

Evaluating a New Time-Triggered CAN FD Protocol with Periodic Messages

Mahmut Tenruh
Mugla S. K. University
Electrical and Electronics Engineering
Mugla, Turkey

Periklis Charchalakis
Vetronics Research Centre (VRC)
University of Brighton, Lewes Road
Brighton, BN2 4GJ, UK

Elias Stipidis
Vetronics Research Centre (VRC)
University of Brighton, Lewes Road
Brighton, BN2 4GJ, UK

ABSTRACT

Controller Area Network (CAN) is a real-time communication bus widely accepted and used in automotive and industrial applications. CAN provides robust and low-cost solutions for real-time control systems. Its priority based bit-wise arbitration mechanism ensures that the highest priority messages have fast access to the bus. However, the lower priority messages may experience long message delays under heavy bus load and low transmission bit-rate conditions. In CAN, message transmission speeds are limited at certain bus lengths. CAN with Flexible Data-rate (CAN FD) offers a solution by applying higher transmission bit-rates for the data-phase of the message frame. However, CAN FD uses the same medium access method as CAN. This may cause CAN FD to inherit the features of CAN causing long delays for the lower priority messages. This study investigates the application of Time-Triggered CAN with Flexible Data-rate (TTCAN FD) with periodic messages. Combining the time-triggered feature of TTCAN with fast CAN FD provides an exceptional opportunity for periodic messages to transmit faster without arbitration, which is not possible with CAN or CAN FD. In order to evaluate the performance improvements, the system models with the PSA (Peugeot Societe Anonyme) message set, comprised of periodic messages, have been developed and simulated. This study also introduces a new analysis method called the performance ratio, which provides a quantitative performance comparison opportunity. The results show that the investigated new versions of TTCAN FD protocol models provide considerable performance improvements with deterministic immediate access for periodic message sets, and more comprehensive analysis is achieved with the introduced performance ratio method.

General Terms

Fieldbus, Real Time, Performance Analysis

Keywords

Message Scheduling, Controller Area Network, TDMA

1. INTRODUCTION

Electronic systems in today's vehicles are responsible for many functions such as the electric power steering, antilock braking system, communication, entertainment, and navigation. These systems are controlled by electronic control units (ECUs) comprised of mi-

crocontrollers, sensors, and actuators. These ECUs communicate through in-vehicle communication networks to accomplish their functions. Controller Area Network (CAN) is a real-time communication bus widely used in automotive and industrial applications. Its low cost and robust protocol features make it an attractive choice for real-time distributed control systems [1]. CAN uses a priority based bit-wise arbitration mechanism as the medium access control (MAC) method [2, 3, 4]. This mechanism provides fast access to the bus for high priority messages. However, the lower priority messages may experience extensive access delays under heavy bus load and low transmission bit-rate conditions [5, 6, 7].

CAN uses limited transmission bit-rates at certain bus lengths, maximum 1 Mbps at 40 m, due to the bit-wise arbitration mechanism and acknowledgement method. In order to increase the bandwidth capacity, the CAN with Flexible Data-rate (CAN FD) protocol has been introduced [8]. CAN FD improves the protocol capability by increasing the transmission speed, and by extending the data payload size. CAN FD aims a transition from CAN with minimum hardware and software changes [9]. Therefore, CAN FD inherits similar characteristics to CAN which may cause long message delays for lower priority messages.

Time-triggered networks, such as Time Triggered Protocol (TTP) and FlexRay [10, 11, 12], provide deterministic communication environment. In order to provide a deterministic access feature for CAN systems, the Time-Triggered CAN (TTCAN) protocol has been introduced [13, 14, 15, 16]. However, TTCAN uses the traditional CAN messages with limited CAN transmission bit-rates. In order to combine the deterministic feature of time-triggered structure with the fast transmission speed of CAN FD, the Time-Triggered CAN FD (TTCAN FD) protocol can be used. This protocol provides an opportunity to transmit the periodic CAN FD message frames without arbitration mechanism due to the pre-allocated message scheduling of TTCAN.

This paper introduces a method to develop a new version of TTCAN FD protocol which transmits periodic messages faster using the arbitration-free feature of the pre-allocated exclusive-time message windows. The new protocol version apply an approach which makes the transmission speed of the periodic message frames independent from the limitations imposed by the arbitration process, which is not possible in CAN or CAN FD. This study presents the implementation of periodic messages with TTCAN FD protocol in three development stages: TTCAN FD1 version as the existing TTCAN FD, and TTCAN FD2, and the proposed TTCAN FD3 versions. Each stage provides gradually faster

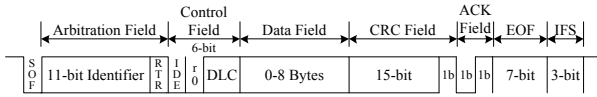


Fig. 1. This is example of the image in a column.

message transmission by utilising the arbitration-free feature of the time-triggered access. The study investigates the comparative performance of the protocol with a message set comprised of periodic messages. In order to investigate the performance of the protocol with periodic messages, the system models with the PSA message set [17, 18] have been developed and simulation results have been analysed. This study also introduces a new analysis method called performance ratio which provides a quantitative performance comparison opportunity together with the graphical analysis. The results show that the presented new protocol versions provide an opportunity to transmit periodic messages with higher performance improvements, and the introduced method offers more comprehensive performance analysis.

The paper is organised as follows. Section 2 provides an overview of the CAN, CAN FD, and TTCAN protocols. Section 3 explains the development of the TTCAN FD1, TTCAN FD2 protocols, and introduces the TTCAN FD3 protocol. Section 4 gives details about the TTCAN FD matrix cycle construction and system modelling with PSA message set. Section 5 provides the performance analysis of simulation results with the performance ratio method. Section 6 concludes the paper with final remarks.

2. CAN PROTOCOL BACKGROUND AND PRELIMINARIES

2.1 CAN protocol

CAN is a prominent real-time communication protocol used in automotive and industrial applications. Figure 1 shows the standard CAN protocol frame structure with 11-bit identifier. The protocol also provides the extended 29-bit identifier version. However, this study focuses on the standard message frames with 11-bit identifier. CAN uses the carrier sense multiple access (CSMA) with non-destructive bit-wise arbitration method for medium access. In this access method, the message with the lowest binary value has the highest priority. Every node has the right to send a message whenever the bus is idle. If two or more nodes start transmission at the same time, the highest priority message in the arbitration is sent first, and the lower priority messages are transmitted in following arbitration sessions according to their priorities [2, 3]. This method provides fast message transmission for high priority messages. However, the lower priority messages may experience extensive delays under heavy bus load and low transmission bit-rate conditions. The worst-case response time R_m for message m_m can be expressed as

$$R_m = J_m + W_m + C_m \quad (1)$$

where J_m is the queuing jitter, which gives the latest queuing time of message m on the host CPU. W_m includes the queuing delay due to the higher priority messages and the blocking time due to a lower priority message that may have already taken the bus since the protocol is non-preemptive. C_m is the longest time taken to transmit message m_m on the bus, including the time taken for frame overheads, payload, and stuff bits [5, 6, 7].

The CAN protocol applies bit-stuffing in order to provide enough edges for re-synchronisation during message transmission, in

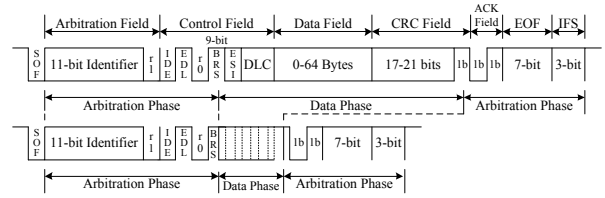


Fig. 2. Message frame for CAN FD.

which an opposite bit level is inserted after every five consecutive same level bit transmission. Therefore, the message transmission time C_m with the worst-case bit-stuffing for a standard CAN frame can be computed as

$$C_m = [47 + 8s_m + \lfloor (34 + 8s_m)/4 \rfloor] \times \tau_a \quad (2)$$

where s_m is the payload size in bytes, and 47 is the fixed form size of a standard CAN frame without payload. The floor operator $\lfloor \cdot \rfloor$ gives the number of stuff bits for the 34-bit part of frame subject to bit stuffing and payload field, where the stuff bits are also subject to bit stuffing causing the divisor to be 4. τ_a is the bus bit-time, 1 μ s if the bit-rate is 1 Mbps. Eq. (2) can be simplified as

$$C_m = (55 + 10s_m)\tau_a \quad (3)$$

2.2 CAN FD protocol

In order to overcome the bandwidth and payload limitations of CAN, the CAN FD protocol has been introduced [8, 9]. The CAN FD protocol divides a message frame into two phases as the arbitration-phase and the data-phase. The protocol frame with 11-bit identifier can be seen in Figure 2. The CAN FD protocol improves the system capability, firstly, by increasing the transmission bit-rate, secondly, by increasing the payload size. From the Start Of Frame (SOF) bit to the Bit-Rate Switch (BRS) bit, the standard CAN protocol bit-rates up to 1 Mbps are used. If the BRS bit is sampled recessive, the transmission speed is switched to a higher bit-rate for the data-phase, and the bit-rate is switched back to the arbitration-phase bit-rate at the sample point of the CRC delimiter bit. Unlike the CAN protocol, the bit-rate for the data-phase of CAN FD is not limited by the bit-wise arbitration mechanism. Therefore, up to 8 Mbps, or even 15 Mbps experimental data-phase bit-rates are possible to achieve [9]. The payload size, indicated in the Data Length Code field (DLC), is also extended from 8 bytes maximum CAN payload size to 64 bytes in CAN FD.

CAN FD also uses the bit-stuffing mechanism in the same way as CAN for the arbitration-phase and payload field. However, the stuff bits are located at fixed positions in the CRC field in order to improve the error detection capability of the protocol [9, 19]. Since the CAN FD protocol uses two different transmission bit-rates, and the CRC bit stuffing is also different, the message transmission duration can be expressed as

$$C_m = (29 + \lfloor s_a/4 \rfloor)\tau_a + (28 + 8s_m + (8s_m/4) + CRC)\tau_d \quad (4)$$

where 29 is the number of bits in the arbitration-phase including the BRS bit and excluding the CRC delimiter bit [20], and s_a is the number of bits subject to bit-stuffing in the first arbitration-phase, where the second arbitration-phase is not subject to bit-stuffing. The CRC field includes all the CRC bits and additional stuff bits, 17 CRC bits and 4 stuff bits for the payload sizes up to 16 bytes,

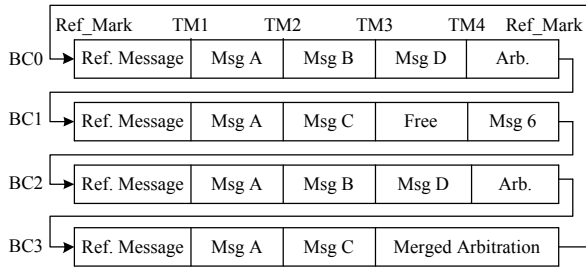


Fig. 3. An example of matrix cycle.

and 21 CRC bits and 5 stuff bits for the payload sizes more than 16 bytes [3, 20]. τ_a and τ_d are the arbitration-phase and data-phase bit-times, respectively.

The Extended Data Length (EDL) bit, r0 bit, and BRS bit have 1, 0, and 1 values, respectively. Therefore, 13 bits remain for bit stuffing in the first arbitration-phase, which can produce a maximum of 3 stuff bits, and the total number of bits in the arbitration-phase becomes 32 [20]. As a result, Eq. (4), which gives the message transmission time with the worst-case bit-stuffing, can be simplified as

$$C_m = 32\tau_a + (28 + 10s_m + CRC)\tau_d \quad (5)$$

2.3 TTCAN protocol

The CAN protocol is event-triggered, where the messages are transmitted asynchronously with the priority based arbitration mechanism which may result in long message delays. TTCAN provides a deterministic access feature for the system with time-triggered transmission [13]. TTCAN corresponds to the Session Layer of the Open System Interconnection (OSI) model [14], whereas the CAN protocol is described in the Physical and Data Link Layers. Time-triggered scheduling is realised with a system matrix (SM) also known as matrix cycle (MC), which is repeated cyclically. An SM is composed of Basic Cycle (BC) rows and Transmission Columns (TCs). Each BC starts with a Reference Message transmitted by the Time Master node which synchronises message transmissions. The synchronisation is realised at two levels. Level 1 provides synchronisation for the time-triggered scheduling and is based on the Network Time Unit (NTU) related to the network bit-time. Level 2 is based on a high precision global time and allows TTCAN to synchronise and interface to the other networks. Each TC starts with a Time Mark (TM), and the Reference Message starts with the Reference Mark. TCs together with BCs construct the message transmission windows, and the message size in the first BC determines the corresponding TC column size. The message windows are organised in three types as the exclusive, arbitrating, and free windows. Each exclusive window is reserved for a specific periodic message, whereas arbitrating-time windows are scheduled for sporadic messages where the transmission is realised with the usual priority based arbitration mechanism. Free windows are reserved for future extensions. Figure 3 shows the structure of a matrix cycle. Each node has registers, called triggers, to keep Time Mark information for messages related to the node. In a time window, transmit trigger (Tx_Trigger) and receive trigger (Rx_Trigger) indicate the message to be transmitted and received, respectively. A trigger keeps the window type and the TC information of the related message.

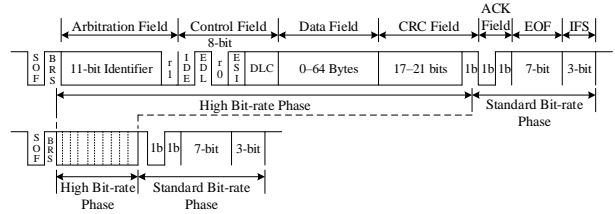


Fig. 4. Structure of TTCAN FD2 message frame.

3. TTCAN FD PROTOCOL DEVELOPMENT

The CAN protocol applies the priority based bit-wise arbitration mechanism to all sporadic and periodic messages in the system. The TTCAN protocol provides a deterministic bus access feature for periodic messages where all the periodic messages are transmitted without jitter in exclusive-time windows. This study presents the three-stage development method which combines the fast message transmission feature of the CAN FD protocol with the deterministic access feature of TTCAN for periodic message sets. The Time-Triggered CAN FD protocol development is realised in three stages. In the first stage, called TTCAN FD1, the CAN FD frames are assigned in the TTCAN message windows. This is the present state of the protocol which can be readily applied with some recent TTCAN controllers produced with CAN FD features [21, 22]. The second and third development stages exploit the arbitration-free feature of exclusive-time windows for periodic messages. In CAN FD, the identifier field has to be transmitted with a limited bit-rate due to the priority-based bit-wise arbitration process used for medium access. However, in TTCAN FD, the exclusive-time windows provide an opportunity for periodic messages. Since periodic messages are pre-allocated in the exclusive-time windows, there is no need for the arbitration process to access the bus. This allows a higher transmission bit-rate to be used for the arbitration-phase. Therefore, the main part of the CAN FD frame, which includes the fields from the BRS bit, reallocated just after the SOF bit for synchronisation, to the CRC-delimiter bit, can be transmitted with the fast data-phase transmission speed. This process constitutes the second stage of the protocol and called TTCAN FD2 [23]. As an important advantage of this method, the transmission speed is no longer limited by the arbitration process for periodic messages, and the main parts of the messages can be transmitted with higher speeds that the physical layer properties allow. This feature is not possible in CAN or CAN FD. Although some recent commercial products include the TTCAN architecture with CAN FD [21, 22], they do not provide the development proposed in this study. The resulting TTCAN FD2 protocol structure can be seen in Figure 4. The message transmission time for TTCAN FD2 can be computed as

$$C_m = 13\tau_a + [16 + \lfloor s_{arb}/4 \rfloor] + 28 + 8s_m + (8s_m/4) + CRC\tau_d \quad (6)$$

In Eq. (6), the 16-bit first arbitration part of the frame is included in the data-phase field, re-named as the high bit-rate phase. In this way, the fields of the frame related to actual data are transmitted with the fast data-phase bit-rate, and only the second 13-bit part of the arbitration-phase is transmitted with the low arbitration-phase bit-rate, re-named as standard bit-rate phase, to keep the acknowledgement feature and to allow time for processing of the received message. The transmission time of a TTCAN FD2 message frame can be written in a simplified way as

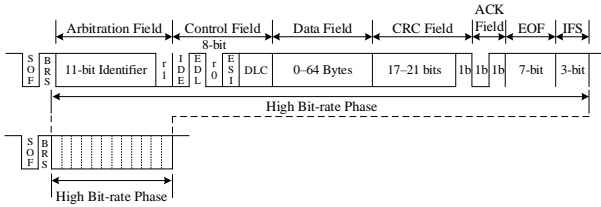


Fig. 5. Structure of TTCAN FD3 message frame.

$$C_m = 13\tau_a + (47 + 10s_m + CRC)\tau_d \quad (7)$$

This study also introduces the third development stage of the protocol, called TTCAN FD3. The structure of TTCAN FD3 message frame can be seen in Figure 5. In this stage, the whole protocol frame is transmitted with the high data-phase bit-rate. Some standard CAN protocol features, such as the bit-monitoring in one bit-time, are not applied in CAN FD in order to achieve higher bit-rates. Similarly, in TTCAN FD3, the ACK bit is also included in the high bit-rate phase. In this case, although the acknowledgement feature is not used for error detection, the CRC check and other error detection mechanisms in the protocol will provide enough degree of safety for communication for most applications. There will be enough time for message processing after the message transmission as the even distribution scheduling algorithm provides enough space between messages [24], and enough processing time can also be allocated in exclusive-time windows. In TTCAN-FD3, as the whole frame is transmitted with high transmission bit-rate, the worst-case message transmission time can be written by re-organising Eq. (7) as follows:

$$C_m = (60 + 10s_m + CRC)\tau_d \quad (8)$$

These three TTCAN FD message frame types can be used according to the features of messages and system requirements. As the messages are pre-allocated, all nodes in the system will have the matrix cycle to receive each message in its allocated time window according to the specified frame type.

From the point of hardware consideration, the first stage development, TTCAN FD1, can be readily applied with the emerging TTCAN controllers produced with CAN FD [21, 22]. However, for the TTCAN FD2 and TTCAN FD3 protocols, the proposed features should be included in the controllers. The proposed protocol versions require the bit-rate switch from the start of the frame, earlier than it is required in CAN FD. This feature can be provided without significant changes in the controller. Since the reduced matrix cycle method is used in this study, there will be enough number of triggers in each node [24, 25].

As for the synchronisation, the standard CAN FD frame applies bit-rate switching during the BRS bit. This provides the necessary conditions for the synchronisation for standard CAN FD and it is also applied in TTAN FD1 in the same way. In order to provide the same synchronisation conditions for TTCAN FD2 and TTCAN FD3, BRS bit is reallocated just after the SOF bit. As all the message types are known in pre-allocated exclusive-time windows in the matrix cycle, the BRS bit position in the frame is also known beforehand and the bit-rate switching is realised according to the described frame type.

TTCAN lacks bandwidth as it allocates the traditional CAN messages. However, the TTCAN FD protocol provides the required bandwidth as it employs the bandwidth advantage of the CAN FD

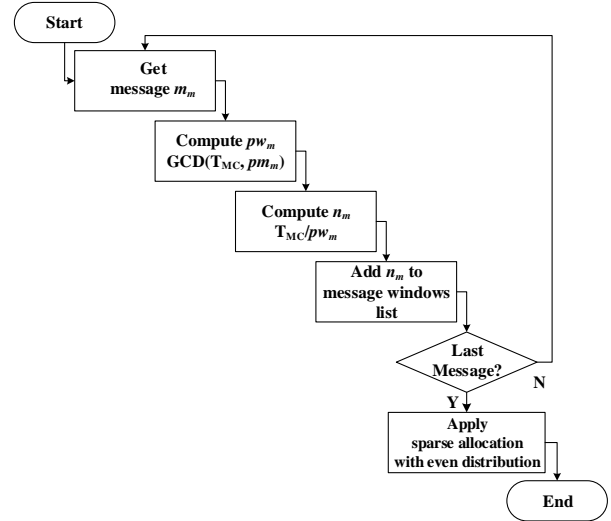


Fig. 6. Simplified message scheduling algorithm.

protocol, and the bandwidth performance is further improved with the method introduced in this study. Some recent controllers with both TTCAN and CAN FD features are making it possible to implement the TTCAN FD applications [21, 22]. In addition to the deterministic feature provided for CAN FD with TTCAN FD1 and TTCAN FD2, the proposed TTCAN FD3 version also provides a further increase in transmission speed which is crucial in meeting real-time control requirements. This feature offers further implementation opportunities in distributed real-time control systems for the proposed protocol versions. In this way, this paper offers more than simply implementing the CAN FD protocol in TTCAN. It exploits the arbitration-free feature of the pre-allocated message windows for periodic messages making it possible to transmit messages with higher bit-rates beyond arbitration limitations.

4. TTCAN FD MATRIX CYCLE AND SIMULATION MODEL DESIGN

In the matrix cycle design, the PSA (Peugeot Societe Anonyme) message set, composed of periodic messages, has been used. In Table 1, the messages are given in priority order from the highest (M1) to the lowest (M12). DLC indicates the payload size in bytes, and T is the message period in ms. In matrix cycle construction, if all periodic messages are 2^j multiples of the smallest message period pm_1 , with the condition $0 \leq j \leq 6$, the scheduling can be realised easily. In some studies, the modified PSA message set is used in order to provide simplicity in scheduling [18, 26]. In this study, the reduced matrix cycle with sparse allocation method is used in matrix cycle design. This method provides smooth scheduling with message periods any integer multiples of pm_1 . In this study, the original message set periods are used in order to demonstrate the possibility of scheduling, even with message periods not multiples of pm_1 via the reduced matrix cycle with sparse allocation method [24]. The simplified algorithm can be seen in Figure 6.

In order to construct the matrix cycle, the BC time is set equal to the smallest message period, $T_{BC} = pm_1$, which is 10 ms. The Least Common Multiple (LCM) of periodic messages can be chosen as the matrix cycle time (T_{MC}). However, the resulting BC line number may exceed the limit defined as 2^j with the condition $0 \leq j \leq 6$. Therefore, the reduced matrix cycle method is used

Table 1. PSA message set.

Message	DLC (bytes)	T (ms)
M1	8	10
M2	3	14
M3	3	20
M4	2	15
M5	5	20
M6	5	40
M7	4	15
M8	5	50
M9	4	20
M10	7	100
M11	5	50
M12	1	100

in the MC design [24, 25]. As the exclusive message size in the first BC determines the entire column size, the following messages should have the same or close message sizes to the first message in the column for the construction of the SM with the optimal configuration [25]. In this study, the optimal configuration approach has been applied in the message allocation in columns. As the approach in [25] does not consider the scheduling of messages with periods not multiple of pm_1 , the method in [24] is used for matrix cycle construction. However, a more advanced and simplified form of the algorithm is used in this study to determine the number of exclusive-time windows to be allocated in the system matrix. This algorithm provides scheduling results with less computation, reducing the amount of processing required. In the modified algorithm, the Greatest Common Divisor (GCD) function plays a key role to produce the matrix cycle. In the algorithm, first, the repetition period of an exclusive window allocated for message m_m is computed with $pw_m = GCD(T_{MC}, pm_m)$. Then, the number of the exclusive windows for message m_m is computed as $n_m = T_{MC}/pw_m$. The repetition period pw_m and number of the exclusive-time windows (n_m) for each message in the message set is computed to get the complete list of exclusive-time windows in the MC.

In the design process, the number of BCs is set to be 4, resulting in a 40 ms MC time (T_{MC}). The repetition period of exclusive windows for message period pm_1 is computed as $pw_1 = GCD(40, 10)$, which is equal to 10 ms. The number of exclusive windows allocated for the message with period pm_1 is computed as $n_1 = 40 \div 10$, which is equal to 4 exclusive-time windows. The only message in the set with period pm_1 is M1, therefore, 4 exclusive-time windows are assigned for message M1. This process is repeated for all messages in the set until the whole exclusive windows list is prepared. The messages with any integer multiples of pm_1 can be allocated in the MC smoothly with sparse allocation method. Only the messages M2, M4, and M7, whose periods are not integer multiples of pm_1 , need special attention in allocation. These messages can be allocated after the allocation of the messages whose periods are the integer multiples of pm_1 . These messages require higher numbers of exclusive-time windows n_m than the others, and the sparse allocation method provides flexibility in the allocation process. In the case of modified PSA message set, in which all the message periods are the integer multiples of pm_1 , the allocation process can be realised smoothly in a simple way. However, it is also possible to allocate the original message set with the scheduling algorithm used in this study. With this method, it is possible to meet the requirements for schedulability constraint where the real-time message deadlines are crucial especially for safety-critical applications. As the number of triggers in a TTCAN controller is limited, it

Ref	M1	M4	M2	M3	M5	M2	M7	M8	M2	M4	M10	M2	M11	M7	M2
Ref	M1	M4	M2	M12	M6	M2	M7	M8	M2	M4	M9	M2	M11	M7	M2
Ref	M1	M4	M2	M3	M5	M2	M7	M8	M2	M4	M10	M2	M11	M7	M2
Ref	M1	M4	M2	M12	Free	M2	M7	M8	M2	M4	M9	M2	M11	M7	M2

Fig. 7. Designed matrix cycle for the TTCAN FD simulation models.

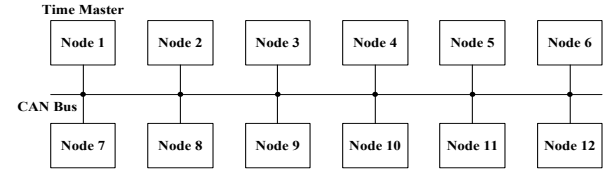


Fig. 8. Simplified representation of the system model.

is also important to meet this hardware constraint requirement. The reduced matrix method applied in this study also offers a solution for this constraint by reducing the number of required triggers. The resulting matrix cycle design can be seen in Figure 7.

This matrix cycle can be used for all three types of TTCAN FD protocol frames. TTCAN simply provides time windows for messages. Whether they are CAN or CAN FD frames, the same support is provided for all message types. Therefore, the same algorithm can be used for all TTCAN FD frame types described in this study. Since only the message transmission time is different for the TTCAN FD message types from the point of the algorithm, just the time-window sizes will be different. Therefore, it is possible to allocate described TTCAN FD frame types. TTCAN FD1 is straightforward to allocate, and also there are already existing controllers for this frame version. From the point of the algorithm, it is even more convenient to allocate the TTCAN FD2 and TTCAN FD3 frames since they require smaller time-window sizes in the matrix cycle.

This study provides a comparative performance analysis of the PSA message set based TTCAN FD systems with CAN FD systems in order to investigate the improvements obtained with the TTCAN FD protocol versions. In the model, each message signal is packed in one frame by a node. In the development of system models, the SimEvents toolbox of Matlab Simulink software package is used [27]. Figure 8 shows the simplified representation of the simulation model. The model is configured for the TTCAN FD1, TTCAN FD2, TTCAN FD3, and CAN FD systems. The time-master node is only configured in TTCAN FD models, and Level-2 synchronisation is used. In order to investigate the performance of the systems at different bus transmission speeds, 1 Mbps, 500 kbps, 250 kbps, and 125 kbps transmission bit-rates are used for the CAN FD and TTCAN FD arbitration-phase. For the CAN FD and TTCAN FD data-phase, 5 Mbps transmission bit-rate is used as it is also supported in recent controllers with CAN FD feature [21, 28].

5. PERFORMANCE ANALYSIS

In a real-time system, meeting message deadlines is a crucial issue. The worst-case response time or worst-case message delay analysis indicates the capability of the system to meet the critical real-time deadline requirements. Therefore, the worst-case message delay analysis is realised at four different bus transmission speeds and message delays are investigated with their priorities. The average message delay is also another system performance parameter investigated in this study.

As the priority based bit-wise arbitration is used in CAN FD for bus access, the system is modelled with and without offsets. The studies show that the use of offsets reduces message delays in the system [29, 30]. Without offsets, the periodic message start-times coincide and cause higher delays. However, the models without offsets are also used in this study in order to investigate the possible worst-case message delays experienced in the simulations.

5.1 The Performance ratio method

In this study, a method called performance ratio (Pr_m) is introduced for message delay analysis. In addition to the graphical performance analysis, this method also provides an opportunity for precise quantitative performance comparison of different systems. It is especially useful for the performance analysis graphs with a logarithmic scale, in which the ratio is hard to estimate. The worst-case message delay performance ratio for a message m_m can be described as the ratio of worst-case delay (Rm_m) of the message in one system to the worst-case delay of the same message in another system. The worst-case message delay performance ratio for the message m_m can be written as follows:

$$Pr_m = Rm_m(\text{System1})/Rm_m(\text{System2}) \quad (9)$$

The ratio indicates the message delay performance improvement. For example, it is possible to calculate the worst-case message delay performance improvement of TTCAN FD protocol against the CAN FD protocol for the message m_m as $Rm_m(\text{CANFD})/Rm_m(\text{TTCANFD})$. The performance ratio is computed for each message m_m in the message set. Then the mean, minimum, and maximum performance ratio values can be produced for the message set. In this way, in addition to the mean performance ratio, it is also possible to see the performance boundaries for a system with maximum and minimum ratio values. However, the average performance ratio is usually enough to have an idea about the general performance improvement of a system.

5.2 Worst-case message delay analysis

In order to investigate the worst-case message delays and the performance improvements achieved with the TTCAN FD protocols against CAN FD, the graphs in Figure 9 are analysed. As there is a big difference between the highest and lowest graph values, the logarithmic scale is used in the figure. The graphs CAN-FD max1 and CAN-FD max2 show the worst-case CAN FD message delays encountered in the simulations without and with offsets, respectively. The graph TTCAN-FD max1 shows the worst-case message delays for the time-triggered system model with the TTCAN FD1 protocol. The graph TTCAN-FD max2 shows the worst-case message delays with the TTCAN FD2 protocol. The graph TTCAN FD max3 shows the worst-case message delays for the TTCAN FD3 protocol.

From the graph CAN-FD max1 in Figure 9, it can be seen that the highest message delays are experienced with the CAN FD protocol without offsets. Since all messages are periodic and the message start-times coincide, the delays build up through the lower priority messages. The graph CAN-FD max2 shows that the use of offsets improves the delay performance for the lower priority messages where it reduces the delay performance of higher priority messages [30].

The study investigates the performance improvements achieved with the TTCAN FD1, TTCAN FD2, and TTCAN FD3 models. As can be seen in the figures, the first stage protocol, TTCAN FD1, provides considerable worst-case message delay performance im-

Table 2. TTCAN FD worst-case delay performance ratios against CAN FD without offsets.

Bit-rate	TTCAN FD1	TTCAN FD2	TTCAN FD3
1 Mbps	6.61	10.57	16.13
500 kbps	6.55	12.73	27.67
250 kbps	6.53	14.66	50.76
125 kbps	6.51	16.07	96.93

Table 3. TTCAN FD worst-case delay performance ratios against CAN FD with offsets.

Bit-rate	TTCAN FD1	TTCAN FD2	TTCAN FD3
1 Mbps	3.10	4.95	7.53
500 kbps	3.25	6.30	13.64
250 kbps	3.68	8.27	28.57
125 kbps	4.63	11.42	68.86

provement compared to the CAN FD protocol. The second stage development protocol, TTCAN FD2, provides a further performance improvement. As all the messages in PSA message set are periodic, the whole message set can be transmitted as the TTCAN FD2 protocol frames. The results achieved with the deterministic and fast transmission features of the TTCAN FD2 protocol can be seen in Figure 9. The TTCAN FD3 protocol provides even faster message transmission since the arbitration bit-rate is not used, and the whole frame is transmitted with the high transmission bit-rate. Therefore, as can be seen from the graphs, the performance difference for TTCAN FD3 becomes more noticeable from 1 Mbps through 125 kbps.

This study investigates the performance improvements with the average performance ratio values. Table 2 and Table 3 provide the worst-case message delay performance ratios (Pr_m) for the TTCAN FD models against CAN FD in order to investigate the performance improvements of the TTCAN FD1, TTCAN FD2, and TTCAN FD3 protocols respectively at four different arbitration-phase bit-rates. For example, in Table 2, the mean worst-case delay performance ratio of TTCAN FD1 against CAN FD without offsets at 1 Mbps is 6.61. That is, from the point of the worst-case message delays, the TTCAN FD1 protocol transmits messages 6.61 times faster on average than the CAN FD protocol without offsets at 1 Mbps.

Table 2 shows the mean ratio values for TTCAN FD protocols against CAN FD without offsets. The TTCAN FD1 mean performance ratio of worst-case message delay ranges between 6.51 and 6.61 against CAN FD without offsets. The minimum delay ratio, computed for the highest priority message, is 1 since CAN FD transmits the highest priority message first, and provides the same delay as TTCAN FD1 for M1. However, lower priority messages experience higher bus access delays in CAN FD. The TTCAN FD1 mean performance ratio ranges from 3.10 to 4.63 against CAN FD with offsets.

As can be seen from Table 2, the performance improvement with the TTCAN FD2 protocol is much higher than the TTCAN FD1 protocol. This is because the TTCAN FD2 protocol uses the arbitration-free feature of time-triggered access and the first arbitration-phase including the identifier field is also transmitted with the high data-phase bit-rate. The worst-case message delay performance ratio of TTCAN FD2 ranges from 10.57 to 16.07 on average against CAN FD without offsets, whereas the ratio ranges from 4.95 to 11.42 against CAN FD with offsets. From the table, it can also be seen that the worst-case delay performance is further improved with TTCAN FD3 since the whole frame is trans-

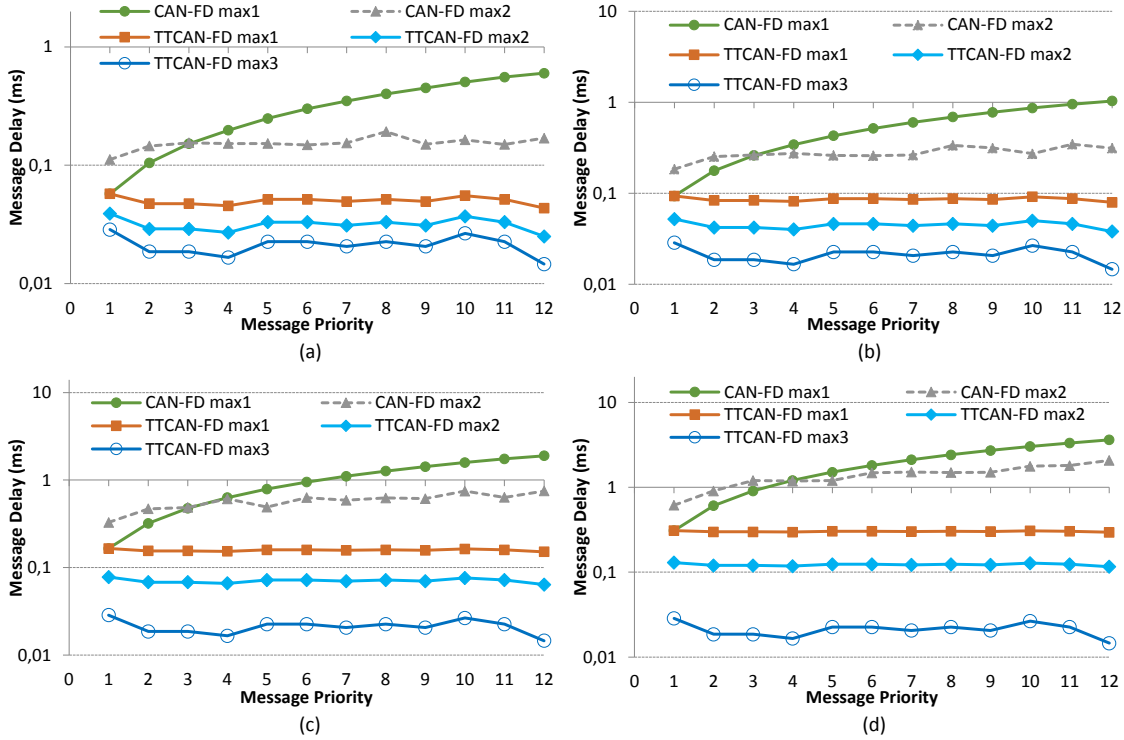


Fig. 9. Worst-case message delays at (a) 1 Mbps, (b) 500 kbps, (c) 250 kbps, (d) 125 kbps.

mitted with the high data-phase bit-rate. The worst-case delay performance ratio of TTCAN FD3 against CAN FD without offsets ranges from 16.13 to 96.93 on average, and it ranges from 7.53 to 68.86 against CAN FD with offsets. With TTCAN FD3, since the whole frame is transmitted with the high transmission bit-rate, the ratio against CAN FD at lower bit-rates becomes much higher.

Table 3 shows the worst-case message delay performance ratios of TTCAN FD protocols against CAN FD with offsets. It can be seen that the TTCAN FD1 mean performance ratio ranges from 3.10 to 4.63, whereas the ratio ranges from 4.95 to 11.42 for TTCAN FD2, and it ranges from 7.53 to 68.86 for TTCAN FD3. With TTCAN FD3, since the whole frame is transmitted with the high transmission bit-rate, the ratio against CAN FD at lower bit-rates becomes much higher.

5.3 Average message delay analysis

The average message delay graphs can be seen in Figure 10. The graphs CAN FD avrg1 and CAN FD avrg2 show the average message delay values for CAN FD systems without and with offsets, respectively. The graphs TTCAN FD avrg1, TTCAN FD avrg2, and TTCAN FD avrg3 show the average message delays for the TTCAN FD1, TTCAN FD2, and TTCAN FD3 system models, respectively.

The effect of message transmissions without offsets can be seen on CAN FD model for also average message delays in Figure 10. As all messages are periodic and most of the message start-times coincide without the use of offsets, the average message delays are high and the delay values increase through the lower priority messages due to the priority based bit-wise arbitration. The message delays for messages 2, 4, and 7 show descending values in graphs com-

Table 4. TTCAN FD average delay performance ratios against CAN FD without offsets.

Bit-rate	TTCAN FD1	TTCAN FD2	TTCAN FD3
1 Mbps	4.26	6.84	10.45
500 kbps	4.20	8.17	17.83
250 kbps	4.17	9.37	32.59
125 kbps	4.17	10.30	62.47

pared to the other messages since these messages have periods not multiples of pm_1 , and therefore have less possibility of coinciding with the other message start-times. This effect can be observed in the CAN FD average message delay graph values without offsets. The main factor in average delay for a message without offset is the priority based arbitration process and resulting queuing delay (W_m) for bus access.

However, the main factor in the average delay for a message with offset is the message transmission time (C_m), which is determined with the message size and the transmission bit-time. In the figures, the CAN FD average message delay graphs with offsets show very close delay values to TTCAN FD1, since both models use the same message frames. However, the TTCAN FD models have the advantage of guaranteed low worst-case message delays, where ideally the worst-case and average message delays have the same values due to the deterministic medium access. The average message delay graphs show that the TTCAN FD1 protocol model has better performance than the TTCAN FD2 protocol model since the first arbitration field is also transmitted in the high bit-rate phase. The TTCAN FD3 protocol has the smallest message delay values since the whole message frame is transmitted with the high bit-rate phase.

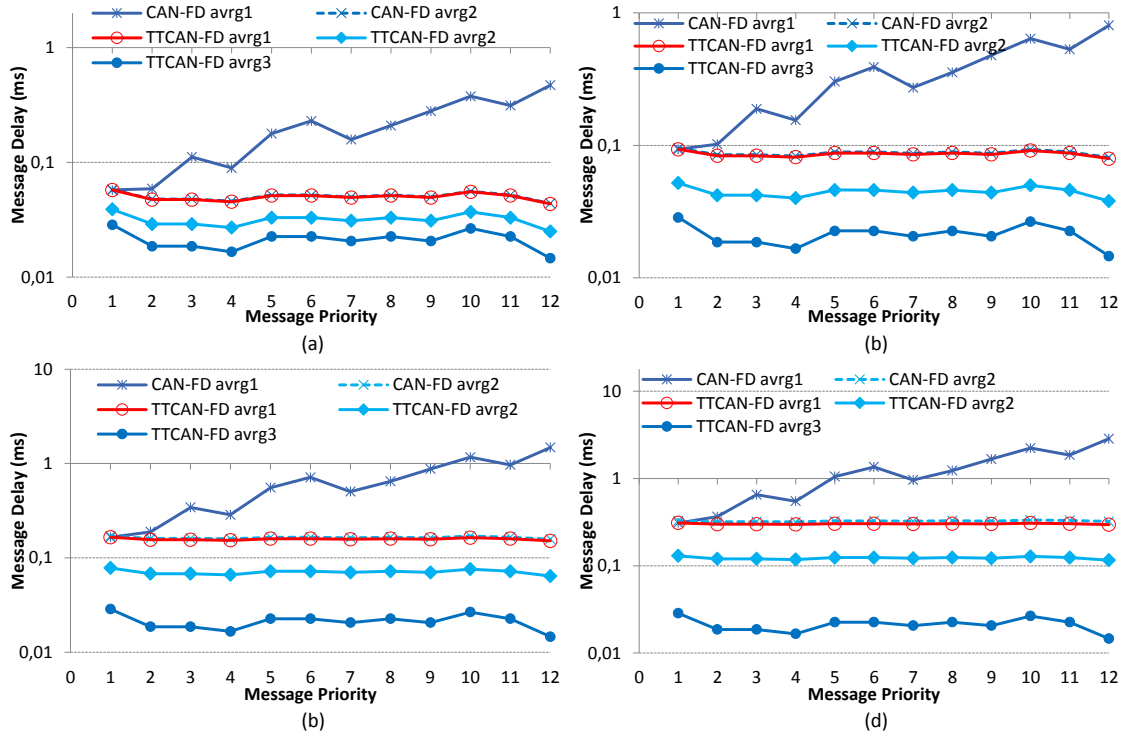


Fig. 10. Average message delays at (a) 1 Mbps, (b) 500 kbps, (c) 250 kbps, (d) 125 kbps.

Table 5. TTCAN FD average delay performance ratios against CAN FD with offsets.

Bit-rate	TTCAN FD1	TTCAN FD2	TTCAN FD3
1 Mbps	1.01	1.61	2.43
500 kbps	1.02	1.97	4.23
250 kbps	1.04	2.32	7.93
125 kbps	1.08	2.65	15.76

The performance ratios are also computed for average message delay values in order to investigate the statistical performance improvements of the TTCAN FD protocols. Table 4 and Table 5 show the average delay performance ratios for TTCAN FD against CAN FD without and with offsets respectively. As can be seen in Table 4 and Table 5, the TTCAN FD1 mean performance ratio against the CAN FD protocol model without offsets is 4.26, whereas, the ratio is just over 1 against CAN FD with offsets. Therefore, it can be considered that TTCAN FD1 and CAN FD with offsets have almost the same average message delay performances since the same message frames are used in both models.

The performance ratio of TTCAN FD2 against CAN FD without offsets ranges from 6.84 to 10.30 on average, whereas, it ranges from 1.61 to 2.65 against CAN FD with offsets. Table 8 shows that TTCAN FD3 mean performance ratio ranges from 10.45 to 62.47 against CAN FD without offsets, and from 2.43 to 15.76 against CAN FD with offsets. Tables 4 and 5 show that TTCAN FD3 mean performance ratio ranges from 10.45 to 62.47 against CAN FD without offsets, and from 2.43 to 15.76 against CAN FD with offsets. As can be seen from the tables, the TTCAN FD2 protocol has improved message delay performance ratios against the

TTCAN FD1 protocol, and TTCAN FD3 also has further performance improvements.

6. CONCLUSIONS

This study introduced a new version of the TTCAN FD protocol. The study presented the performance improvements achieved with the TTCAN FD protocol versions implemented with the time-triggered PSA message set based simulation models. The protocol models include three development stages as the existing TTCAN FD1, TTCAN FD2, and the proposed new TTCAN FD3. In this study, the performance ratio (Pr_m) method is used for message delay analysis. This method provides an opportunity for quantitative performance comparison of different systems in addition to the graphical performance analysis. In this way, the method also provides an opportunity to analyse the performance graphics quantitatively. In order to realise the comparative performance improvement analysis, the simulation models for CAN FD, TTCAN FD1, TTCAN FD2, and TTCAN FD3 were developed. The models were simulated with 1 Mbps, 500 kbps, 250 kbps, and 125 kbps arbitration-phase bit-rates. A higher bit-rate of 5 Mbps was used for the data-phase of CAN FD and TTCAN FD message frames. In the comparative analysis, the worst-case and average message delay performances of the systems were investigated.

The simulation results showed that TTCAN FD provides better worst-case message transmission compared to CAN FD. The improvement was up to 6.51 times on average for TTCAN FD1, and 16 times for TTCAN FD2, and 96 times for TTCAN FD3 compared to CAN FD worst-case message delays. TTCAN FD2 provides 2.46 times on average improved worst-case message delay performance compared to TTCAN FD1. TTCAN FD3 provides 5.93 times and

14.64 times better worst-case message delay performances compared to TTCAN FD2 and TTCAN FD1, respectively. The average message delay performances also showed improved results for TTCAN FD protocols. Considerable performance improvements were achieved especially with the TTCAN FD2 and TTCAN FD3 protocols in order to provide fast and deterministic message transmissions to meet the requirements of real-time systems.

Acknowledgment

This research was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant 2219-International Postdoctoral Research Fellowship Program.

7. REFERENCES

- [1] Navet N. CAN in automotive applications: a look forward. in: Proc. 13th International CAN Conference; 5-6 March 2012; Hambach Castle, Germany. pp. 4/1-4/6.
- [2] CAN Specification, Version 2.0, Robert Bosch GmbH, Germany, 1991.
- [3] ISO 11898-1:2015 - Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling, Second edition, International Organization for Standardization, 2015.
- [4] Scarlett JJ, Brennan RW. Evaluating a new communication protocol for real-time distributed control, Robot Comput Integr Manuf 2011; 27: 627-635.
- [5] Tindell K, Burns A. Guaranteeing message latencies on control area network (CAN), In: Proc. 1st International CAN Conference; 1994; Germany. pp. 1-11.
- [6] Davis RI, Burns A, Bril RJ, Lukkien JJ. Controller area network (CAN) schedulability analysis: refuted, revisited and revised. Real-Time Syst 2007; 35: pp. 239-272.
- [7] Mubeen S, Maki-Turja J, Sjodin M. Integrating mixed transmission and practical limitations with the worst-case response-time analysis for controller area network. J Syst Software 2015; 99: pp. 66-84.
- [8] Bosch. CAN with flexible data-rate, Specification version 1.0, Robert Bosch GmbH, Germany, 2012.
- [9] Hartwich F. CAN with flexible data-rate, In: 13th International CAN Conference; 5-6 March 2012; Hambach Castle, Germany. pp. 14/1-14/9.
- [10] Kopetz H, Bauer G. The time-triggered architecture, Proc IEEE 2003; 91: 112-126.
- [11] FlexRay communications system protocol specification, version 3.0.1, The FlexRay Consortium, October 2010.
- [12] Saha I, Roy S, Ramesh S. Formal verification of fault-tolerant startup algorithms for time-triggered architectures: A survey, Proc IEEE 2016; 104: 904-922.
- [13] Leen G, Heffernan D. TTCAN: A new time-triggered controller area network. Microprocess Microsy 2002; 26: 77-94.
- [14] ISO 11898-4 - Road vehicles Controller area network (CAN) Part 4: Time-triggered communication, International Organization for Standardization, (reviewed in 2013), 2004.
- [15] Xia J, Zhang C, Bai R, Xue L. Real-time and reliability analysis of time-triggered CAN-bus. Chinese J. Aeronaut. 2013; 26: 171-178.
- [16] Short M, Sheikh I, Rizvi SAI. A transmission window technique for CAN networks. J. Syst. Archit. 2016; 69: 15-28.
- [17] Navet N, Song QY, Simonot F. Worst-case deadline failure probability in real-time applications distributed over controller area network. J Syst Architect 2000; 46: 607-617.
- [18] Fonseca J, Cotohino F, Barreiros J. Scheduling for a TTCAN network with a stochastic optimization algorithm. In: Proc. 8th International CAN Conference; 2002; pp. 07-10 - 07-16.
- [19] Mutter A. CAN FD and the CRC issue. CAN Newsletter, March 2015; pp. 4-10.
- [20] Bordoloi UD, Samii S. The frame packing problem for CAN-FD. In: IEEE Real-Time Syst Symp; 2-5 Dec. 2014; Rome, Italy. pp. 284-289.
- [21] IPMS-CAN ISO CAN FD / CAN 2.0B Controller Core. Fraunhofer Institute for Photonic Microsystems IPMS, Dresden, Germany. 2016.
- [22] MPC5777M Microcontroller datasheet, rev. 6. NXP Semiconductors, Eindhoven, The Netherlands, 2016.
- [23] Tenruh M, Charchalakakis P, Stipidis E. Modelling and Performance Analysis of a New Time-Triggered CAN FD Protocol for Real-time Distributed Control Systems. International Journal of Computer Applications, 2019; vol. 178, no. 39: pp. 38-46.
- [24] Tenruh M. Message scheduling with reduced matrix cycle and evenly distributed sparse allocation for time-triggered CAN. J Netw Comput Appl 2011; 34: pp. 1240-1251.
- [25] Schmidt K, Schmidt EG. Systematic message schedule construction for time-triggered CAN. IEEE T Veh Technol 2007; 56: 3431-3441.
- [26] Qiao X, Wang K, Sun Y, Huang W, Wang F. A genetic algorithms based optimization for TTCAN. In: IEEE Int Conf on Vehicular Electronics and Safety; 13-15 Dec. 2007; Beijing, China. pp. 1-7.
- [27] Simulink User's Guide, Matlab-Simulink R2018a. The MathWorks, Inc. Natick, MA. 2018.
- [28] TCAN33x 3.3-V CAN transceivers with CAN FD (Flexible Data Rate) datasheet. Texas Instruments, Dallas, TX, 2016.
- [29] Grenier M, Havet L, Navet N. Pushing the limits of CAN - scheduling frames with offsets provides a major performance boost. In: 4th European Congress on Embedded Real Time Software (ERTS 2008); 2008; Toulouse, France.
- [30] Yomsi PM, Bertrand D, Navet N, Davis RI. Controller area network (CAN): response time analysis with offsets. In: Proc. 9th IEEE International Workshop on Factory Communication Systems (WFCS); 21-24 May 2012; Lemgo, Germany. pp. 43-52.