Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1149842, 18 pages
https://doi.org/10.1155/2022/1149842

Research Article

IoT-Based Response Time Analysis of Messages for Smart Autonomous Collision Avoidance System Using Controller Area Network

Anil Kumar Biswal,[1] Debabrata Singh,[2] Binod Kumar Pattanayak ,[1] Debabrata Samanta ,[3] Amit Banerjee ,[4] A. Y. Seteikin,[5,6] and I. G. Samusev [5]

[1]*Department of CSE, ITER, SOA Deemed to Be University, India*
[2]*Department of CA, ITER, SOA Deemed to Be University, India*
[3]*Department of Computer Science, CHRIST University, India*
[4]*Physics Department, Bidhan Chandra College, Asansol 713 303, India*
[5]*Immanuel Kant Baltic Federal, University, Kaliningrad, 236000, Russian Federation and Amur State University, Blagoveshchensk 675027, Russia*
[6]*Amur State University, Blagoveshchensk 675027, Russia*

Correspondence should be addressed to Amit Banerjee; amitbanerjee.nus@gmail.com and I. G. Samusev; is.cranz@gmail.com

Received 12 August 2021; Accepted 3 March 2022; Published 8 April 2022

Academic Editor: Rajesh Kaluri

Many accidents and serious problems occur on the road due to the rapid increase in traffic congestion in all sections of the country. Autonomous vehicles provide a solution to successfully and cost-effectively avoid this problem while minimizing user disruption. Currently, more engaging electromechanical elements with an analog interface are used to develop affordable automobiles for efficient and cost-effective operation for a smart driving platform with a semiautonomous automobile, strengthening the vehicle involvement of the driver while increasing safety. As a result, it takes longer for various car elements to respond, which causes more problems during message transmission. This project aims to create a Controller Area Network (CAN) for analyzing message response times by incorporating a few application nodes on the IoT platform, such as an antilock braking system, flexible cruise control, and seat belt section, for some real-time control system applications. These application nodes are car analytical parts that are linked to IoT modules to prevent collisions. An autonomous device for collision avoidance and obstacle detection in a vehicle can impact road accidents if the CAN protocol is implemented.

1. Introduction

In recent years, traffic congestion, driver drowsiness and reckless driving represent a big problem in the different areas around the world that are seriously affected by road accidents in the transport system [1]. Unconditional circumstances are controlled by intelligent autonomous vehicles because of the above. Therefore, through the implementation of collision avoidance mechanisms in the vehicle, the automation domain offers a forum for monitoring reckless driving as well as driver fatigue [2]. The "Automotive Serial Controller Area Network" protocol is used for designing an intelligent control car with a huge range of serial bus communication control system [3].

Large types of embedded systems require high-speed communication platforms for providing automotive industrial control. But various industries are not supported with automation that needs to the operated with Controller Area

Network (CAN) protocol [4]. A serial communication bus like CAN was considered in the International Standardization Organization (ISO), which has replaced the complex wiring control with a two-wire bus, thereby adding a multi-master communication serial bus that can transmit messages to various parts of the network system [5].

Currently, the Automobiles are being constructed with microcontrollers and more electrical parts as we know that is the central part of the controlling unit and various types of devices or circuitries connected to it. This process is very complex to interpret and improve performance by using several connections and electrical lines linked to a microcontroller [6]. The communication area for the project is an implanted CAN networking system which provides effective data transfer, allowing multiple microcontrollers and devices to be connected with a popular CAN bus using the CAN protocol, then the connection of all items with the consideration of optimum priority and speed [7].

This protocol also offers a high-speed serial data frame communication interface, low-cost physical medium, short message frame length, and, at the same time, it adds a high-level detection or correction mechanism for errors in different communication network nodes [8]. The evolution of embedded systems and software has been used in modern times to build smart autonomous vehicles over 40-50 percent globally and this percentage of progress is only expected to increase with improvising road safety and security features [9] [10]. Due to the process of digitization for constructing smart vehicles with the use of IoT modules that can create a huge number of datasets [11]. So that, the dimensionality reduction and security of datasets are required to be managed and also yield good results through the blockchain-big data technique [12]. In general, the protocols between the network (sensor) nodes for physical communication with the IoT data link layers, the sensor nodes, are described by the CAN protocol [13]. Here, the ultrasonic sensor node application process is used to measure the distance between the vehicle and the road barrier [14] [15].

This system determines the distance of the obstacle by an ultrasonic sensor to control the motor speed that has been designed using Arduino UNO, IoT modules, and CAN base serial communications protocol. When the sensor node is connected to the CAN bus, which provides a rapid response to measure the distance of an obstacle, the message is automatically forwarded to the Arduino Uno module to track the vehicle's engine movement and steering. This proposed system provides an environment to enhance the driver-vehicle platform to make a semi-autonomous vehicle system with the help of developing and implementing a digital driving system [16] [17]. The optimal response time calculation is impossible in the existing system, which takes maximum time to communicate with each part of the vehicle due to massive datasets. These datasets shorten the response time of an existing system that causes vehicle accidents while driving.

1.1. Motivation of This Work. Designing a process of response time analysis of a smart autonomous collision avoidance system messages based on IoT modules and CAN serial communications protocol to prevent any road accidents by taking an optimized range of message length and message ID for providing timing response in conversion. In recent years, the autonomous operation has been extensively applied to vehicles for road safety issues.

1.2. Contribution to This Work

(i) The smart vehicle is developed by the Controller Area Network (CAN) that is accessed in various real-time suits to link internal-level communication facilities to shared units of car control systems, e.g., industrial and home automation and medical equipment, which is a "broadcast" type of bus. In other words, there is no address part of sending or receiving nodes. The network can accept to receive or transmit the messages sent by all nodes, where the acceptance test is performed after receiving the message from each node

(ii) The messages are checked by each node whether it is irrelevant to that particular node or not. When a message is pertinent, then it is received by that node. Otherwise, it is not accepted. The priority node can send the first message for transmission, which depends on an 11-bit identifier. Here, an identifier is uniquely identified all over the network and is used to tag the content of the message. A numeric value is added to each message, which controls its priority on the bus, thereby recognizing the contents of the message

(iii) When the bus is not loaded by any task, then some nodes can be ready to communicate with each other. But during this period, where the CAN bus attempts to forward messages from more than two nodes concurrently, then the identifier field is uniquely defining the priority of the message through the network. The messages are securely transmitted in the sequence of priority without missing anyone, which is possible with this technique. If a numerical value of the identifier is lower, then it is treated as a higher priority. This means that the message with more prevalent ID bits (i.e., Bit 0) will overwrite all nodes so that only the predominant message will finally be acknowledged by each node after arbitration of the ID

(iv) Using the Arduino controller and serial communication protocol on different device nodes, collision avoidance, and obstacle detection techniques are implemented on the smart vehicle via IoT modules. To detect any obstacle through a different ultrasonic sensor on the road, we can produce a message frame to relay the node to a vehicle's engine

The rest of the paper is structured as follows: the literature survey pertaining to this field is included in Section 2. Section 3 describes the network protocol model and the proposed system, and Section 4 represents the proposed framework and operating theory, along with its implementation processes and components. In Section 5, we describe the simulation setup and result from analysis, and in Section 6, we conclude our paper with some references.

2. Literature Survey

Control Area Network (CAN) is detailed in [18] that provides a communication network between control units in automotive industries. CAN provides vast advantages and then it is widely used in distinct industries including military, aviation, electronics, factories and many more. Here, the microcontrollers and devices communicate among themselves using CAN in the absence of a host computer and there is no need to follow heavy access of the main controller. In [19], The author describes CAN-BUS to be an essential network technology for communication which is implemented in the automobile network communication sector with some characteristics like real-time implementation, reliability, and flexibility.

In [20], the current wireless home automation requires a greater amount of RF recipients and thus the frequency range varies. Electromagnetic waves may lose messages, and the cost and complexity of multi-home automation will be significant. This process is crucial for detecting problem areas. Due to that, it allows controlling only limited devices. The author talks about a Controller Area Network (CAN) bus used to send and receive messages between automotive devices [21]. There are possibilities of errors when transmitting messages via the nodes. To detect those errors, a Controller Area Network Adaptive Fault Diagnostic Algorithm detects all of the CAN's defective nodes.

In [22], the authors discuss the current parameters of vital signs for patients in critical care units; patient's bedside are equipped with devices that keep intensivists and other medical staff informed. This information allows paramedical personnel to take the necessary measures for disease prevention and cure. Extracting CAN messages from automotive ECUs can be made successfully as detailed in [23]. It gives the details of the construction of software and hardware, which interfaces directly into the car with the CAN network.

It includes CAN bus transceiver behavioural models. Thermal behaviour can be allowed for different types of simulations for verification in reasonable CPU time from core verifications for detailed analysis of the integrity of signals [24]. This review examines the research done on the Controller Area Network (CAN) reliability analysis. In recent decades, schedulability analysis has been extended to an advanced technique that can determine whether or not the time limits of several jobs performed by a single CPU or a distributed system for nontrivial systems. [25]. This is a description and illustration of a reliability analysis method that focuses on auto systems based on CAN, that also considers the impact of the error on schedulability analysis [26].

In [27] proposed an automotive CAN cluster for processing messages by using a gateway mechanism. So, this is used for worst-case response analysis (WCRT) for finding lower and upper bound on the response time of the CAN cluster of automobiles. It is efficient to monitor a large-scale CAN cluster, and then its performance will be improved by reducing unnecessary conservation in process of designing. The WCRT analysis of CAN with sporadic message execution in a multicore automotive gateway protocol has been designed by Xie et al. [28]. This process is constructed on global and portioned scheduling to evaluate real message sets and guides the design optimization. This gateway technique can remove the bottleneck of the message execution with the use of a small message execution delay, but its real-time process can be improved through multiple execution units. Alaei et al. [29] proposed a method for improving message response time by using a statistical based algorithm. In this paper, the stuff bits were reduced through the Statistical Mask Calculation (SMC) which provided better performance than the existing process. But the validation will require to improve in the reliability of the CAN network by minimizing bit stuffing.

The CAN is generally applied in various sectors like industrial, home automation, transportation, medical sector, and thermal plant, etc. that shows worst-case response time (WCRT) at the time of execution. This occurs due to delay in the periodic frame of the message, desynchronization of the message frames, improper scheduling of frames, etc., which is the main reason for WCRT. But in this current or proposed system is improvising the process response time of message transmission in various units of the system that makes the whole process full of automation. This paper provides a technique for analyzing the response time of CAN through an enhanced method of bit stuffing, message format, and error handling mechanism to optimal way for the handling of huge datasets. The existing process of the CAN bus is designed to measure lower and upper bound on the response time by using a cluster gateway algorithm which is performed their activity on a large-scale cluster of the message. But it sometimes does not give perfect responses due to unnecessary conversion in the process. That is why this proposed approach algorithm provides a perfect observation to finding the response time of optimized or prioritized message conversion in CAN bus that helps to avoid the cause of the collision.

3. Proposed System

An automation domain that is a versatile way to monitor the motor movement from a collision due to any obstacle is the objective of the proposed method. The primary goal of this phase is to ensure protection from unconditional road accidents due to traffic congestion and reckless driving.

3.1. Serial Communication Protocols. The growing demand for transmission of message has been controlled through the different protocols for communication, which is based on applications to build in a networked and internet-connected environment [30] [31] [32]. But these protocols vary from one another at the time of communication. So the upper and lower end protocols form the transmission procedures covering the various communication nodes of automobiles [33]. The device's CAN control bus and address bus are referred to as the higher end. Together with the increased focus on the distributed systems and networking, the cost benefit and advanced capabilities of silicon technology have led to the need for new highly organized communication methods in the area of field bus application. The automobile expects scalable control systems with a high

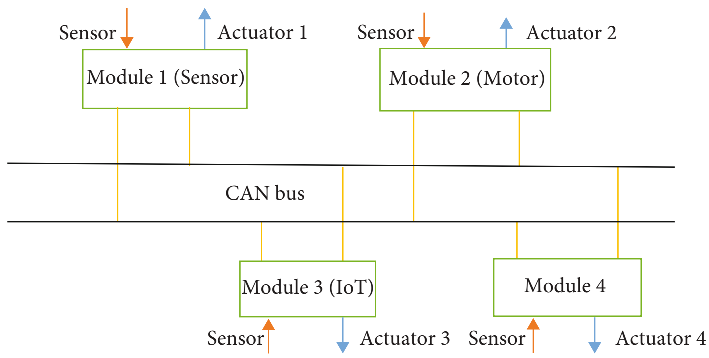


Figure 1: Illustration of a distributed CAN bus network.

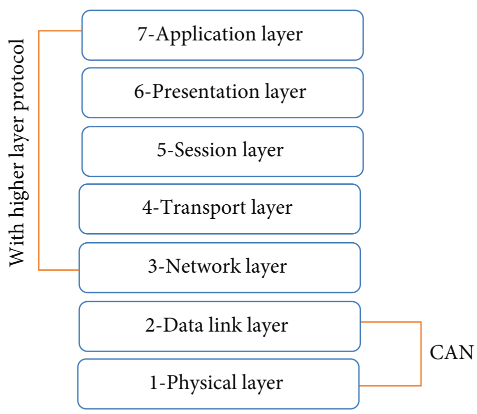


Figure 2: Controller Area Network (CAN) protocol defines in OSI model

standardization degree. The high level of standardization in the hardware and software modules leads to reusable systems that are ready to adapt in any single application setting to the various requirements and solutions [34]. The automotive vehicle has been developed by adding many electronic components with IoT modules for providing safety and to improve collision avoidance systems. Consequently, they need more and more hardwired, dedicated signal lines because of the complexity of the sharing data transmission control architecture of the system. This prompted the replacement by network architecture of the current wiring mode where the network system communicates via a common bus to all of the nodes. So this communication is ideally performed through CAN protocol which was developed by R. Bosch [35]. The CAN is used to communicate together in real time at speeding up to 1 Mbps, sensor and actuator via a two-wire serial data bus [36]. Focused on the concept of the "Shared Variables," the Virtual Levelled Systems Architecture (VLSA) model forms the generic interface architecture that is central to the CAN protocol [37]. Individual tasks are handled by distributed controllers in the VLSA architecture, with each one responsible for a portion of the total control programme. Through their sensors and actuators, nodes in a distributed system interact with the real process. The nodes use a dynamic, priority-based arbitration system to send messages on the bus. Figure 1 shows illustration of a distributed CAN bus network.

The nodes use a dynamic arbitration method based on priority to pass messages on the bus [38]. The nodes filter out the corresponding messages by filtering the message algorithm. Any message sent on the bus is delivered to every node in your network. Based on the message received, the application will send control signals to the device via the actuators. Jitter occurs when data packets are sent over your network connection with a temporal delay. Congestion on the network, as well as route changes, are common causes.

3.1.1. CAN Protocol. The CAN is described as two protocol standards such as ISO 11898 and ISO 11519 [39]. The ISO 11898 standard monitors high speed communication up to 1Mbps in physical layer of OSI model. The upper limit for ISO 11519 is 125 kbps that is consisting of a sub-layer of Logical Link Control (LLC) and a sub-layer of Media Access Control (MAC) in data link layer [40] [41]. Controller Area Network (CAN) Protocol defines in OSI model represents in Figure 2.

To keep data and monitor information, the Data Link Layer constructs data frames. Generally, some additional services such as detecting frames with bit stuffing and also used to re-transmit faulty data frames at the time of communication.

The CAN Physical Layer in one given network transmits data between different nodes; it decides the mode of transmission of signals and thus addresses issues such as the encryption, timing and synchronization of the communicated data signal [42]. With the implementation of the CAN protocol, the receivers of the sensor nodes data set will be transmitted to the control unit of the device, which normalizes the physical and data link layers of the OSI communication model for the automation domain, while the higher-layer protocols such as CAL/CAN Open and the CAN Kingdom, System Net, define the application layer [41]. The upper layer of ISO/OSI model's highest level, the application layer, communicates with an application program. The OSI layer closest to the end-user is the application layer.

3.2. CAN Message Transfer. The maximum load for utility is 94 bits and restricted format communications of varying but limited lengths are used by CAN. There is no particular address in the messages. Instead, it can be thought that the messages are addressed by four separate frame types for

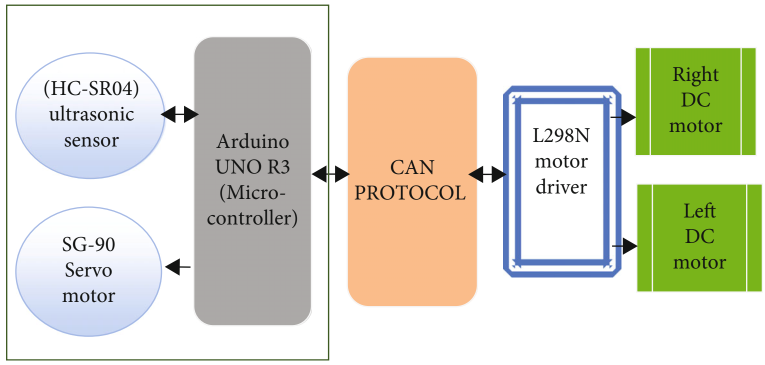


FIGURE 3: Proposed block diagram of smart autonomous collision avoidance system using the CAN protocol.

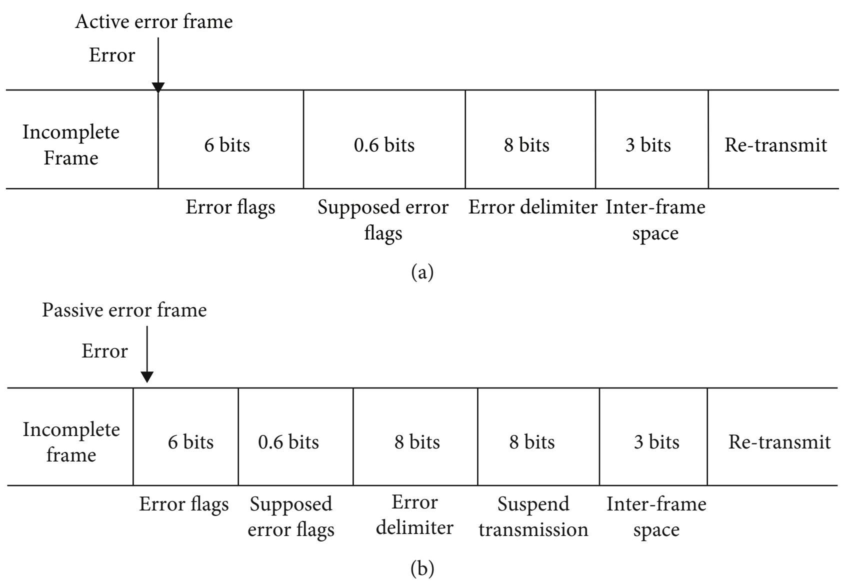


FIGURE 4: CAN error frame: (a) active error frame and (b) passive error frame.

for$i = jdownto1$
$\pi_{i,2\cdots n_i} = \pi_{i,1}$
$RT_i = WCRT(\pi_{i,2\cdots n_i})$
$while$(RT_i > DL_i)do
$decrement\pi_{i,2\cdots n_i}$ (maximum priority)
if$(\pi_{i,2\cdots n_i} < \pi_{i,1})then$
$returnFail$
$endif$
$RT_i = WCRT(\pi_{i,2\cdots n_i})$
$endwhile$
$endfor$
Return success

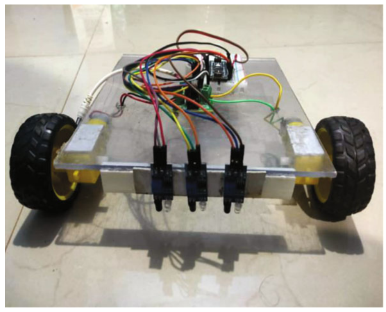
ALGORITHM 1: Message priority task algorithm.

communications such as a data frame, a remote frame, and an error frame for sending and reporting a detected data.

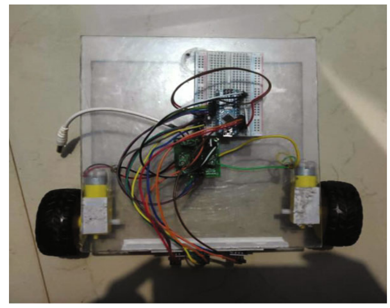
3.2.1. Data Frame. The CAN systems are used to transfer eight bytes of data frames with fixed data lengths through the network. Eight separate bit fields are composed of a message frame: frame start bit, data arbitration, control, data, CRC frame, acknowledgement, frame end field and Inter-Data space. So this protocol is defined by two frameworks base and extended format [43] [44].

The CAN 2.0A specifies base format CAN systems with standard 11-bit frame identifiers. But the CAN 2.0B identifies extended a format CAN system that has 29-bit frame identification. Where the CAN 2.0B supports both 11 bit and 29 bit identifiers, but the CAN 2.0A only supports 11-bit frame. The extended format is used on complex heavy traffic networks where the number of messages generated by network transmitters is greater than the number of possible CAN ID codes that may be given to them. The Standard CAN 11-bit ID provides the Extended CAN 29bit for 2, or 2047 separate message ID, whereas the CAN 29bit ID is stretched to provide 2 or 538 million identifiers [45].

So the vehicular conflicts may arise due to cross-wind, unbalanced friction coefficients, also a flat tire, the driver's behaviour is taken into account. The CAN bus is used by the Arduino UNO R3 module system as it relies on many IoT control units residing in the vehicle's Engine Control Unit (ECU- L298N Motor Driver) which depends significantly on the selection of the braking mechanisms (such as



(a)



(b)

Figure 5: (a) Front and (b) top view of Smart Autonomous Collision Avoidance Vehicle.

Table 1: Details of Network Parameters.

Network parameters	Value in numbers (nos) (variable message ID)	Value in numbers (nos) (variable message length)
Message length	5-8	1-5000
Message ID	1-5000	1-500

hydraulic, pneumatic systems, electro-hydraulics, or even the electro-mechanics) and the usability of the Electronic Control Unit (ECU) as depicted in Figure 3 [46].

4. Working Principle and Methodology

4.1. Measuring Response Time of CAN. Measurement of the CAN message worst-case latencies for real-time analysis can be conducted on a fixed priority response time analysis scheduling standard [47]. The response time can be calculated using a worst-case message queuing configuration. The typical way to express the worst case action is to assume a collection of streams of traffic and also producing a fixed priority of queue messages on a periodic basis [48].

In the proposed model with message streams (MS) is processed in CPU scheduling with three elements of the messages $<J_i, Q_i, C_i>$, where J_i is the queuing jitter, Q_i is the queuing delay and C_i is the communication delay of message i. When lower priority messages are forwarded it take a long time to be delayed in the queue (Q_i). Then the real-time need to send the message by bus, due to communication delay (C_i). The response time of worst-case error messages of the CAN bus is generally calculated, which shows the overhead error EO_i in terms of E(t) denotes the maximum time required to signal and retrieve errors during the interval t. The response time analysis of worst-case (WR) can be determined by:

$$WR_i = J_i + Q_i + C_i,$$

$$Q_i = B_i + \sum_{j\epsilon hp(i)} \left\lceil \frac{(Q_i + J_j + \tau_{bit})}{T_j} \right\rceil C_j + E(Q_i + C_i),$$

$$WR_i = J_i + B_i + \sum_{j\epsilon hp(i)} \left\lceil \frac{(Q_i + J_j + \tau_{bit})}{T_j} \right\rceil C_j + P_k,$$

$$P_k = C_i + E(Q_i + C_i), B_i = \max_{\forall k\epsilon lm(i)} (C_k),$$

$$C_i = \left(mh + 8P_i + 15 + \left\lceil \frac{(mh + 8P_i - 1)}{4} \right\rceil \right) T_{bit},$$

$$EO_i = 15_t au_{bit} + \max_{K\epsilon hp(i)\cup\{i\}} (C_k + 31_t au_{bit}), \quad (1)$$

(i) Captures the effect of external interference as an error in many frames, rather than allowing the interference pattern to be defined and that explanation gives the consequence of message transmissions.

(ii) The period of interference does not reflect the potential delay, e.g. assuming that interference with duration $15_t au_{bit}$ will in the worst case give an error overhead.

(iii) Only allows relatively simple interference patterns with an initial burst and a residual error rate.

(iv) Does not conveniently capture interference from multiple sources.

4.1.1. Features of CAN Error Handling. Error Active is the default mode for a node. When any of the two Error Counters rises above 127, the node goes into Error Passive mode, and when the Transmit Error Counter rises beyond 255, the node goes into Bus Off mode. When an Error Active node identifies errors, it transmits Active Error Flags. The signal was scattered throughout propagation due to several nodes exchanging sample thresholds. The CAN bus can cause errors which are also used for error finding and auto-checking tools to attain resources of source-based controlling, bit stuffing, CRC bit checks, as well as testing format of the message frame [49]. So, it is depicted in Figures 4(a) and 4(b).

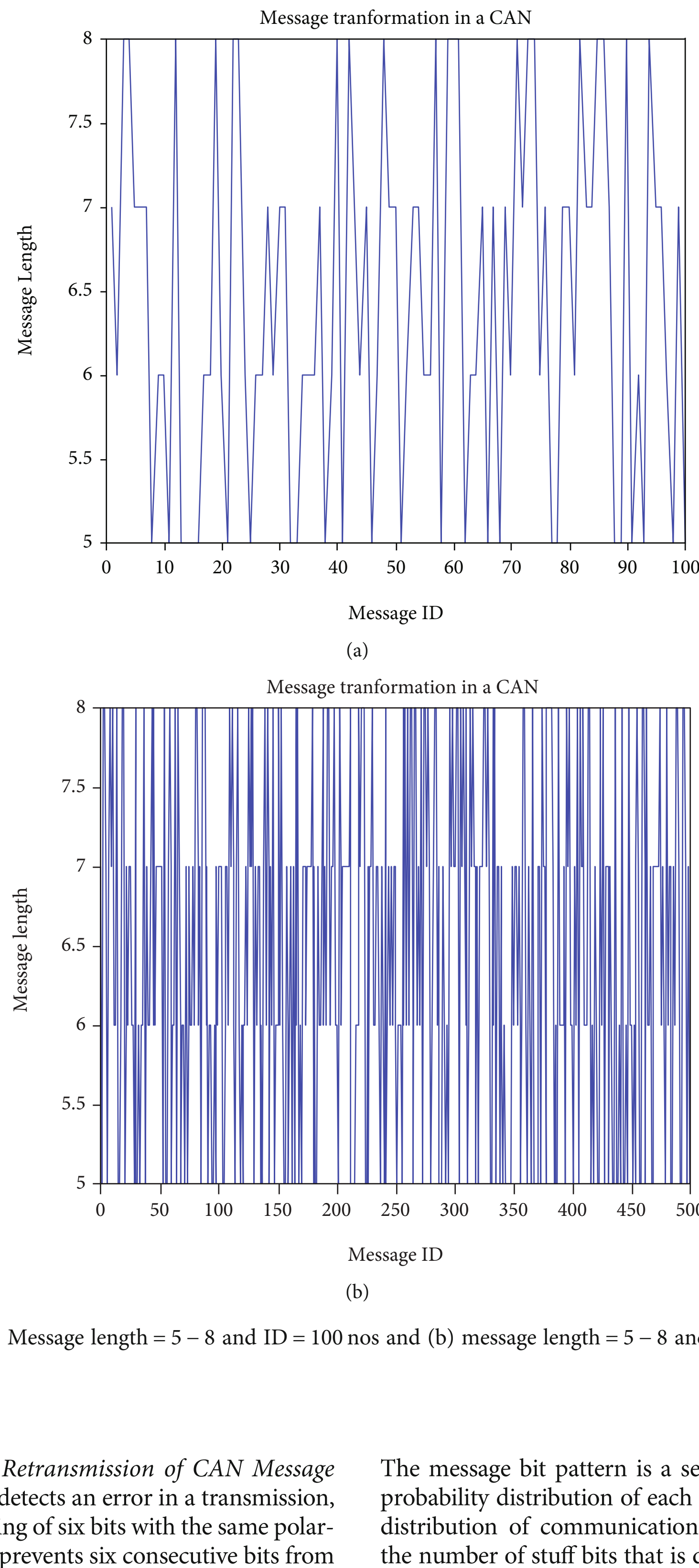


FIGURE 6: (a) Message length = 5 – 8 and ID = 100 nos and (b) message length = 5 – 8 and ID = 500 nos.

4.2. Bit-Stuffing Effect and Retransmission of CAN Message Frame. When a CAN node detects an error in a transmission, it sends an error flag consisting of six bits with the same polarity. The bit stuffing method prevents six consecutive bits from having the same polarity by adding a bit of opposite polarity after the fifth bit. If the number of bit-stuffing increases then the re-transmission of CAN messages can be increased. But the bit-stuffing decreases then the re-transmission of CAN messages can be decreased at the time of communication. The message bit pattern is a set of stuffed bits that requires probability distribution of each bit frame format [50]. So, the distribution of communication time can be collected from the number of stuff bits that is defined as

$$CD_m(t) = CD_m(t) + \phi(b)\tau_{bit} \tag{2}$$

At the time when the message communication is not successfully transmitted to the destination, it is due to the delays

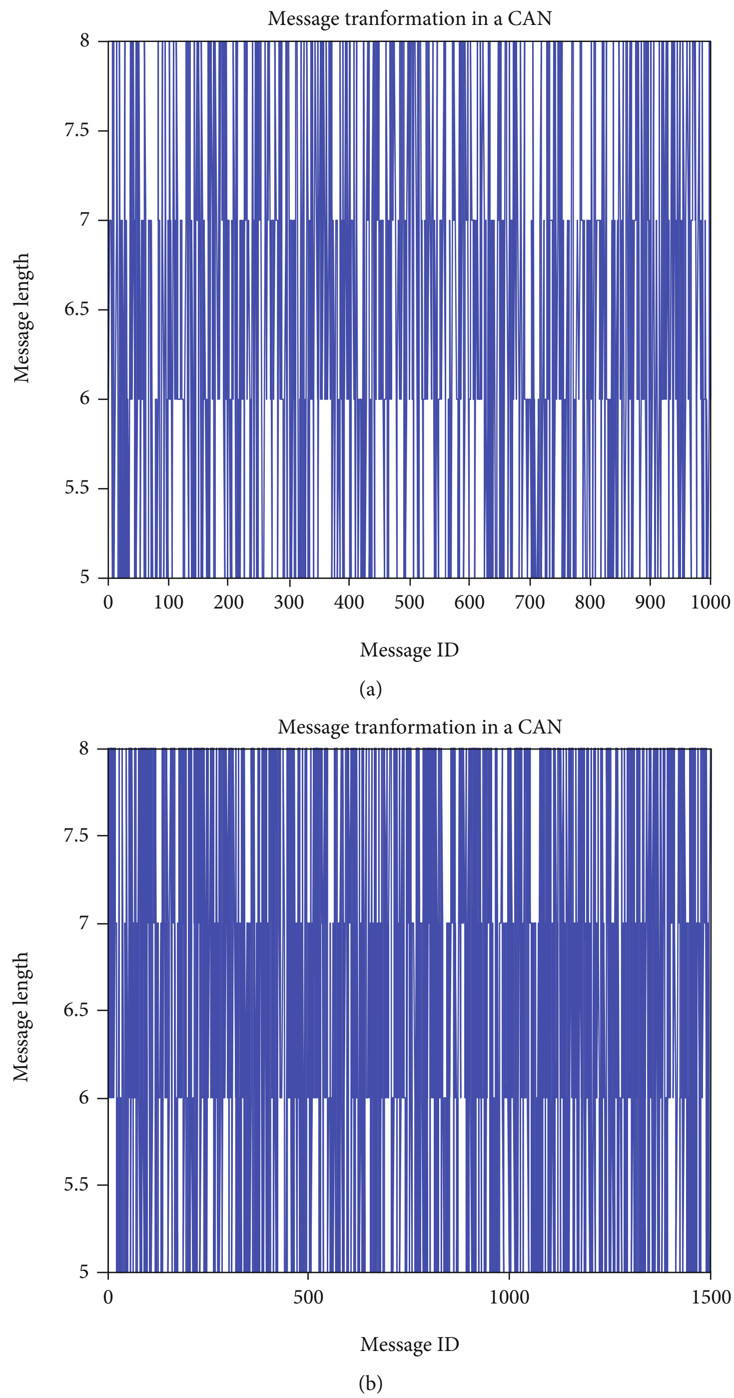


Figure 7: (a) Message length = 5 – 8 and ID = 1000 nos and (b) message length = 5 – 8 and ID = 1500 nos.

or occurrence of any noise. Due to that reason, the message may need to be re-transmitted which is denoted by RT_i for message i. So, It can be expressed as:

$$RT_i = \lceil S_i * PR_i \rceil \quad (3)$$

Where the total frames set defines as S_i of a message i, and the percentage of need for message re-transmission size can be defined by PR_i. In the case of non-complex data, PR_i=0 and for other types of data, it is expressed as PR_i>0. Then the worst-case communication time (WC_i) can be calculated for message i without any error situation and can be expressed as

$$WC_i = S_i * p_k * \tau_{bit} \quad (4)$$

Here, when the bits are stuffed, then the worst case data packet size is denoted as p_k.

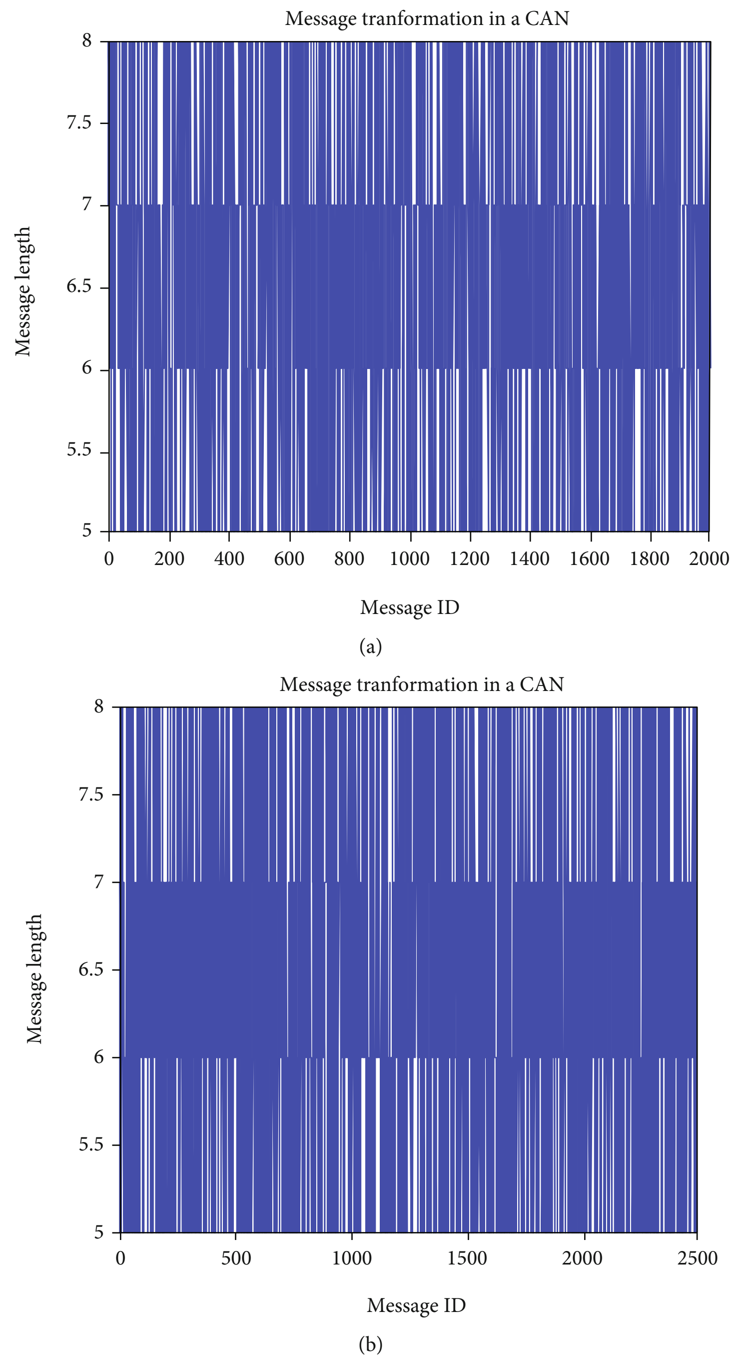


Figure 8: (a) Message length = 5 − 8 and ID = 2000 nos and (b) message length = 5 − 8 and ID = 2500 nos.

4.3. Phase Communication Time and Optimal Action of Message Frame. The instant jitter function can be expressed as $j_{i,m} = P_{i,m} - (m * TS_i + \phi_i)$, and according to that, system the jitter sum can be given as

$$J_S = \sum_{n=1}^{t} \sum_{n=1}^{t} j_{i,m} \tag{5}$$

where $j_{i,m}$ denotes the instant jitter of the $k^t h$ data frame of the sensor node m, the beginning time of the communication is denoted as $P_{i,m}, \phi_i$ is the start-up phasing of the system and the communication time interval of sensor node i is expressed as TS_i. So the alteration between the predicted starting and actual time of communication is expressed by the expression $P_{i,m} - (m * TS_i + \phi_i)$. The fitness can be calculated for optimal action of the system as

$$F(t) = J_S. \tag{6}$$

When a number of messages is used for transmission, then the crossover condition occurs and due to that,

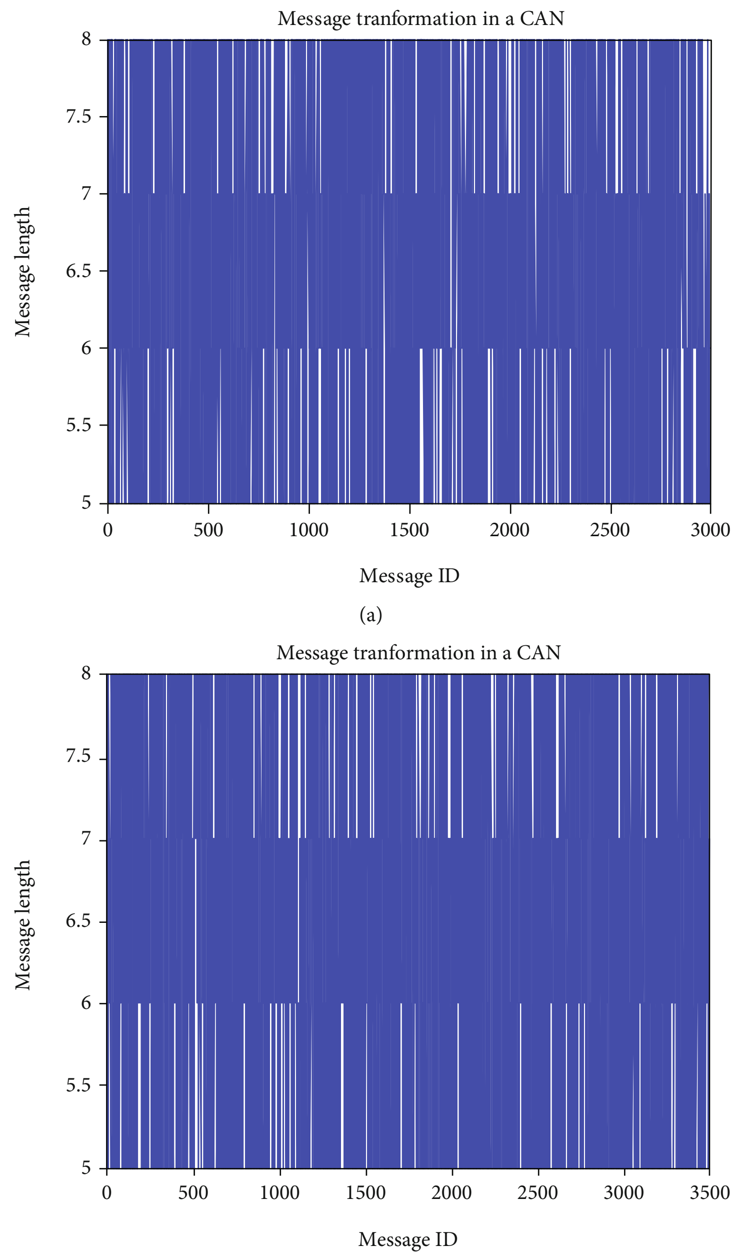


Figure 9: (a) Message length = 5 – 8 and ID = 3000 nos and (b) message length = 5 – 8 and ID = 3500 nos.

messages are queued according to their priorities. The successors are formed during processing by crossover action. It requires optimal scheduling; thus, we can choose a crossover probability of 1.0. Again, this process has improved the optimization by using efficient transformation probability as follows:

$$P_{trans} = \begin{cases} \frac{0.1(f_m - f)}{f_m - f_a}, & f \geq f_a, \\ 0.2, & f \leq f_a. \end{cases} \quad (7)$$

4.4. Message Analysis in CAN Protocol. The various level of ECUs is accessed in automotive applications to transfer signals as a form of message for steering of wheel speeds, gear selection and position of all controlled nodes of vehicles, measured through the CAN [47]. There can be more than 2500 separate signals in a high-end car, each essentially substituting an isolated connection in a conventional point-to-point connection unit [51].

These signals are used to read the location of a foot-brake; when it is pressed, the back-light section can be finding changes in signal to on brake light to avoid collision by ECU of IoT section of the CAN bus. When the messages

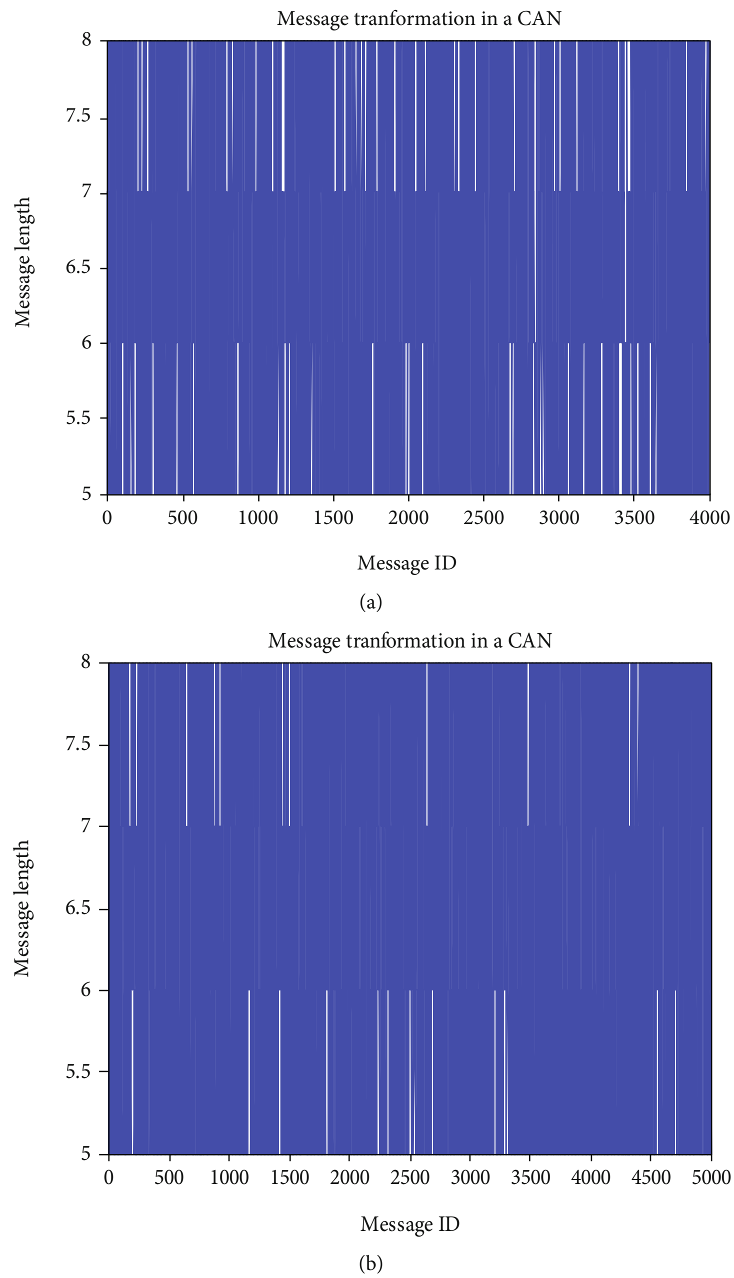


Figure 10: (a) Message length =5-8 and ID =4000 nos and (b) message length =5-8 and ID =5000 nos.

of CAN are linked with the deadline, then it responds within their time constraints as often as once every five milliseconds due to the stability and engine control system of an automobile.

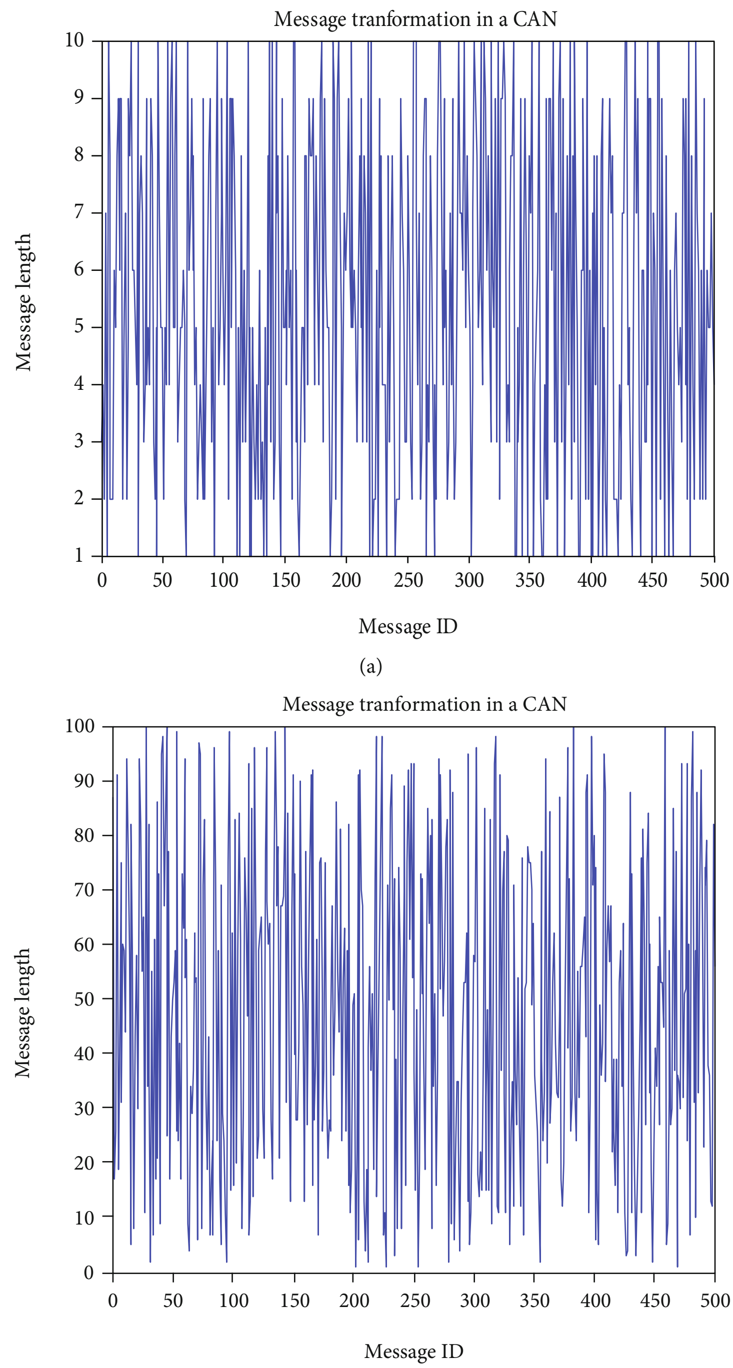
4.4.1. Message Formats. The restricted data context is discussed in four message formats such as message frame, isolated node, error bit, and excess load bit [52]. A data frame starts with the begin-of-frame bit (BOF), 11-bit ID, and the distant transfer request (DTR) bit [53] [54]. The area of arbitration forms the ID and the DTR bit.

The control field contains six bits that also specify the length of bytes in the data field, which can range from 0 to 8 bytes. Whereas a CRC bit is used to verify whether the bit sequence has been modified or not in the data field. The transmitter uses the 2-bit acknowledgment field (ACK) to obtain correct frame recognition from of receiver. The end of a message frame signal is denoted in the 7-bit end of frame (EOF) which is expanded to a twenty-nine-bit ID recipient. A 21-bit extended database framework is also available.

(i) Response time calculations under normal case

(ii) Graph of maximum achievable utilization versus T_f

(iii) Graphs of response time of any message versus T_f



(b)

Figure 11: (a) Message length = 1-10 and ID =500 nos and (b) message length = 1-100 and ID =500 nos.

(iv) Worst case tolerable value for T_f

Suppose for every first frame, message priorities are already assigned; that is Π_i, 1 is assigned for each $1 \leq i \leq m$ to first frame. In the absence of general losses, it can be assumed that messages are organized accordingly: that is $i < j$ implies that $\pi_i, 1 < \pi_j, 1$. If all subsequent frames are assigned priority π_i,2 n_i, assume that WCRT(i, π_i,2n_i) will use the methods shown above to find the worst-case response time RT_i designed for message i. In order to determine whether a feasible priority assignment exists, one can then use the algorithm given below in Algorithm 1.

This algorithm is optimal in that it always identifies one of the priorities for the two levels allocated. This algorithm starts with the lowest priority message frame, and then the worst time complexity is O(n2). The algorithm is proposed for analyzing the response time of the priority of messages from message ID and length in the CAN network. However, this algorithm evaluates response time based on message ID and length.

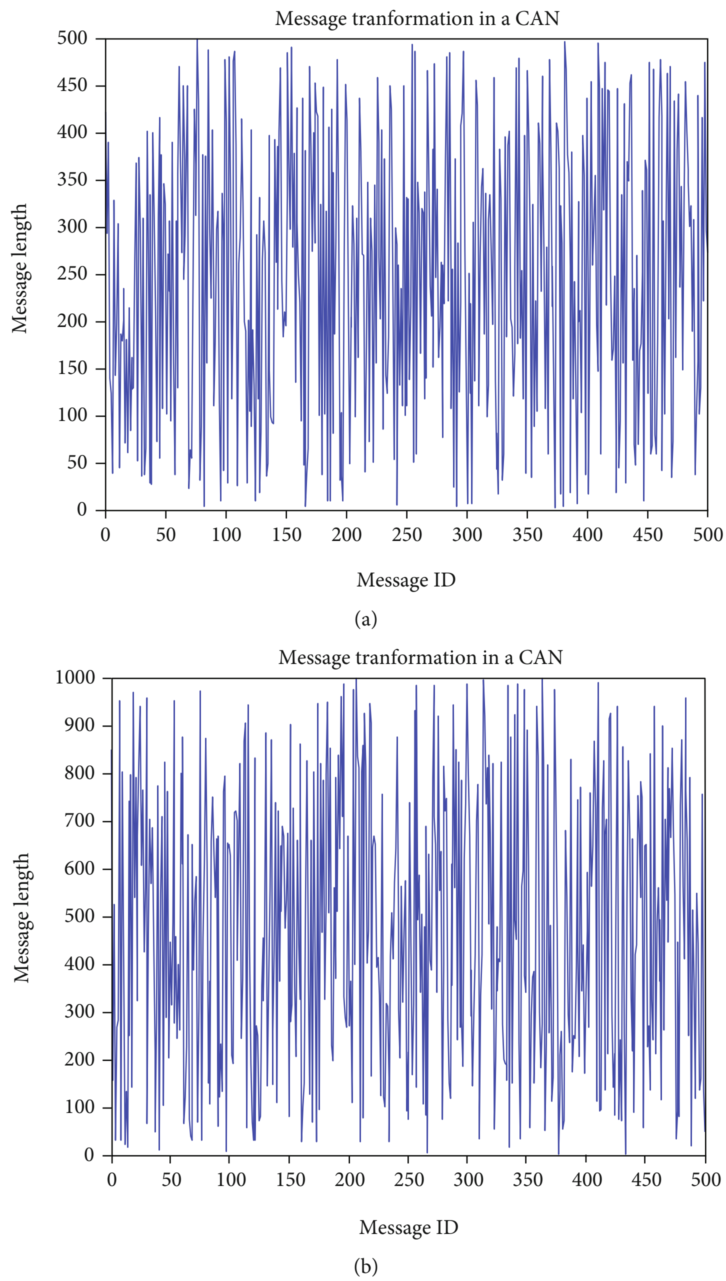


Figure 12: (a) Message length = 1-500 and ID =500 nos and (b) message length = 1-1000 and ID =500 nos.

4.5. Components Description. Different types of hardware components are required to design this proposed system:

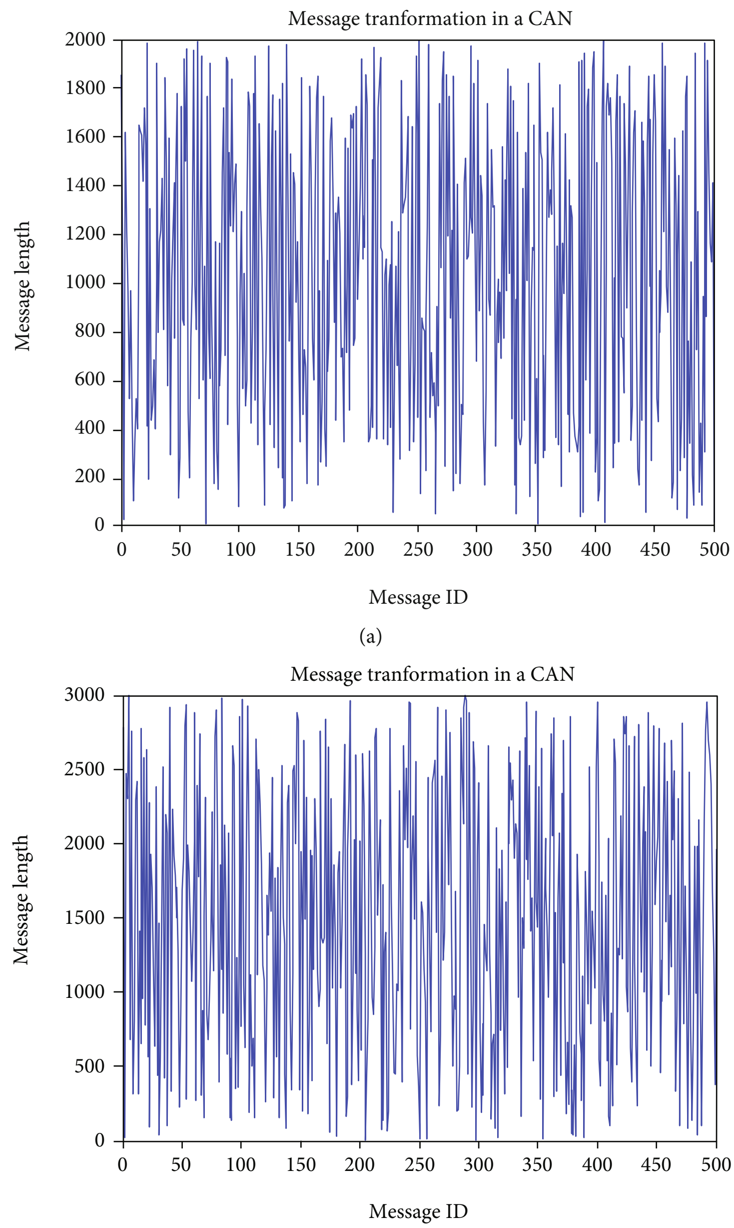
4.5.1. Arduino UNO R3 Controller. It is an open-source and IDE microcontroller that controls every movement of sensor nodes and other system network nodes [55] [56]. The C or C ++ language are simply used for programming.

4.5.2. HC-SR04 Ultrasonic Ranging Module. The sensor module is typically used in the 2 cm-400 cm range to measure the distance of the obstacle [57]. Thus, the angle of 15 degrees with a voltage of 5 V dc is made.

4.5.3. L298N Dual Bridge Motor Driver Module. L298N is a driver circuit with two inputs that makes the system to be independently enabled or disabled and the motor movement can also be controlled [58] [59]. In this case, the pulse of PWM is used to set the service period for signalling.

4.5.4. DC Motors. DC motor is accompanied by the two 150 rpm DC motors, which needs 12 V of voltage and 1-2 Amp current to start moving from right to left.

4.5.5. SG 90 Microservo Motor. It is a very lightweight server motor with high strength, which can rotate easily about 180



(a)

(b)

Figure 13: (a) Message length = 1-2000 and ID =500 nos and (b) message length = 1-3000 and ID =500 nos.

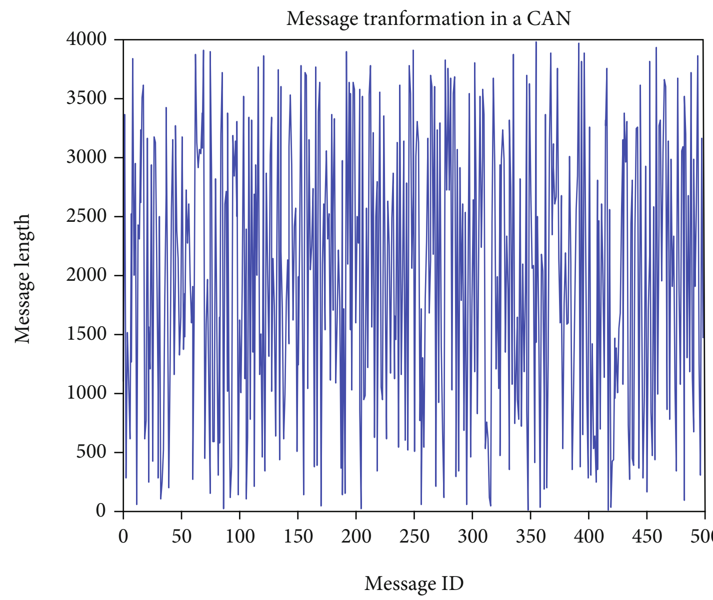
degrees (90 per path) [60] [61]. However, the movement is regulated through servo code, hardware and library.

5. Simulation Setup and Results Discussion

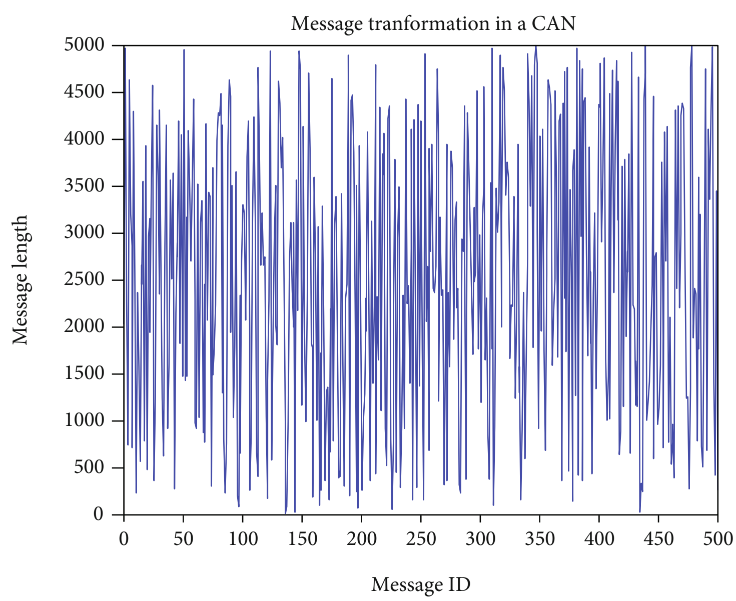
The intelligent self-employed vehicle is moving forward, which calculates the distance of an obstacle automatically. When an obstacle is detected within 20 cm via an ultrasonic sensor, the message frames obtained are transmitted to the controller. The CAN protocol code is received by the controller, which instructs the command to regulate motor movement from left to right and back. The collision avoidance algorithm was successfully implemented in order to reduce the problem. The front and top view of the Smart Autonomous Collision Avoidance Vehicle is shown in Figures 5(a) and 5(b).

5.1. Response Time Analysis. One of the types of field bus control devices used in networking is CAN. It is a protocol system based on a packet. Communication can be accomplished using the CAN protocol between different devices. The CAN bus is used to control the unit of transmission and receiving unit, which is mainly implemented due to low costs. The CAN multi-master node cannot simultaneously be transmitting and accepting messages that consist of a message ID as well as the message frame is communicated consecutively to the bus.

The CAN carriers detect multiple access protocols with collision detection and message priority arbitration, and



(a)



(b)

FIGURE 14: (a) Message length = 1-4000 and ID =500 nos and (b) message length = 1-5000 and ID =500 nos.

there are two types of protocols used in it. For flow control, the technique is used to confirm sensor data integrity, the cyclic redundancy checks (CRC) for an error control mechanism, which also manages the remote frames and the overload frames. CAN-based Database file (.dbc) is taken from the website CAN RT.dbc. It includes numerous attributes such as Name, Statement, ID, Duration, Signals, and Extended. We just take value for message length and message ID in this document which is shown in Table 1. So, these parameters observe the response of messages in the field of communication within the CAN network. After checking various data numbers, we get the following results, which are shown below. Figure 6 shows (a) message length = 5 − 8 and ID = 100 nos and (b) message length = 5 − 8 and ID = 500 nos.

5.1.1. Result for Constant Message Length and Variable Message ID. Figure 7 shows (a) message length = 5 − 8 and ID = 1000 nos and (b) message length = 5 − 8 and ID = 1500 nos. Figure 8 shows (a) message length = 5 − 8 and ID = 2000 nos and (b) message length = 5 − 8 and ID = 2500 nos. Figure 9 shows (a) message length = 5 − 8 and ID = 3000 nos and (b) message length = 5 − 8 and ID = 3500 nos. Figure 10 shows (a) message length = 5 − 8 and ID = 4000 nos and (b) message length = 5 − 8 and ID = 5000 nos. In the simulation result, we got some figures which represent different outcome from the use of CAN RT.dbc. Figures 6–10 represent that the message transformation is performed in the CAN network of two classes of inputs generated by taking fixed values for message length 5-8 and random values for message ID like 100 nos, 500 nos, 1000 nos, 1500 nos, 2000 nos, 2500 nos, 3000 nos, 3500 nos, 4000 nos, 5000 nos, respectively. When the simulation of the above network is performed by changing of message ID parameter, which shows response time (located through the white line) that varies through the process. The above analysis shows that the response time is more delayed by increasing message ID numbers. Similarly, the message is highly responded to at fewer messages ID which is shown in the above figures.

5.1.2. Result for Constant Message ID and Variable Message Length. We have some figures in the simulation outcome that show different outcomes from the use of CAN RT.dbc. In the CAN network, figures reflect the message conversion of two groups of inputs created by taking fixed values for message ID 500 numbers and random values for message length such as 1-10, 1-100, 1-500, 1-1000, 1-2000, 1-3000, 1-4000, 1-5000. When this network simulation is performed by changing the message length parameter with fixed message length and it shows no more changes in the response time of the process in the above figures. Figure 11 shows (a) message length = $1-10$ and ID = 500 nos and (b) message length = $1-100$ and ID = 500 nos. Figure 12 shows (a) message length = $1-500$ and ID = 500 nos and (b) message length = $1-1000$ and ID = 500 nos. Figure 13 shows (a) message length = $1-2000$ and ID = 500 nos and (b) message length = $1-3000$ and ID = 500 nos. Figure 14 shows (a) message length = $1-4000$ and ID = 500 nos and (b) message length = $1-5000$ and ID = 500 nos.

5.2. Discussion. In a practical scenario, several attributes such as Name, Statement, ID, Extended, Length, Signals have to be considered from the collected database. If all these attributes are combined, we can get an unschedulable framework in certain situations. Therefore, some of the attributes already in the database, such as Name, Statement, Expanded, and Signals, need to be removed. For communication, some powerful attributes such as ID and length have to be considered for a particular message.

We might easily get a very negative analysis to find errors if there are many such data and we compose them. We have shown from the simulation outcome that there was no difference in the simulation outcome for different message length values by holding the message ID unchanged. In other words, we can assume that message length variance has less influence than message ID. We take the message ID from 1 to 5000 in this paper and check the network conjunction result. We also shift the message's duration from 1 to 5000 and examine the transformation impact of this change in values.

The existing approaches tested in a limited range of message length and message ID values which is not provided a clear idea about the performance of the CAN network. But this paper is taking 1-5000 numbers values for message length and message ID. According to the proposed experiment is evaluating performance in two ways like (i) constant message length with variance number of values (1-5000) for message ID and (ii) constant message ID with variance number of values (1-5000) for message length. Thus the performance is varied on the case of constant message length with variance number of values (1-5000) for message ID, but the consistent performance is evaluated on the case of constant message ID with variance number of values (1-5000) for message length.

6. Conclusion and Future Scope

While several solution algorithms and concepts have been developed over several years to solve conflict issues using CAN communication on the vehicle network, there have been few attempts to develop a solution to the handling of errors. This is a comparison-based analysis of the variable message ID and constant message length figures that we concluded that if we send no more messages at a time, then there would be more conflict on the network. The recipient will not get the real message due to conflict, and there will also be a risk of receiving more than one message at a single node. The CAN protocol is used to provide a secure and robust serial communication bus from sensor nodes to the control unit of an automated system. When the sensor node of the IoT module is received, a message frame can be transmitted to the destination node that can be responded to in time. The phase of communication time and the optimal action of the message frame are utilised to build a flexible format for transmission of a frame from sender end to receiver end, which implies that a system node can receive a message frame and respond to it via an acknowledged frame bit. The proposed scheme achieves high precision, determining the location of an obstacle and then monitoring the impact of collision time.

More number of experiments could be carried out and future directions are:

(a) To assess the efficiency of the algorithms, a large number of experiments with more tasks have to be tested

(b) Secondly, it is important to evaluate large-size problem cases using periodic preemptive tasks

A new automatically moving algorithm between the EDF algorithm and the ACO scheduling algorithm should be developed in the future to work with overloaded conditions.

Data Availability

The IoT data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors of this manuscript declared that they do not have any conflict of interest.

Acknowledgments

This paper was partially supported by the Ministry of Science and Higher Education of the Russian Federation (Assignment No. 075-02-2021-1748) and Device Development Programme (DST/TDT/DDP-38/2021), by the Department of Science Technology, Ministry of Science and Technology, Government of India.

References

[1] G. Li, W. Lai, X. Sui et al., "Influence of traffic congestion on driver behavior in post-congestion driving," *Accident Analysis & Prevention*, vol. 141, p. 105508, 2020.

[2] P. A. Hancock, T. Kajaks, J. K. Caird et al., "Challenges to human drivers in increasingly automated vehicles," *Human Factors*, vol. 62, no. 2, pp. 310–328, 2020.

[3] U. Kiencke, S. Dais, and M. Litschel, "Automotive serial controller area network," *SAE Transactions*, vol. 95, pp. 823–828, 1986.

[4] Y.-J. Kim, H.-Y. Lee, and J.-G. Chung, "4-bit data arrangement algorithm for can compression," in *2018 International SoC Design Conference (ISOCC)*, pp. 216-217, Daegu, Korea (South), 2018.

[5] S. Dekanic, R. Grbic, T. Maruna, and I. Kolak, "Integration of can bus drivers and uds on aurix platform," in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, pp. 39–42, Novi Sad, Serbia, 2018.

[6] I. Gonzalez and A. J. Calderon, "Integration of open source hardware arduino platform in automation systems applied to smart grids/micro-grids," *Sustainable Energy Technologies and Assessments*, vol. 36, article 100557, 2019.

[7] G. Kornaros, O. Tomoutzoglou, D. Mbakoyiannis et al., "Towards holistic secure networking in connected vehicles through securing CAN- bus communication and firmware-over-the-air updating," *Journal of Systems Architecture*, vol. 109, p. 101761, 2020.

[8] D.-S. Kim and H. Tran-Dang, "Industrial sensors and controls in communication networks," in *Computer Communications and Networks*, Springer International Publishing, Cham, 2019.

[9] H. Lu, Q. Liu, D. Tian, Y. Li, H. Kim, and S. Serikawa, "The cognitive internet of vehicles for autonomous driving," *IEEE Network*, vol. 33, no. 3, pp. 65–73, 2019.

[10] L. Chelouah, F. Semchedine, and L. Bouallouche-Medjkoune, "Localization protocols for mobile wireless sensor networks: a survey," *Computers & Electrical Engineering*, vol. 71, pp. 733–751, 2018.

[11] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[12] N. Deepa, Q.-V. Pham, D. C. Nguyen et al., "A survey on blockchain for big data: Approaches, opportunities, and future directions," 2020, https://arxiv.org/abs/2009.00858.

[13] A. L. Kun, S. Boll, and A. Schmidt, "Shifting gears: user interfaces in the age of autonomous driving," *IEEE Pervasive Computing*, vol. 15, no. 1, pp. 32–38, 2016.

[14] J. Sanchez-Garcıa, J. Garcıa-Campos, M. Arzamendia, D. G. Reina, S. Toral, and D. Gregor, "A survey on unmanned aerial and aquatic vehicle multi-hop networks: Wireless communications, evaluation tools and applications," *Computer Communications*, vol. 119, pp. 43–65, 2018.

[15] A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, "Iot-based smart alert system for drowsy driver detection," *Wireless Communications and Mobile Computing*, vol. 2021, 13 pages, 2021.

[16] IndustryJournalPro, "Forward collision avoidance radar market research report 2019-2025," 2019, 2021, https://industryjournalpro.com/forward-collision-avoidance-radar-market-research-report-2019-2025/.

[17] Hyundai, "12 Hyundai safety features that make driving today safer than ever," 2020, 2021, https://www.hyundai.com/au/en/why-hyundai/myhyundai-news/issues/2020/issue-02/12-hyundai-safetyfeatures-that-make-driving-today-safer-than-ever.

[18] T. Nolte, M. Nolin, and H. A. Hansson, "Real-time server-based communication with can," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 3, pp. 192–201, 2005.

[19] W. L. Ng, C. K. Ng, B. M. Ali, N. K. Noordin, and F. Z. Rokhani, "Review of researches in controller area networks evolution and applications," *Proceedings of the Asia-Pacific Advanced Network*, vol. 30, pp. 14–21, 2013.

[20] A. S. Shinde and V. B. Dharmadhikari, *Controller Area Network for Vehicle Automation*, CiteSeer, 2012.

[21] G. Yu, C. Zhou, and S. Huang, "A protocol for automatic node-id binding in canopen networks," *JCM*, vol. 7, no. 10, pp. 765–773, 2012.

[22] T. Vijayan, *Controller Area Network in Modern Home Automation*, CiteSeer, 2012.

[23] C.-L. Wey, C.-H. Hsu, K.-C. Chang, P.-C. Jui, and M.-T. Shiue, "Emi prevention of can-bus-based communication in battery management systems," *International Journal of Engineering & Computer Science IJECS-IJENS*, vol. 13, no. 5, pp. 6–12, 2013.

[24] K. C. Emani, K. Kam, M. Zawodniok, Y. R. Zheng, and J. Sarangapani, "Improvement of can bus performance by using error-correction codes," in *2007 IEEE Region 5 Technical Conference*, pp. 205–210, Fayetteville, AR, USA, 2007.

[25] H. A. Hansson, T. Nolte, C. Norstrom, and S. Punnekkat, "Integrating reliability and timing analysis of can-based systems," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1240–1250, 2002.

[26] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (can) schedulability analysis: refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[27] G. Xie, G. Zeng, R. Kurachi, H. Takada, R. Li, and K. Li, "Exploiting Sparsity to Accelerate Fully Connected Layers of CNN-Based Applications on Mobile SoCs," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 17, no. 2, pp. 1–25, 2018.

[28] G. Xie, G. Zeng, R. Kurachi et al., "Wcrt analysis and evaluation for sporadic message-processing tasks in multicore automotive gateways," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 2, pp. 281–294, 2019.

[29] R. Alaei, P. Moallem, and A. Bohlooli, "Statistical based algorithm for reducing bit stuffing in the controller area networks," *Microelectronics Journal*, vol. 101, p. 104794, 2020.

[30] K. Liu, E. Fridman, and L. Hetel, "Networked control systems in the presence of scheduling protocols and communication delays," *SIAM Journal on Control and Optimization*, vol. 53, no. 4, pp. 1768–1788, 2015.

[31] B. H. C. Orak, F. Y. Okay, M. Guzel, S. Murt, and S. Ozdemir, "Comparative analysis of iot communication protocols," in *2018 International symposium on networks, computers and communications (ISNCC)*, pp. 1–6, Rome, Italy, 2018.

[32] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 17–31, 2015.

[33] S. Krishnamoorthy, *Design of an ASIC chip for a Controller Area Network (CAN) protocol controller, [Ph.D. thesis]*, Texas Tech University, 2006.

[34] J. Long, "Automobile electronic control network design based on can bus," in *2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pp. 9–12, Xiamen, China, 2018.

[35] K. Pazul, *Controller area network (can) basics*, Microchip Technology Inc, 1999.

[36] K. H. Johansson, M. Torngren, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook of networked and embedded control systems*, Springer, 2005.

[37] W. E. Lawrenz, *System design tools for networked systems in cars*, SAE Technical Paper, 1990.

[38] Q. Zhu, Z. Dongmei, and S. Xunwen, "Distributed remote temperature monitoring and acquisition system based on can bus," in *2010 Prognostics and System Health Management Conference*, pp. 1–4, Macao, China, 2010.

[39] SC HPL, "Introduction to the controller area network (can)," in *Application Report SLOA101*, Texas Instruments, 2002.

[40] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys (CSUR)*, vol. 37, no. 2, pp. 164–194, 2005.

[41] M. A. C. Aung and K. P. Thant, "Detection and mitigation of wireless link layer attacks," in *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 173–178, London, UK, 2017.

[42] Y. Liu, Q. Zhang, and L. Ni, "Opportunity-based topology control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 3, pp. 405–416, 2010.

[43] S. Hasnaoui, O. Kallel, R. Kbaier, and S. B. Ahmed, "An implementation of a proposed modification of can protocol on can fieldbus controller component for supporting a dynamic priority policy," in *38th IAS Annual Meeting on Conference Record of the Industry Applications Conference, 2003*, pp. 23–31, Salt Lake City, UT, USA, 2003.

[44] S. Hong and W.-H. Kim, "Bandwidth allocation scheme in can protocol," *IEE Proceedings-Control Theory and Applications*, vol. 147, no. 1, pp. 37–44, 2000.

[45] X. Wang and W. Guo, "The design of rs232 and can protocol converter based on pic mcu," *Computer and Information Science*, vol. 2, no. 3, pp. 176–181, 2009.

[46] Q. Lin, Y. Zhang, A. Van Mieghem et al., "Design and experiment of a sun-powered smart building envelope with automatic control," *Energy and Buildings*, vol. 223, article 110173, 2020.

[47] G. I. Mary, Z. C. Alex, and L. Jenkins, "Response time analysis of messages in controller area network: a review," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 148015, 11 pages, 2013.

[48] Y. Wang and M. Saksena, "Scheduling fixed-priority tasks with preemption threshold," in *Proceedings Sixth International Conference on Real-Time Computing Systems and Applications. RTCSA'99 (Cat. No. PR00306)*, pp. 328–335, Hongkong, 1999.

[49] J. Van Waes, J. Lannoo, A. Degraeve, D. Vanoost, D. Pissoort, and J. Boydens, "Effectiveness of cyclic redundancy checks under harsh electromagnetic disturbances," in *2017 International Symposium on Electromagnetic Compatibility-EMC EUROPE*, pp. 1–6, Angers, France, 2017.

[50] G. Leen and D. Heffernan, "Ttcan: a new time-triggered controller area network," *Microprocessors and Microsystems*, vol. 26, no. 2, pp. 77–94, 2002.

[51] K. Tindell, A. Burns, and A. J. Wellings, "Calculating controller area network (can) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163–1169, 1995.

[52] R. Sato and S. Fukumoto, "Response-time analysis for controller area networks with randomly occurring messages," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 3893–3902, 2020.

[53] S. Mubeen, J. Maki-Turja, and M. Sjodin, "Extending Worst Case Response-Time Analysis for Mixed Messages in Controller Area Network With Priority and FIFO Queues," *IEEE Access*, vol. 2, pp. 365–380, 2014.

[54] G. Cena, I. C. Bertolotti, T. Hu, and A. Valenzano, "Can with extensible in-frame reply: protocol definition and prototype implementation," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2436–2446, 2017.

[55] N. S. Kumar, B. Vuayalakshmi, R. J. Prarthana, and A. Shankar, "Iot based smart garbage alert system using arduino uno," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 1028–1034, Singapore, 2016.

[56] A. K. Biswal, D. Singh, and B. K. Pattanayak, "Iot-based voice-controlled energy-efficient intelligent traffic and street light monitoring system," in *Green Technology for Smart City and Society*, Springer, 2021.

[57] V. Zhmud, N. Kondratiev, K. Kuznetsov, V. Trubin, and L. Dimitrov, "Application of ultrasonic sensor for measuring distances in robotics," *Journal of Physics: Conference Series*, vol. 1015, article 032189, 2018.

[58] M. H. Y. Yumei, "High-precision servo controller of dc micromotor based on dspic30f4011," *Electronic Measurement Technology*, vol. 10, 2010.

[59] M. M. Maung, M. M. Latt, and C. M. Nwe, "Dc motor angular position control using pid controller with friction compensation," *International Journal of Scientific and Research Publications*, vol. 8, no. 11, p. 149, 2018.

[60] S. K. Mostaque and B. Karmakar, "Low cost arduino based voice controlled pick and drop service with movable robotic arm," *European Journal of Engineering and Technology Research*, vol. 1, no. 5, pp. 29–33, 2018.

[61] K. Neeraja, P. R. C. Rao, D. S. Maloji, and D. M. A. Hussain, "Implementation of security system for bank using open cv and rfid," *International Journal of Engineering & Technology*, vol. 7, no. 2-7, p. 187, 2018.