

Vehicle's speed is adjusted to keep a safe space for both the vehicle with ACC and the vehicle ahead of it in the very same lane. This section includes different subsections that describe different driving situations along with the interactions between the driver and the systems for example how to set the speed and how to disengage the system. The subsections are: Switching The System On, Setting A Speed, Following A Vehicle, Setting the Gap Distance, Disengaging The System, Overriding The System, Changing The Set Speed, Resuming The Set Speed, and Switching The System Off. Moreover, each subsection has operational steps that will be mentioned.

A. SWITCHING THE SYSTEM ON

Activate the On button which is a DataWriter that we have writing instances of the ON topic. Moreover. a CRZ_RQST instance will be published and whenever the ACC Controller receives the issue, ACC will be ready.

B. SETTING A SPEED

1. Increase the speed to the desired level.
2. Press the SET+ button. Set+ is a topic. Moreover, the memory stores the vehicle's speed and because of that the durability QoS policy is employed here, to make sure the system does not lose the initial speed after accelerating, at some point the driver might want to resume to the pre-set speed, therefore the topic must outlive its DataWriter.
3. The instrument cluster shows that ACC had been activated, gap setting currently in use and intent speed. This different information are instances and issues of the ACC_info_msg topic.
4. Release the accelerator pedal.
5. When the system detects a vehicle in front of the ACC-equipped car, a vehicle graphic would light in the instrument cluster. In that case the Distance topic will release an issue.

C. FOLLOWING A VEHICLE

Whenever a vehicle ahead of the ACC-equipped vehicle joins the same way or when a slower vehicle joins the same lane, the vehicle's speed is adjusted to preserve a pre-set gap space. The distance adjustment is user-adjustable. Until one of the following conditions occurs, the car maintains a consistent distance from the vehicle ahead;

- The leading car accelerates at a higher speed than the predetermined pace, an issue of the V_Lead will be published.
- The leading vehicle moves out of the lane or out of view.
- The vehicle's speed is reduced to less than 12.4 mph (20 km/h).
- The driver establishes a new separation distance.

The system applies the brakes to the vehicle in order to decelerate it down and ensure a reasonable distance from the vehicle ahead.

D. SETTING THE GAP DESTINCE/TIME

The driver is responsible for selecting a gap that is acceptable for the driving circumstances. The time and distance gaps can be adjusted by the driver using the 'TimeGap +' and 'TimeGap –' buttons. By pressing the 'Time Gap +' switch, the time gap value is increased, resulting in an increase in the clearance between the two cars. By tapping the 'Time Gap –' switch, the time gap value decreases, resulting in a decrease in the clearance between the two cars.

E. DISENGAGING THE SYSTEM

Cruise Control activity can be terminated manually or automatically through the ACC system. ACC will be deactivated if one of the following criteria exists:

- The Brake pedal is engaged (Brake_Switch Topic)
- The 'Off' button is held (OFF topic)
- The vehicle is traveling at a speed of 12.5 mph (20.12km/h) or less (V_Speed topic needs to be always updated)
- An ACC system fault is detected

F. OVERRIDING THE SYSTEM

Driver can alter the pre-set speed and gap distance by pressing the accelerator pedal. When driver let off of the accelerator pedal, the system resumes normal operation. The car's speed is reduced to the predetermined value, or to a lower value if it is trailing a slower vehicle, BRK_DEC_RQST topic in that case releases an issue

G. CHANGING THE SET SPEED

- Set the desired speed by accelerating or braking, then pressing and releasing SET+.

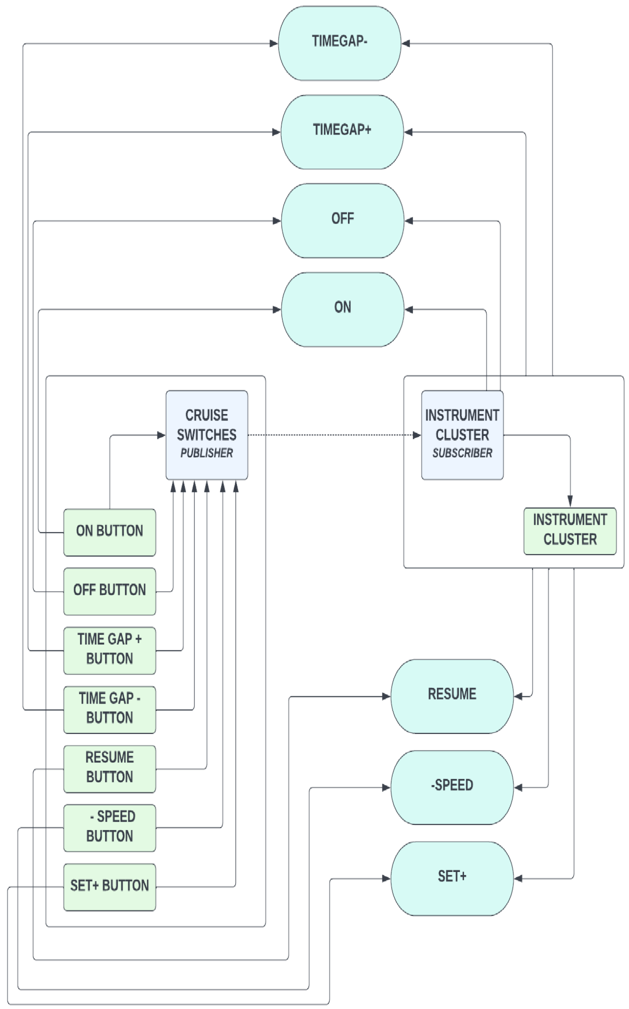

FIGURE 3. Relation between cruise switches as a publisher and instrument cluster as a subscriber.

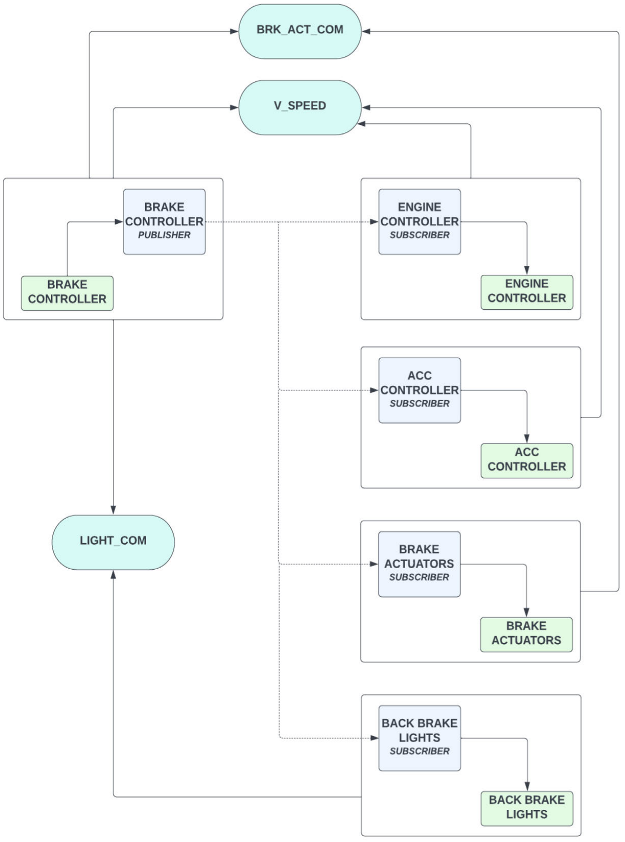

FIGURE 4. Relation between brake controller as a publisher and engine and ACC controllers, brake actuators, and back brake lights as subscribers.

- Hold down Set+ or Speed- until the information display indicates the required set speed. The vehicle's speed progressively increases until it reaches the specified speed.
- Press and release the Set+ or Speed- buttons. The programmed speed fluctuates by roughly 1.2 mph (2 km/h).
- The system may decelerate in order to reduce the vehicle's speed to the new specified value. In that case the BRK_DEC_RQST topic will be having a new issue to publish.

H. RESUMING THE SET SPEED

The resume option can be used only if the driver is knowledgeable of the pre-set speed and want to return to the pre-set speed. Press and release resume. The vehicle resumes its pre-set speed.

I. SWITCHING THE SYSTEM OFF

Press and release OFF.

VI. DESIGN DIAGRAMS AND DESCRIPTIONS

In Figure 3, the Publish/Subscribe relationship between Cruise Switches and Instrument Cluster is illustrated. The "cruise switches" is the publisher with seven DataWriters which are: on button, off button, Time gap+ button, Time gap- button, resume button,-speed button, set+ button. These DataWriters are responsible for issuing these topics; TimeGap+, Off,On,Resume,-Speed,Set+. The "instrument cluster" is a subscriber to these topics. In other words, every change that occur in any of these topics will be displayed right away on the instrument cluster for the driver to monitor.

In Figure 4, the Publish/Subscribe relationship between Brake Controller, Engine Controller, ACC Controller, Brake Actuators, and Back Brake Lights is illustrated. The "Brake Controller" is the publisher with one DataWriter that is responsible for issuing 3 different topics. Two of these topics; BRK_ACT_COM and V_speed, are in the main domain of interest for the ACC Controller, Engine Controller, and Brake Actuators. Furthermore, the Back Brake Lights is subscribing to the Light_COM topic, as every time the vehicle is braking it issues an instance in the Light_COM topic and the Back Brake Lights DataReader will be able to always keep track.

In Figure 5, the publish/subscribe relationship between Brake Switch, Instrument Cluster, Engine Controller, ACC Controller is illustrated. The Brake Switch is the publish and is responsible for issuing only one single topic which is the

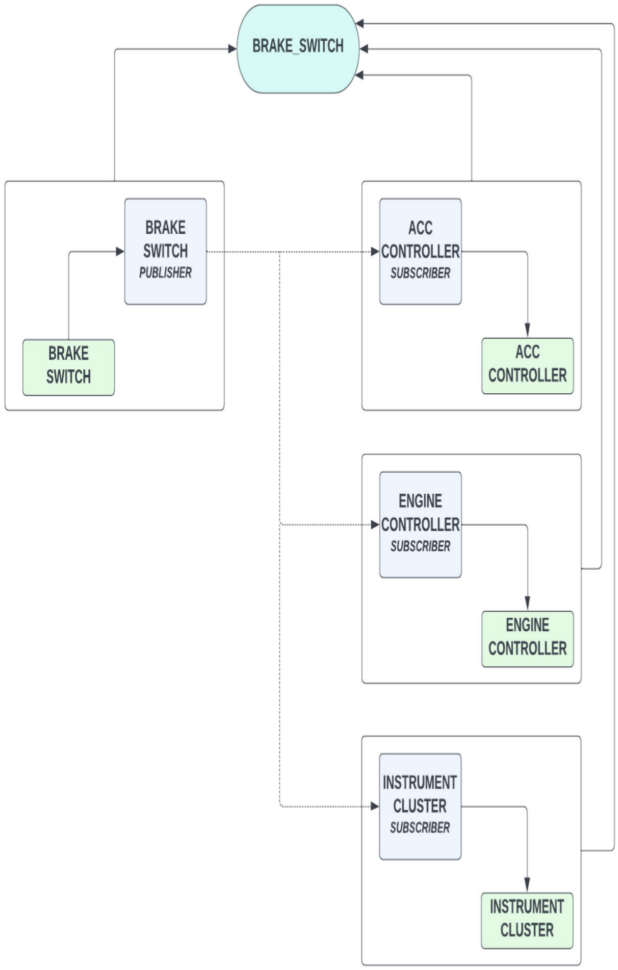

FIGURE 5. Relation between brake switch as a publisher and ACC and engine controllers, and instrument cluster as a subscribers.

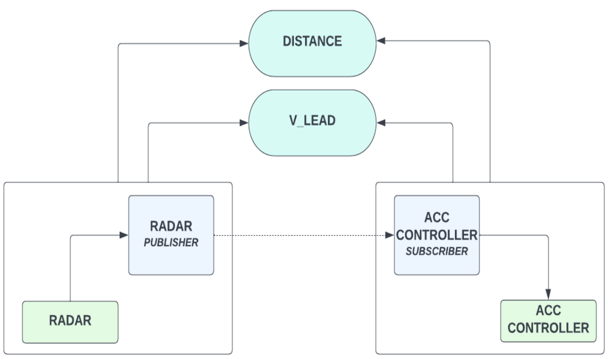

FIGURE 6. Relation between radar as a publisher and ACC controller as a subscriber.

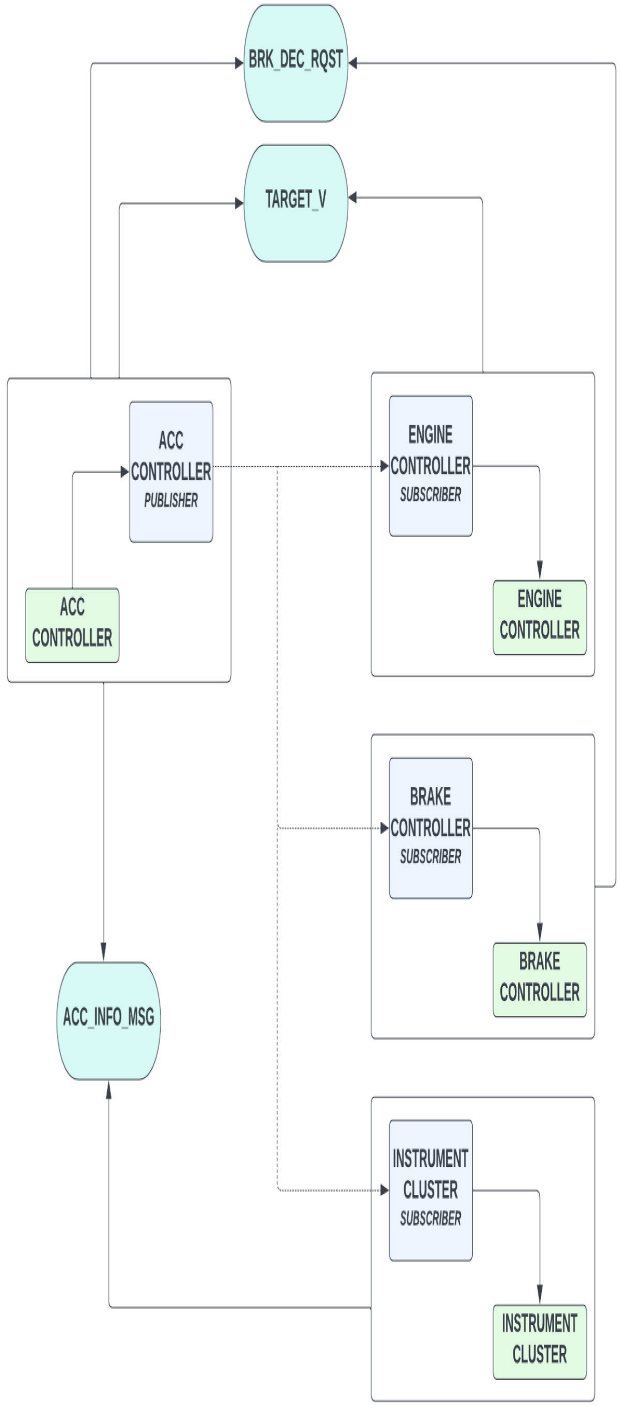

FIGURE 7. Relation between ACC controller as a publisher and engine and brake controllers, and instrument cluster as a subscribers.

Brake_Switch topic which releases a new instance each time the brake switch is activated by the driver.

In Figure 6, the publisher/subscriber relationship between the Radar and the ACC controller is illustrated. The Radar in this case is the publisher and is offering two topics that the ACC Controller is subscribing to. These two topics are the Distance and the V_Lead.

In Figure 7, the publisher/subscriber relationship between ACC Controller, Engine Controller, Brake Controller, Instrument Cluster is illustrated. The ACC Controller is this case publishes three topics. Furthermore, two of these topics; Target_v, BRK_DEC_RQST, are in the interest domain of

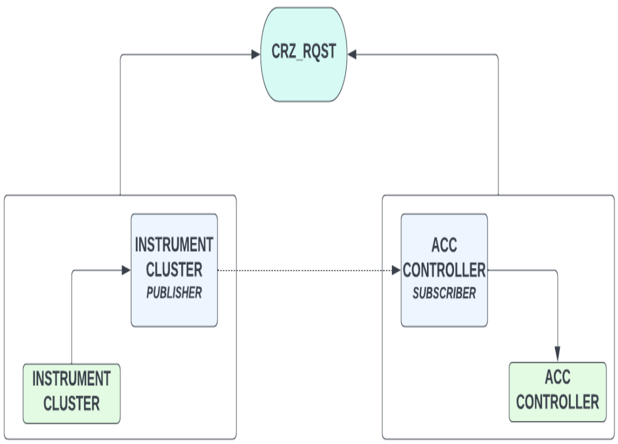

FIGURE 8. Relation between instrument cluster as a publisher and ACC controller as a subscriber.

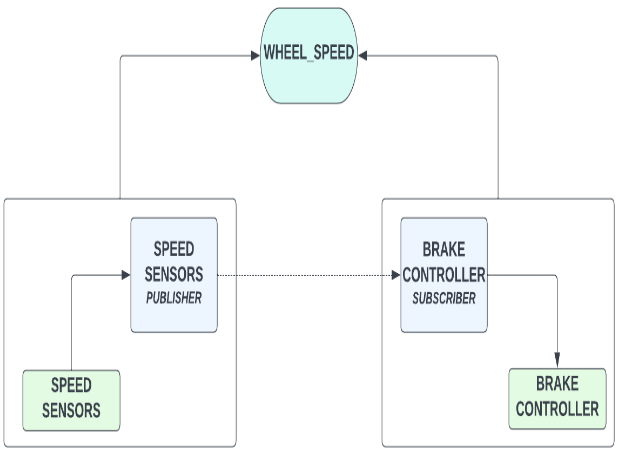

FIGURE 9. Relation between speed sensors as a publisher and brake controller as a subscriber.

the Controllers. The last topic is the ACC_info_msg which releases a new issue every time there is a change in the ACC and the Instrument Cluster subscribers to the topic to let the driver know any new updates regarding the ACC system.

In Figure 8, the publisher/subscriber relationship between the Instrument Cluster and the ACC Controller. The Instrument Cluster in this case is the publisher and is offering a single topic that the ACC Controller is subscribing to. This topic is the CRZ_RQST, an issue gets released when the driver activates the ACC system.

In Figure 9, the publisher/subscriber relationship between the Speed Sensors and the Brake Controller. The Speed Sensors in this case are the publishers and are offering a single topic that the Brake Controller is subscribing to. This topic is the Wheel_Speed. Furthermore, for the Speed Sensors publisher there can be multiple sensors for redundancy, writing the same topic.

VII. EXPERIMENTAL WORK

This section provides an evaluation of the proposed DDS-based model for ACC system. The main factors that are evaluated and have a significant impact on ACC performance are data rate, and number of publishers/subscribers. The impact of these factors is evaluated using the following performance metrics:

- ***Packet Delivery Ratio (PDR):*** it is calculated by dividing the total number of successfully received messages at the subscriber side by the total sent messages from the publisher side, as shown in Formula 1. For ideal behavior, this metric is equal to one for all scenarios

$$PDR = \frac{\sum_0^t Successfully\ received\ messages\ by\ subscribers}{\sum_0^t sent\ messages\ by\ publishers} \tag{14}$$

- ***End-to-End Delay (EED):*** it is measured from the moment of sending or publishing data on a publisher side (T_{sent}) until it is successfully received on the subscriber side ($T_{Received}$), Formula 2 shows the calculation of EED.

$$EED = \frac{\sum_0^i (T_{Receive}(i) - T_{sent}(i))}{Number\ of\ sucessfuly\ received\ messages} \tag{15}$$

- ***Memory footprint:*** this metric is measured as the number of bytes consumed by the DDS code, when it is uploaded to the sensor/actuator platform. Both RAM and ROM memories are considered in evaluating this metric.

Simulation Setup:

As per the proposed model, we found that the best simulator to be used is TOSSIM simulator [35], where this simulator allows to use sensors and actuators that have embedded systems such as Tiny OS [36] and TinyDDS [37]. The proposed model has been implemented and the simulation run for 500 seconds for each results' data point. Because the area in our proposed solution is very limited (a car system), we set the area as 10×10 meters^2. Table 4 shows the specific values for the TOSSIM simulation parameters.

TABLE 4. Simulation setup.

Parameter	Value
Topology	Squared grid
Area	10 X 10 Meter^2
Number of Nodes	5, 10, 15, 20, 25
Packet Size	20 bytes
Simulation time	500 seconds
Runs per results' data point	10

Figure 10 shows the end-to-end delay of the ACC messages for different number of nodes (2-25 nodes) and different data rates within the range of 1-5 sec inter-packet interval (IPI). As discussed earlier in this paper, the ACC system includes around 10 nodes including the ACC, brake, and engine controllers. In this figure, the delay for 10 or less is roughly less than 100 ms, which is acceptable as a real-time system.

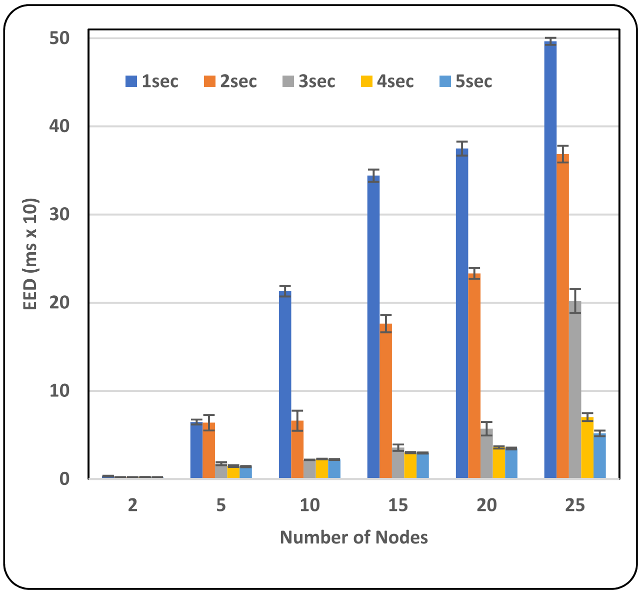

FIGURE 10. End-to-end delay with different data rates.

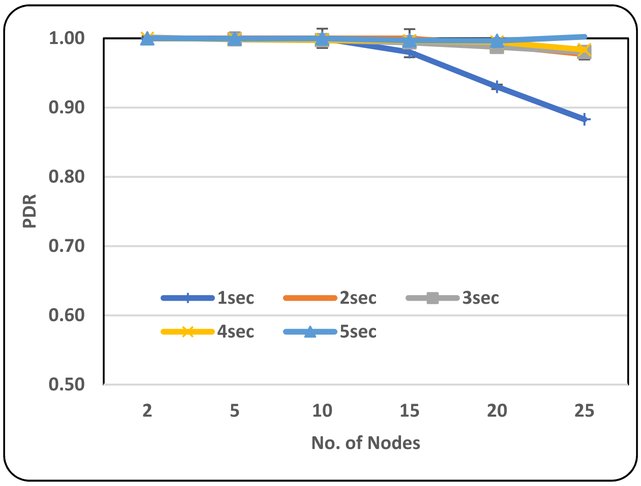

FIGURE 11. Packet delivery ratio with different data rates.

To measure how the proposed model is robust we extended the number of nodes, as depicted in the figure the worst case for the model is in the case of 1 sec IPI and 25 nodes, where the system is very overloaded, however, it reaches around 5 sec EED.

The packet delivery ratio in Figure 11 shows that the proposed model is reliable even when it is overloaded. Since the proposed model has mostly 10 nodes the results show that almost the PDR is 100% with high data traffic and condense publish/subscribe nodes. Like the EED, the PDR worst case is in the case of 1 sec IPI and 25 nodes, where the system is very overloaded, that is most likely because of interferences as well as limited queue buffer size in the used motes.

Figure 12 shows that the proposed model including the TinyOS operating system occupies only 42 % and 54 % of the system RAM and ROM, respectively, which indicates that the proposed model is light and applicable to be deployed over the embedded systems.

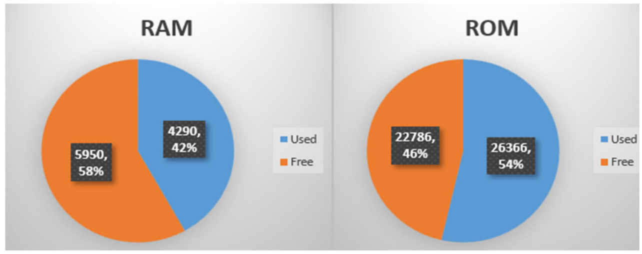

FIGURE 12. Total memory consumption by the proposed system.

VIII. FUTURE WORK

We can extend this research by integrating more systems and increasing the domain of interest, as in add the car safety system, the lighting system, and more. Furthermore, make the DDS middleware the base communication framework. This research can also be extended by investigating more QoS policies and by searching for a methodology that assist in the process of determining the parameters of QoS policies. Additionally, this study can be applied to electric vehicles to monitor the advantages of employing DDS in an EV whether it will increase the bandwidth or not, and whether latency will be improved or not. Lastly, a specific ACC strategy such as PID, MPC, and FLC can be investigated on its own.

IX. CONCLUSION

The primary aim of this study was to examine the viability of incorporating the widely recognized DDS middleware and utilizing its functionalities to augment the ACC systems. The findings, both theoretical and experimental in nature, have demonstrated the efficacy and feasibility of incorporating DDS middleware into ACC systems. One of the most significant discoveries of this investigation involves the identification of appropriate DDS QoS policies that can effectively enhance the performance of ACC systems, as well as the determination of their optimal values. Of the 22 QoS policies available, a subset of 7 policies were selected and implemented to effectively govern the behavior of the proposed model. These policies were deemed applicable and useful in achieving the desired outcomes. A comprehensive analysis of each of these QoS policies was conducted, both individually and in conjunction with one another, to elucidate their collective impact on the overall behavior of the system. It is imperative that the QoS policies and their corresponding values are precisely delineated and tailored to align with the authentic system specifications.

The experimental findings suggest that DDS is a viable and practical solution, and that the proposed model is best suited for an ACC system with around 10 nodes. This can be concluded from the data obtained. The salient observations pertaining to latency indicate that it was slightly below 100 milliseconds, thereby qualifying it as a real-time system.

Furthermore, with regard to PDR, it achieved a perfect score of 100% and remained dependable even under conditions of excessive load. Moreover, the model that has been suggested is deemed to be lightweight and suitable for deployment on embedded systems.

The experimental work conducted ultimately demonstrates the feasibility of the proposed system within the constraints of limited devices and technology. In contemporary times, it is widely held that the current array of devices exhibits a higher degree of sophistication, thereby rendering the suggested solution a suitable fit for the ACC prerequisites.

ACKNOWLEDGMENT

The authors would like to thank the Real-Time Distributed Systems Laboratory, Computer Engineering Department, King Fahd University of Petroleum and Minerals, and also would like to thank the RTI for their continuous support during the research.

REFERENCES

[1] Y. Li, Z. Li, H. Wang, W. Wang, and L. Xing, "Evaluating the safety impact of adaptive cruise control in traffic oscillations on freeways," *Accident Anal. Prevention*, vol. 104, pp. 137–145, Jul. 2017.

[2] Fatality Analysis Reporting Systems, National Highway Traffic Safety Administration. Accessed: Jun. 24, 2022. [Online]. Available: https://www.nhtsa.gov

[3] (Feb. 2022). *Early Estimate of Motor Vehicle Traffic Fatalities for the First 9 Months (January–September) of 2021*. Accessed: Jun. 24, 2022. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813240

[4] N. A. Stanton and M. S. Young, "Driver behaviour with adaptive cruise control," *Ergonomics*, vol. 48, no. 10, pp. 1294–1313, Aug. 2005.

[5] Z. Yang, Z. Wang, and M. Yan, "An optimization design of adaptive cruise control system based on MPC and ADRC," *Actuators*, vol. 10, no. 6, p. 110, May 2021.

[6] L. Yu and R. Wang, "Researches on adaptive cruise control system: A state of the art review," *Proc. Inst. Mech. Eng. D, J. Automobile Eng.*, vol. 236, nos. 2–3, pp. 211–240, Feb. 2022.

[7] S. Zhang and X. Zhuan, "Study on adaptive cruise control strategy for battery electric vehicle," *Math. Probl. Eng.*, vol. 2019, Dec. 2019, Art. no. 7971594.

[8] Z. Wei, Y. Jiang, X. Liao, X. Qi, Z. Wang, G. Wu, P. Hao, and M. Barth, "End-to-end vision-based adaptive cruise control (ACC) using deep reinforcement learning," in *Proc. 99th Annu. Meeting Transp. Res. Board (TRB)*, Washington, DC, USA, Jan. 2020, pp. 1–16.

[9] Y. Chen, G. Feng, S. Wu, and X. Tan, "A new hybrid model predictive controller design for adaptive cruise of autonomous electric vehicles," *J. Adv. Transp.*, vol. 2021, Mar. 2021, Art. no. 6626243.

[10] M. Reke, D. Peter, J. Schulte-Tigges, S. Schiffer, A. Ferrein, T. Walter, and D. Matheis, "A self-driving car architecture in ROS2," in *Proc. Int. SAUPEC/RobMech/PRASA Conf.*, Jan. 2020, pp. 1–6.

[11] A. Adiththan and K. Ravindran, "QoS-oriented management of multi-vehicle coordinated cruise control in uncertain environments," in *Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Nov. 2017, pp. 55–62.

[12] A. Reschka, M. Nolte, T. Stolte, J. Schlatow, R. Ernst, and M. Maurer, "Specifying a middleware for distributed embedded vehicle control systems," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Dec. 2014, pp. 117–122.

[13] Y. Bai, Z. Wang, X. Wang, and J. Wang, "AutoE2E: End-to-end real-time middleware for autonomous driving control," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 1101–1111.

[14] X. Chen, J. Yang, C. Zhai, J. Lou, and C. Yan, "Economic adaptive cruise control for electric vehicles based on ADHDP in a car-following scenario," *IEEE Access*, vol. 9, pp. 74949–74958, 2021.

[15] R. C. Zhao, P. K. Wong, Z. C. Xie, and J. Zhao, "Real-time weighted multi-objective model predictive controller for adaptive cruise control systems," *Int. J. Automot. Technol.*, vol. 18, no. 2, pp. 279–292, Apr. 2017.

[16] D. L. Luu, C. Lupu, and T. Van Nguyen, "Design and simulation implementation for adaptive cruise control systems of vehicles," in *Proc. 22nd Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2019, pp. 1–6.

[17] X.-W. Chen, J.-G. Zhang, and Y.-J. Liu, "Research on the intelligent control and simulation of automobile cruise system based on fuzzy system," *Math. Problems Eng.*, vol. 2016, Jun. 2016, Art. no. 9760653.

[18] A. Patel, N. B. Haupt, and P. Liggesmeyer, "Evaluating adaptation behavior of adaptive cruise control (ACC) in autonomous vehicles," 2019, doi: 10.13140/RG.2.2.17306.16327.

[19] B. AL-Madani and H. Ali, "Data distribution service (DDS) based implementation of smart grid devices using ANSI C12.19 standard," *Proc. Comput. Sci.*, vol. 110, pp. 394–401, Dec. 2017.

[20] A. Llorens-Carrodeguas, C. Cervello-Pastor, and I. Leyva-Pupo, "A data distribution service in a hierarchical SDN architecture: Implementation and evaluation," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–9.

[21] M. M. Madden and P. C. Glaab, "Distributed simulation using DDS and cloud computing," in *Proc. 50th Annu. Simul. Symp.*, Apr. 2017, pp. 1–12.

[22] S. El-Ferik, B. Almadani, and S. M. Elkhider, "Formation control of multi unmanned aerial vehicle systems based on DDS middleware," *IEEE Access*, vol. 8, pp. 44211–44218, 2020.

[23] B. Almadani, S. Khan, M. N. Bajwa, T. R. Sheltami, and E. Shakshuki, "AVL and monitoring for massive traffic control system over DDS," *Mobile Inf. Syst.*, vol. 2015, Oct. 2015, Art. no. 187548.

[24] A. Cantelli-Forti, A. Capria, A. L. Saverino, F. Berizzi, D. Adami, and C. Callegari, "Critical infrastructure protection system design based on SCOUT Multitech seCurity system for intercOnnected space control groUnd staTions," *Int. J. Crit. Infrastruct. Protection*, vol. 32, Mar. 2021, Art. no. 100407.

[25] V. Ibarra-Junquera, A. González, C. M. Paredes, D. Martínez-Castro, and R. A. Nuñez-Vizcaino, "Component-based microservices for flexible and scalable automation of industrial bioprocesses," *IEEE Access*, vol. 9, pp. 58192–58207, 2021.

[26] H. Muslim and M. Itoh, "A theoretical framework for designing human-centered automotive automation systems," *Cognition, Technol. Work*, vol. 21, no. 4, pp. 685–697, Nov. 2019.

[27] H.-A. Jacobsen, "Topic-based publish/subscribe," in *Encyclopedia of Database Systems*. Boston, MA, USA: Springer, 2009, pp. 3127–3129.

[28] J. F. Inglés-Romero, A. Romero-Garcés, C. Vicente-Chicote, and J. Martínez, "A model-driven approach to enable adaptive QoS in DDS-based middleware," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 1, no. 3, pp. 176–187, Jun. 2017.

[29] *Data Distribution Service (DDS)*. Accessed: Jun. 24, 2022. [Online]. Available: https://www.omg.org/omg-dds-portal/

[30] *The Object Management Group Official*. Accessed: Jun. 24, 2022. [Online]. Available: https://www.omg.org/index.htm

[31] T. Guesmi, R. Rekik, S. Hasnaoui, and H. Rezig, "Design and performance of DDS-based middleware for real-time control systems," *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 188–200, 2007.

[32] J. Yongguo, L. Qiang, Q. Changshuai, S. Jian, and L. Qianqian, "Message-oriented middleware: A review," in *Proc. 5th Int. Conf. Big Data Comput. Commun. (BIGCOM)*, Aug. 2019, pp. 88–97.

[33] H. Pérez and J. J. Gutiérrez, "Modeling the QoS parameters of DDS for event-driven real-time applications," *J. Syst. Softw.*, vol. 104, pp. 126–140, Jun. 2015.

[34] D. Alghazzawi, G. Aldabbagh, and A. S. A.-M. AL-Ghamdi, "ScaleUp: Middleware for intelligent environments," *PeerJ Comput. Sci.*, vol. 7, p. e545, Jun. 2021.

[35] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst.*, Nov. 2003, pp. 126–137.

[36] *TinyDDS: Publish/Subscribe Middleware for Wireless Sensor Networks*. Accessed: Jun. 24, 2022. [Online]. Available: https://code.google.com/archive/p/tinydds/
[37] TOSSIM. (2003). *TinyOS Documentation Wiki*. Accessed: Jun. 24, 2022. [Online]. Available: http://docs.tinyos.net/index.php/TOSSIM

BASEM ALMADANI received the B.Sc. degree in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), in 1997, and the M.Sc. degree in industrial automation and the Ph.D. degree from the Institute for Automation, Montan University of Leoben (MUL), Austria, in 1999 and 2005, respectively. He joined the SABIC International Team to manage industrial automation project in Vienna, Austria. He joined KNAPP Systems Integration in 2001, Leoben, Austria, as a Logistics Automation Specialist. He was the Chairperson of the Computer Engineering Department, KFUPM, from 2009 and 2014, where he is currently the Director of the Alfozan Academy, for Leaders Development in Non-Profit Sector Program. His research interests include real-time systems integration, distributed systems, middleware, the IIoT, and IR 4.0.

NAJOOD ALSHAMMARI received the B.S. degree in electrical engineering from Prince Mohammed Bin Fahd University, in 2020. She is currently pursuing the M.S. degree in computer networks engineering with the King Fahd University of Petroleum and Minerals. She studies real-time systems integration, distributed systems, middleware, and autonomous EVs with the King Fahd University of Petroleum and Minerals. She is also a licensed game developer.

ANAS AL-ROUBAIEY received the B.S. degree in computer engineering from AAST Alexandria, Egypt, in 2001, and the M.S. degree in computer networks and the Ph.D. degree in computer science and engineering from the King Fahd University of Petroleum and Minerals, Saudi Arabia, in 2009 and 2015, respectively. He was a Teaching Assistant with Sana'a and Taiz University, Yemen, until 2004. He is currently a Researcher with the King Fahd University of Petroleum and Minerals. He works as a reviewer for a number of ISI journals. His research interests include intrusion detection systems, middleware, systems integration, real time and distributed systems, wireless sensor and actuator networks, energy aware protocols, and power harvesting in WSN. He has several publications and patents in these areas.