Investigation of real-time optimization based on CAN bus

Feng Du¹, Yijie Cai²₅⁺, Zhiwei Guan¹₆⁺, Fengmin Tang³₄, Di Wu², Heping Shi¹, Tong Wang⁵

¹School of Automobile and Transportation, Tianjin University of Technology and Education, Tianjin, China
²School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin, China
³China Automotive Research Institute (Tianjin) Automotive Engineering Research Institute Co., Ltd
⁴School of Mechanical Engineering, Hebei University of Technology, Tianjin, China
⁵China Automotive Technology Research Center Co., Ltd., Tianjin, China
⁶School of Automobile and Rail Transportation, Tianjin Sino-German Unv. of Applied Sciences, Tianjin, China

E-mail: caiyijie1997@163.com

Abstract Aiming at the problem of CAN bus network transmission delay which affects the transmission rate, a bus network transmission delay model is established. The main indicators of network transmission delay are modeled. A hybrid optimization combining message offset compensation and identifier optimization is proposed. In terms of overall optimization, an optimization method for message segmentation and encoding scheme is designed, and a method for offset compensation of delayed messages is designed for partial parts. Symtavision software is used to simulate network conditions of the two CAN buses. Analyse the main indicators of delay after the optimization plan is adopted. The simulation results show that the hybrid optimization scheme can reduce the delay by more than 16% compared with before, which significantly improves the network communication quality, and improves the network transmission rate and scalability.

1. Introduction
The vehicle-mounted CAN bus system, which is widely used, is famous for its high performance, reliability, real-time and relatively flexible design. With the development of technology, nodes inside the bus increase, which also increase the complicity of the system, resulting in the long time high-load performance and data transmission delay in the bus network[¹].

At present, intensive researches have been carried out focusing on the data delay phenomenon worldwide. Among them, there are generally two ways that can effectively improve the communication quality of the bus system. The first method mainly includes the investigation of the bus protocol scheduling algorithm, such as the fixed priority algorithm proposed by Tindell et al.[²], the DS(deferrable server) and PE(priority ex-change) methods proposed by Lehoczky et al., the hierarchical scheduling algorithm proposed by Zuberi et al.. The other methods focus on the network communication delay, which mainly includes the research of message identifier coding and load rate algorithm research[³-⁴]. However, these methods show poor stability at the high load situation[⁵].

The research goal of this article is to optimize the bus to reduce transmission delay without changing the load. We analyzed the delay characteristics of the bus transmission and proposed a
hybrid optimization scheme of message identifier optimization and message offset compensation. The proposed scheme is applied to analyse and optimize the PTCAN and BDCAN buses. The results show that the hybrid optimization scheme using offset compensation and identifier optimization can reduce the delay by more than 16% compared with the pre-optimization.

2. Transmission delay modelling

The CAN bus system is actually a distributed network built on the basis of serial communication, consisting of multiple nodes shared on a CAN bus. Among them, each node has a different working clock and has the right to independently send requests. When the load is relatively large, multiple nodes may send messages at the same time to request bandwidth resources, which will inevitably cause transmission conflicts. In this case, the CAN bus system will sort out the messages in time, so that the messages will be transmitted in an orderly manner according to the priority.

2.1. CAN bus communication delay

The actual meaning of the CAN bus communication delay is the time period from the start of the message sent by the system to the end of the message reception. The communication delay of a single message (indicated as m in the following) can be divided into two parts according to the working principle of the CAN bus: delay in communication and the communication error recovery time. The delay in communication can be obtained by equation (1):

$$\tau_m = R_m + E_m$$  \hspace{1cm} (1)

Where, $\tau_m$ is denoted as the communication delay of message m, $R_m$ is denoted as the delay in communication of message m, $E_m$ is denoted as the recovery time required for communication errors of message m;

The definition of the delay in communication is the execution time and waiting time required for the message from the beginning of sending to the completion of receiving on the premise that there is no communication error. The time it takes to fail to send a message due to a communication error is called the recovery time required for the communication error. Further analysis is as follows:

A. Delay in communication

The delay in communication is composed of three parts: frame delay, software and CAN controller delay, and media access delay, which can be approximated by equation (2):

$$R_m = T_{df} + T_{csc} + T_{dma}$$  \hspace{1cm} (2)

Where, $R_m$ is denoted as the delay of communication of message m, $T_{df}$ is denoted as frame delay, $T_{csc}$ is denoted as software and CAN controller delay, $T_{dma}$ is denoted as media access delay.

The reason of the frame delay is the serialization of information, which is mainly affected by the frame length and bit time of the message. The frame length is derived from the addition of data bits, overhead bits, and padding bits in each frame. The bit time is measured by the baud rate (gross rate). The baud rate is fixed when working in a specific environment, so the frame length is the most important. Equation (3) is the expression of frame delay:

$$T_{df} = Frame_{t\text{th}} \times T_{bit} = (N_{data} + N_{ovhd} + N_{stuff}) \times T_{bit}$$  \hspace{1cm} (3)

Where, $T_{df}$ is denoted as the frame delay, $T_{bit}$ is denoted as the bit time, $Frame_{t\text{th}}$ is denoted as the frame length, $N_{data}$ is denoted as the actual number of data bits transmitted in the data field, $N_{ovhd}$ is denoted as the overhead bit, composed of frame start, arbitration field, control field, check field, response field and frame end, $N_{stuff}$ is denoted as the filling bit, which size needs to be determined according to the actual situation of the bit stream.

The length of the overhead bit is determined by whether the frame is a standard frame or an extended frame. The identifier determines the type of frame, and therefore it becomes the most important factor affecting frame delay.
The software delay is the time required for the main CPU to use the CAN controller to extract and store data when the CAN bus is working. The time for the CAN controller to receive and send the information in the buffer is called the CAN controller delay.

Media access delay is the extra time spent by messages with different priorities in grabbing bus bandwidth resources. This is also a hot spot in automotive electronics research. The unique operating mechanism of the CAN bus can easily have a negative impact on the messages with low-priority. As a result, a larger media access delay occurs.

B. Communication error recovery time

CAN buses have a unique error handling mechanism. The mechanism is divided into three stages: error detection, error definition and error handling. Situations that may appear in the actual system are diverse, complex and changeable, and it is difficult to confirm all the influencing factors of the communication error recovery time. So this article only analyses the error recovery time under normal circumstances.

The recovery time of error can be expressed by equation (4):

\[ E_m = C_e + t_e + \phi_e \]  \hspace{1cm} (4)

Where, \( C_e \) is denoted as the length of time a message has been transmitted before a communication transmission error occurs, \( t_e \) is denoted as the duration of the communication error, \( \phi_e \) is denoted as the sum of the transmission time of messages during communication errors.

In summary, the communication delay of CAN bus message can be obtained by equation (5):

\[ t_m = (N_{\text{data}} + N_{\text{ctrl}} + N_{\text{stuff}}) \times T_{\text{bit}} + T_{\text{dec}} + T_{\text{bus}} + C_e + t_e + \phi_e \]  \hspace{1cm} (5)

2.2. Main indicators of delay

2.2.1. Intensive message transmission. The main reason for the intensive transmission of messages is bus burst transmission[9]. After completing an address transmission on the bus, repeatedly executing data transfers to this address is called the burst transmission of the bus. It is a convenient and effective method, but in the CAN bus system, continuous transmission is extremely easy to affect the communication situation. Studying the situation of the bus burst can show the continuous sending situation in the process of message sending, which is of great significance for analysing the situation of dense message sending. Figure 1 shows the schematic diagram of the bus burst.

**Figure 1.** Burst schematic diagram

2.2.2. Jitter. During the network transmission of messages, the actual arrival time of each message must be different from the scheduled arrival time, this phenomenon is called jitter[10]. Jitter is mainly determined by the software program of the sending node, the relative simulation delay of message
transmission can be calculated by analysing the jitter situation. The calculation method is shown in Equation 6:

\[ D_{rs} = \frac{J_m}{C_d} \] (6)

Where, \( D_{rs} \) is denoted as the relative simulation delay of the message, \( J_m \) is denoted as the duration of jitter, \( C_d \) is denoted as the design cycle of the message.

2.2.3. Worst case response time. The worst case response time analysis is a research on the schedulability of the system[11-12]. Its main task is to determine the maximum possible communication delay of each message and observe whether it can be successfully transmitted within a specific time.

The worst-case response time (\( W_m \)) is mainly composed of three parts, jitter (\( J_m \)), queuing (\( Q_m \)) and transmission delay (\( R_m \)), as shown below:

\[ W_m = J_m + Q_m + R_m \] (7)

Where, \( J_m \) is denoted as the duration of jitter, \( Q_m \) is denoted as the duration of queuing, and \( R_m \) is denoted as the delay during bus transmission.

\( J_m \) is mainly determined by the software program of the sending node, \( Q_m \) is calculated by iteration as follows:

\[ Q_{m}^{n+1} = B_m + \sum_{j \in c_{lp}(m)} \left[ \frac{w_m^n + J_j + \tau_{hit}}{T_j} \right] C_j \] (8)

Where, \( B_m = \sum_{j \in c_{lp}(m)} R_m \).

\( B_m \) is denoted as the time required for the messages being transmitted to complete the transmission, and these messages have the lower priority than message m. \( l_p(m) \) and \( h_p(m) \) respectively represent the set of messages with lower and higher priority than message m. The second half of the equation represents the time required for the transmission of higher priority messages that may occur during the communication of message m. The iteration stop condition is \( Q_{m}^{n+1} = Q_{m}^{n} \).

The message transmission delay \( R_m \) can be obtained from equation (9) as:

\[ R_m = T_{df} + T_{disp} + T_{dna} \] (9)

The above analysis method is based on the ideal situation, when extending the analysis to situations with communication errors, the calculation can be:

\[ W_m = J_m + B_m + \sum_{j \in c_{lp}(m)} \left[ \frac{w_m^n + J_j + \tau_{hit}}{T_j} \right] C_j + (N_{data} + N_{ovhd} + N_{stuff}) \times T_{hit} \]

\[ + T_{disp} + T_{dna} + C_e + \tau_c + \varphi_c \] (10)

3. Optimization scheme

3.1. Message offset compensation

The full name of the OFFSET method is message offset compensation. This method can improve the communication accuracy by accurately analysing the CAN bus message transmission sequence. The process of this method is as follows:

A. Obtain all the message transmitted within a certain period of time on the CAN bus, and confirm the time information of the end of each message transmission, and then store all the received information and arrange them in the order of reception, thereby obtaining the message transmission sequence.
B. Analyse the message transmission sequence. First, define the "benchmark message" in the message record, and then select two messages without queuing delay as the "benchmark message instance". The following principles should be followed when selecting "benchmark message instance".

The worst-case delay percentage (WCRT) of each message relative to the sending cycle must be less than 100%. The relative simulation delay of each message must be less than 40%. If there are individual cases exceeding 40%, specific analysis is required. There must be more than two messages without queuing delay.

After completing the selection, record the two time points when the two benchmark message instances are received and denote them with \( t_{i,j} \) and \( t_{i,k} \), where \( j \) and \( k \) are denoted as the sequence numbers of the message instances, and \( j < k \).

C. The reference line for the entire time period is constructed through the benchmark message instance selected in the previous article, and it is shown in Figure 2. Among them, "B" represents the benchmark message instance. The reference line in the figure represents the correct reception time points. The information at these time points can be obtained by equations (10) and (11):

\[
\begin{align*}
I_{i,j} &= t_{i,j} + TP \cdot \rho + \delta_p \quad \rho \in \{0, 1, ..., k - j\} \\
TP & = \frac{t_{i,k} - t_{i,j}}{k - j}
\end{align*}
\]  

Where, \( I_{i,j} \) is denoted as the time point when the reception should be completed, \( t_{i,j} \) is denoted as the time point when the reference message \( j \) is received, \( TP \) is denoted as the correct interval between messages, \( \rho \) is denoted as the number of intervals for each message, \( \delta_p \) is denoted as message jitter compensation.

D. Finally, the arrival time of each message is recalculated according to the difference in sending time between other message instances and the benchmark message instance. Then, the communication delay time of each message instance is obtained by comparing the calculated arrival time with the actual transmission end time of the message. Let the actual transmission end time of the message instance be \( t_{a,k} \). Let the recalculated arrival time be \( t^*_{a,k} \). The communication delay time of the message can be obtained by equation (12):

\[
B_{a,k} = t_{a,k} - t^*_{a,k}
\]  

Where, \( B_{a,k} \) is denoted as the communication delay time of the message, \( t_{a,k} \) is denoted as the actual transmission end time of the message instance, \( t^*_{a,k} \) is denoted as the recalculated arrival time.

After obtaining the specific communication delay time of a certain message by using the above equation, the message can be compensated according to the measured value.

In summary, the message offset compensation method can accurately analyse the delay of the message when the benchmark message instance is determined. However, this method cannot be applied if the conditions for constituting the benchmark message instance are not met. Therefore, it is necessary to improve the quality of message transmission by using the method of message identifier.
optimization. Only in this way can the conditions for determining the benchmark message instance be met.

3.2. Optimization of message identifier

In the CAN bus system, the most basic unit is the message. Each message has its own unique identifier (ID). The main function of the identifier is to distinguish the priority of the message. The unique operating mechanism of the CAN bus can easily have a negative impact on the messages with low-priority. Based on this, this paper proposes a universal identifier encoding optimization design. The main design is as follows:

Use 11-bit identifiers compatible with CAN2.0A and CAN2.0B specifications and divide them into three segments, as shown in Figure 3.

| 11-digit identifier |
|---------------------|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Information segment | Device segment | System segment |

The upper 4 bits are the information segment of the identifier, which is a segment of information code, and can only take a value in the range of 0-15. Since the CAN specification stipulates that the 7 most significant bits cannot all be 1, the code with a value of 15 cannot be used. The 4, 5, and 6 bits are the device segment, which is the device code in a system or sub-system. It can only take a value in the range of 0-7, so it can only support up to 8 devices. The lowest 4 bits are the system segment, which is used when encoding the entire system or sub-systems. It can only take a value in the range of 0-15 and can carry up to 16 sub-systems. Since there is no obvious distinction between the equipment segment and the system segment, if the two segments are used together, 127 node devices can be connected to the network at the same time.

3.3. Optimization process. The messages should be sorted to get the message transmission sequence when receiving the CAN message. And the message transmission sequence needs to be judged. When it meets the conditions of selecting benchmark message instance, the message offset compensation method can be directly used. If it fails to meet the conditions, the message identifier optimization scheme should be implemented to meet the conditions, and then execute the compensation optimization method. The optimization flowchart is shown in Figure 4.
4. Simulation analysis

4.1. Modeling
Adopt the symTA/S software, build the real vehicle network model on BDCAN and PTCAN buses according to the design document, the network model element established is shown as in Figure 5.

4.2. Scheme selection
The premise of using the message offset compensation method is able to select the benchmark message instance.

Firstly the message sending situation on the CAN bus is analysed. Table 1 shows the analysis data of delay indicators on the BDCAN and PTCAN buses.

| Continuous message transmission situation | BDCAN | PTCAN |
|-------------------------------------------|-------|-------|
| Only the messages sent by the GW_BCM controller are continuously sent with multiple frames | Controllers such as ABSESP, EMS and other controllers send messages continuously with multiple frames |
| The longest bus burst is about 5.2ms, and about 20 frames of messages are continuously sent | The longest bus burst is 6.8ms, and there are about 26 frames of messages sent continuously |
| Each message simulation delay is below 40% | On the PTCAN bus, the simulation delay of Engine4 and SAS1 messages is higher than 40%, which does not meet the standard. Others are below 40% |
| The worst case delay is below 70% | The worst case delay is below 80% |

As shown in Table 1, the simulation delay of each BDCAN message is below 40%, the worst-case delay is below 70%, and the delay of each message is within an acceptable range. But the simulation delay of Engine4 and SAS1 messages on the PTCAN bus are higher than 40%, and the others are below 40%. The worst-case delay of each message is below 80%, which does not meet the standard.

In summary, the BDCAN bus network situation meets the criteria of the selected benchmark message instance, and the method of message offset compensation can be directly adopted. However, the PTCAN bus cannot meet the requirements, and the hybrid optimization scheme of "ID+OFFSET" needs to be adopted.

4.3. Scheme implementation
The specific plan for this optimization is as follows:
In addition to using the identifier reassignment method, some IDs are also optimized. The optimized parameters are shown in Table 2.

| Message name | ID before optimization | ID after optimization | Cycle  | node  |
|--------------|------------------------|-----------------------|--------|-------|
| SAS1         | 0x300                  | 0xc2                  | P(10 ms)| SAS   |

The optimized parameters of the OFFSET scheme are shown in Table 3.

| Message name | ID  | Cycle  | Offset parameters |
|--------------|-----|--------|-------------------|
| ESP4         | 0x209 | P(10 ms) | 8 ms             |
| APB1         | 0x163 | P(20 ms) | 2 ms             |
| ABS1         | 0x200 | P(20 ms) | 6 ms             |
| ABS3         | 0x207 | P(10 ms) | 4 ms             |
| ABS2         | 0x208 | P(20 ms) | 12 ms            |
| ESP1         | 0x153 | P(10 ms) | 0 ms             |
| ESP2         | 0x154 | P(20 ms) | 16 ms            |
| Engine7      | 0x221 | P(10 ms) | 4 ms             |
| Engine3      | 0x308 | P(20 ms) | 6 ms             |
| Engine4      | 0x312 | P(10 ms) | 8 ms             |
| Engine8      | 0x316 | P(20 ms) | 12 ms            |
| Engine6      | 0x322 | P(1000 ms) | 16 ms      |
| Engine5      | 0x608 | P(100 ms) | 0 ms             |

4.4. Simulation results

The BDCAN and PTCAN model are both optimized by the OFFSET+ID method. Figure 6 shows the burst duration of the PTCAN model before optimization, while Figure 7 shows the burst duration of the PTCAN model after optimization. The burst situation of BDCAN is similar to that of PTCAN and will not be listed here. Figure 8 shows the relative delay and worst-case delay of the BDCAN model before optimization, while Figure 9 shows the corresponding results of the same model after optimization. Similarly, the relatively delay and worst-case delay of the PTCAN model before optimization is shown in Figure 10, and the result of the optimized PTCAN model is shown in Figure 11.

![Figure 6. The burst distribution of PTCAN messages before optimization](image-url)
Figure 7. The burst distribution of PTCAN messages after optimization

Figure 8. The delay distribution of BDCAN messages before optimization

Figure 9. The delay distribution of BDCAN messages after optimization
**Figure 10.** The delay distribution of PTCAN messages before optimization

**Figure 11.** The delay distribution of PTCAN messages after optimization

The relative simulation delay comparison is shown in Figure 12, and the worst-case delay data comparison is shown in Figure 13.

**Figure 12.** Relative simulation delay data comparison chart
According to Figure 12 and Figure 13, after optimization, the relative simulation delay of each PTCAN message is below 30%, mainly in the 0-10% interval, and the number of messages with a high delay percentage is greatly reduced. The relative simulation delay of each BDCAN message is below 20%, and the worst-case delay is less than 60%. The optimization scheme proposed in this paper can reduce the delay time by more than 16%, and this scheme can greatly improve the network communication quality and increase the scalability of the subsequent network.

5. Conclusion
This article takes the real-time performance of the CAN bus system as the object, and studies the delay characteristics of the bus communication system. Aiming at the various indicators of message delay, the optimization scheme of "OFFSET+ID" is proposed, and simulation verification is performed on two CAN buses. The specific conclusions are as follows:

A. According to the delay characteristics of CAN bus, three indicators for evaluating real-time performance are proposed. Three models of message burst, jitter and worst-case response time are obtained by establishing a communication delay model. These models are very helpful for evaluating the real-time performance of CAN bus.
B. Based on the goal of reducing the bus transmission delay, an optimization scheme is proposed. Three indicators of real-time performance of the system are analysed through simulation. This scheme greatly improves the real-time performance under the condition of a certain load, and has important application value for the automobile bus network.

Acknowledgment
This research was supported by The National Key Research and Development Program of China under Project of 2017YFB0102501 and Science and Technology Plan Project of Tianjin, China (Grant No. 17XZRGGX00070, 16JCZDJC38200).

Reference
[1]. Chen T. Vehicle CAN network communication function conformance test. Int J Eng. 2020, 6(1)
[2]. Tindell K and Burns A. Fixed Priority Scheduling of Hard Real-time Multi-media Disk Traffic[J]. Comput. J (8):8.
[3]. Qi Jin C,Yong X, Jun Kun F and Kai C. Study on Development and Testing of Vehicle CAN Network Platform Based on CANoe. Appl. Mech. Mater. 2014, 3634:56-59.
[4]. Jiangbo L. Research on CAN Bus for Automotive Driving Control Based on SAEJ1939 Protocol.Int J Comput Eng. 2020, 5(2)
[5]. Alaei R, Moallem P and Bohlooli A. Statistical based algorithm for reducing bit stuffing in the Controller Area Networks. Microelectron J 2020.
[6]. Houda J. Hybrid Task and Message Scheduling in Hard Real Time Distributed Systems over FlexRay Bus. *Comput Tech Appl*. 2013, 4(9).

[7]. Daisuke U and Takeyuki S. Controller Area Network and Its Reduced Wiring Technology. *IEICE T Commun*. 2019, E102.B(7):1248-62.

[8]. Rohrer R A , Pitla S and Luck J D . Tractor CAN bus interface tools and application development for real-time data analysis[J]. *Comput Electron Agr*. 2019, 163(Aug 2019).

[9]. Fang Li, Li Fang Wang and Cheng Lin Liao. Period Optimization Scheduling Algorithm Research of CAN Bus in Electric Vehicle. *Appl. Mech. Mater*. 2015, 3827:273-6.

[10]. Zhen Hua Luo and Fen Jiang. Research on Electronic Automotive Combined Instrument Based on CAN Bus. *Adv Mat Res*.2014, 3349:731-4.

[11]. Kai L, Georgia G, Rodolfo P, Zheng Wu and Nikolay S. A formal approach to the WCRT analysis of multicore systems with memory contention under phase-structured task sets. *Real-Time Syst*. 2014, 50(5-6):736-73.

[12]. Guoqi X, Gang Z, Ryo K, Hiroaki T, Renfa L and Keqin L. Exact WCRT Analysis for Message-Processing Tasks on Gateway-Integrated In-Vehicle CAN Clusters. *Acm T Embed Comput S*. 2018, 17(6):1-29.