

Research Article

Online Real-Time Estimation of Response Time for Periodic Messages in Controller Area Networks

Dahui Gao,^{1,2} Qingfeng Wang,¹ Yong Lei,¹ Zheng Chen,³ and Linying Shangguan¹

¹State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

²Ocean College, Zhejiang University, Hangzhou 310027, China

³Nanjing Research Institute of Electronics Technology, 8 Guorui Road, Nanjing 210039, China

Correspondence should be addressed to Yong Lei; ylei@zju.edu.cn

Received 19 August 2015; Revised 22 November 2015; Accepted 23 November 2015

Academic Editor: Anna Vila

Copyright © 2015 Dahui Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Controller Area Networks (CAN) are widely used in industrial Networked Control Systems (NCSs), such as construction machineries, hybrid vehicles, robotics, and other applications. The message response time (MRT) or communication delay is the main issue to degrade the performance of CAN-based NCSs since its exact value is time-varying and unpredictable. The online acquisition of exact MRT can be quite helpful for the delay compensation of NCSs. However, since the clocks on different nodes are asynchronous, the MRT acquisition in CAN is a challenging work. The current delay acquisition methods for asynchronous systems are not suitable for the delay compensation in CAN-based NCSs because they either increase the bus load of CAN or cannot acquire the exact MRT in real time. In this paper, we propose a novel online real-time MRT estimation method for periodic CAN messages based on the analysis of message traces on CAN bus. The proposed method can estimate the exact MRT of the received message instance in real time without increasing the bus load and can be conveniently embedded into the CAN nodes without requiring additional equipment. In order to validate the proposed method, practical experiments are carried out and the experimental results show that the proposed method can effectively estimate the exact MRT of periodic CAN messages.

1. Introduction

The Networked Control Systems (NCSs) based on Controller Area Networks (CAN) have been widely used in construction machineries, hybrid vehicles, robotics, and other applications. In these NCSs, the feedback signals and control commands carried by CAN messages will suffer from time-varying and unpredictable communication delay or message response time (MRT). When using the random delayed feedback signals and control commands in the control loop, the performance of NCSs will be degraded, and, ultimately, the comfortability, operability, and service life of the equipment will be significantly degraded [1, 2]. How to reduce the adverse impacts of random communication delay in NCSs has become an important research issue. In recent years, many studies have been devoted to the delay in NCSs.

In the communication aspect, new protocols (e.g., TTCAN [3] and FlexRay [4]), new encoding mechanism [5], new scheduling policy [6], and queuing/buffering method [7] are proposed to reduce the delay and its variation. However, since these solutions will cause some new drawbacks, such as side effects in fault tolerant ability, complexity of hardware and software, and artificial increase of delays, the conventional CAN are still widely used in industrial applications. The delay is inevitable for CAN-based NCSs. In the control aspect, various control methods are proposed to compensate the delay in NCSs. The robust control [8, 9], predictive control [10, 11], stochastic optimal control [12], and switched system approach [13] are some of the most popular methods. However, in order to improve the performance of the NCSs when using these methods, diverse key issues need to be solved for each method. For real-time delay compensation

methods, such as predictive control method, one of the most important issues is to obtain the exact delay of the received data packet in real time.

In order to achieve the real-time delay compensation, the delay acquisition should satisfy the following requirements:

- (i) The delay should be acquired in real time. That means the delay of a received message instance should be identified immediately after its receiving completion.
- (ii) The acquired delay should be accurate enough. If the acquired delay is too inaccurate, the NCSs' performance may be degraded further by the delay compensation. The more accurate the delay is, the better the compensation effect can be achieved.
- (iii) The delay acquisition method should not increase the bus load since the communication performance of CAN is greatly affected by its bus load due to the limited bandwidth.

However, due to some limitations, it is a challenging work to acquire the exact delay in the case of meeting the above requirements. As the traffic congestion on CAN bus, which greatly affects the delay, is random, the exact MRT of a data packet cannot be theoretically calculated or predicted. In addition, as the clocks of CAN nodes are asynchronous, the MRT is also difficult to measure. Therefore, it is necessary to study how to acquire the exact MRT of periodic CAN messages in real time without increasing the bus load.

In existing literatures, some methods are proposed to acquire the communication delay in asynchronous clock systems. The clock synchronization methods given by [14, 15] can greatly reduce the difficulty to obtain the delays, but more bus bandwidths and processor resources are required due to the use of synchronous messages. In [16, 17], the authors use the measured round-trip time (RTT) delay in delay compensation. Since the local timestamp is added to each data packet in the RTT measurement, the bus load will also increase. Besides, the lack of identification of each delay part (sensor-to-controller delay and controller-to-actuator delay) will limit its use in some real-time delay compensation methods [7, 11, 13].

Among the literatures about message response time of CAN, the majority are focused on the analysis of MRT in the worst case [18, 19] when using different methods (e.g., stochastic method [20], statistic method [21], and offset based method [22]) and considering different limitations (e.g., communication errors [23], queue policies [24], priority inversion [25], and other limitations [26, 27]). Little attention has been paid to the acquisition of the exact MRT of CAN. Since the actual MRT varies widely and the worst-case MRT is an upper boundary, the compensation effect will be greatly limited when using the worst-case MRT in delay compensation. In severe cases, the NCSs' performance may be degraded further. The average MRT obtained by the method given in [28] also will limit the compensation effect. The most relevant work to this trace is given in [26], where the MRT is obtained by reconstructing the arrival time, which is the start moment of MRT, based on the analysis of message traces. However, as the reconstruction of arrival time needs two

reference message instances that satisfy certain conditions and are separated by a period of time, the MRT of message instances between the reference message instances cannot be estimated in real time at their corresponding receiving completion moments. This makes it unsuitable for real-time delay compensation in NCSs.

As it can be seen from the literatures, all the current delay acquisition methods are not suitable for the real-time delay compensation in CAN-based NCSs since they either increase the bus load or cannot acquire the exact MRT in real time. Therefore, the objective of this paper is to propose a novel MRT estimation method, which can estimate the exact MRT of CAN in real time without increasing the bus load. In this paper, a more in-depth analysis of message traces is conducted, where the possible traffic congestion on CAN bus is studied by taking the message transmission order and message contents into consideration. Based on the analysis, a novel MRT estimation method is proposed, where the MRT is estimated by estimating each component of it in most cases and is estimated by using an incremental method in other cases. The proposed method has three advantages over the existing methods: (1) it only needs to monitor the message traces on CAN bus. The bus load will not increase and no additional bandwidth of CAN bus is required. (2) At the receiving completion moment, the exact MRT of the received data packet can be estimated immediately. (3) It can be conveniently embedded into the destination node as a software package. No additional equipment is required.

The rest of this paper is organized as follows. In Section 2, the system model is introduced, followed by the analysis of MRT composition and message traces in Section 3. Then, in Section 4, the novel online real-time MRT estimation method is introduced. The experiments are carried out in Section 5 to demonstrate the proposed method, where the estimation accuracy and general applicability of the proposed method are discussed. In Section 6, some practical issues are discussed. Conclusion and future work are given in Section 7.

2. System Model

In this paper, we consider a system model widely used in CAN-based NCSs. In the system, a set of nodes are connected to a CAN bus, and each node has asynchronous clock and starts with arbitrary phase. In addition to periodic message, each node may also send sporadic message which is triggered by the events that only happen when certain conditions are satisfied. The period of each periodic message is a multiple of the minimum message period. In the source node, represented as node j in Figure 1, the periodic messages are sent by a single task, which is named as $TxTask$ and has a task cycle equal to the greatest common divider of the message periods in node j . During each task cycle, all the periodic messages to be sent are enqueued at the same time by $TxTask$. In $TxTask$, the messages are copied into the sending buffers according to their queuing order. When all buffers are occupied, the residual messages wait for buffer releases and are copied into the buffer once it becomes empty. Once a message is copied into a buffer, it is ready to be

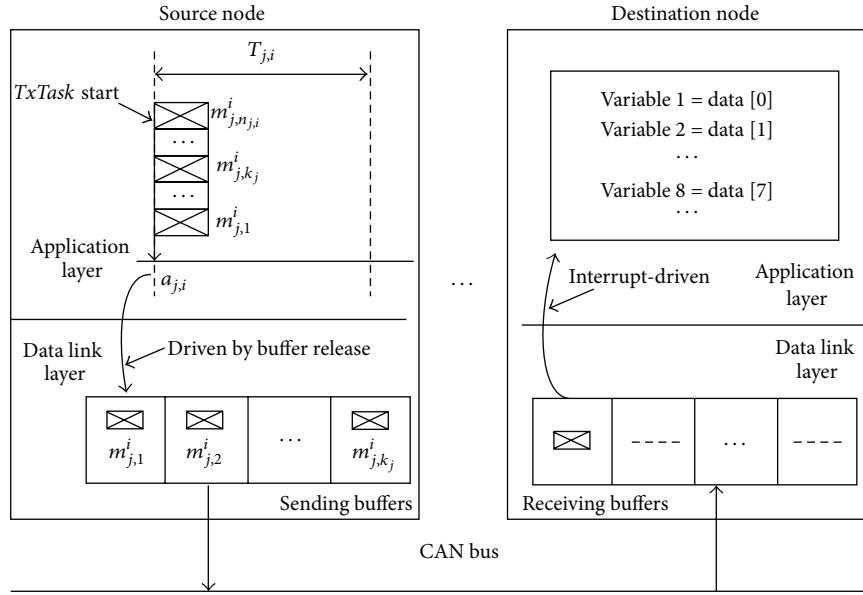


FIGURE 1: The CAN system model used in this paper.

transmitted and becomes nonabortable. The messages that have not been copied into buffers until a new task cycle starts will be aborted. The messages in the sending buffers of the same node are transmitted according to priority policy or FIFO policy. In the destination node, the receiving task is interrupt-driven. Since the delay caused by polling receiving process can be measured in the destination node, we do not consider the polling receiving case in this paper.

The main notations used throughout this paper are as follows: the term T_j denotes the task cycle of node j (or the source node). The term $T_{j,i}$ denotes the i th task cycle. The term $m_{j,\alpha}^i$ ($\alpha = 1, 2, \dots, n_{j,i}$) denotes the instance of message $m_{j,\alpha}$ sent during $T_{j,i}$. The term $a_{j,i}$ denotes the start moment of the $TxTask$ in $T_{j,i}$. Message m represents an arbitrary message to be analyzed. The terms R_m and T_m denote the MRT and the message period of message m , respectively.

In addition to the notations, the important definitions used in this paper are as follows.

Definition 1. The MRT of a message instance is defined as the time from the start of the $TxTask$ sending it to the receiving completion moment of it on the destination node.

Definition 2. The *base rate message* is defined as the message which is sent by the source node in each task cycle no matter whether the other messages are sent.

The assumptions used in this paper are as follows:

- (i) There is no communication error in the CAN.
- (ii) The deadline of each periodic message is no larger than its period and each periodic message can meet its deadline.
- (iii) The copy time of each message is smaller than the minimum transmission time of a message.

- (iv) The periodic messages to be sent by a node during the same task cycle are sent without offset.
- (v) For each node, there is at least one *base rate message*.
- (vi) There is no clock drift and jitter of message period.

Among the assumptions, the first three are made to constrain the applicability of our method. The other three are made to simplify the following analysis. As the first three assumptions can be satisfied in most well-designed NCSs and the following analysis can be extended to the cases where the last three assumptions cannot be satisfied (the extension will be discussed in Section 6), our method is applicable in most cases.

3. Analysis of MRT and Message Traces

3.1. Analysis of Message Response Time. According to the system model, the message sent from one node to another node will experience several delay links shown in Figure 2. The delay links are described as follows:

τ_{task} : the response time of $TxTask$.

τ_{ws}^m : the time used to wait for an empty sending buffer by message m .

τ_{cs} : the time used to copy the message from the application layer to the sending buffer.

w_m : the queuing delay of message m in the sending buffer, which consists of the blocking time B_m caused by the message being transmitted and the interference φ_m caused by other messages when message m is ready for the transmission.

C_m : the transmission time of message m on the CAN bus.

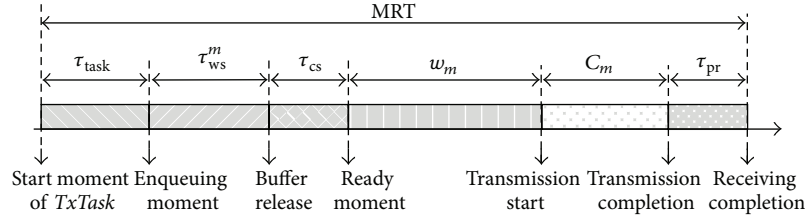


FIGURE 2: Compositions of the message response time (MRT) in CAN.

τ_{pr} : the receiving process time of the message in the destination node.

Hence, the exact MRT of message m (R_m) is

$$R_m = \tau_{task} + \tau_{ws}^m + \tau_{cs} + w_m + C_m + \tau_{pr}$$

$$= \tau_p + B_m + \theta_m + C_m \quad (1)$$

with $\tau_p = \tau_{task} + \tau_{cs} + \tau_{pr}$, $w_m = B_m + \varphi_m$, $\theta_m = \tau_{ws}^m + \varphi_m$,

where τ_p , called *processing time*, is the total process time in source node and destination node and θ_m , called *interference*, is the total time required by the messages preempting message m to be transmitted since message m was sent by the TxTask. In (1), since *processing time* τ_p can be preestimated experimentally because it is determined by the property of CAN nodes, and *transmission time* C_m can be accurately calculated according to the contents in message m in real time, the main obstacles to estimate the MRT R_m lie on the real-time estimations of *blocking time* B_m and *interference* θ_m . Since B_m and θ_m are caused by traffic congestion, they can be identified by analyzing the message traces (including reception timestamp, message ID, transmission order, and data contents) on CAN bus based on the arbitration mechanism given in [29]. Besides, since the messages that delayed message m are transmitted before message m , B_m and θ_m can be identified in real time. Therefore, we estimate R_m by estimating/calculating each component of it in this paper instead of reconstructing its start moment as [26] does. By doing so, R_m can be obtained in real time. In addition, as only the message traces are monitored and no auxiliary timestamps or messages are sent on CAN bus, the bus load will not increase. The details of the message traces analysis for the estimation of B_m and θ_m are given in the following subsection.

3.2. Analysis of Message Traces. On CAN bus, the messages will be transmitted according to the arbitration mechanism given in [29] and form message blocks, which consist of one or more consecutively transmitted messages and are separated by bus idles (longer than 3 bits' time), as shown in Figure 3. Usually, the messages that participate in the arbitration will be transmitted in the same message block. So, we identify the *blocking time* B_m and the *interference* θ_m by analyzing the message traces in the same message block as message m .

In practical CAN, the composition of message block will have different forms. Due to the space limitation, we only analyze the most general form in this paper. In the later

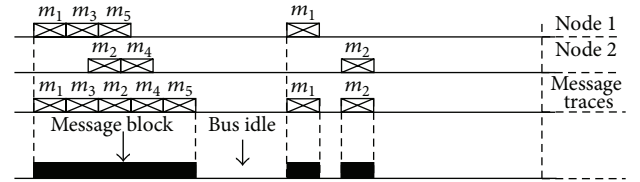


FIGURE 3: The message transmission of an example system with two nodes.

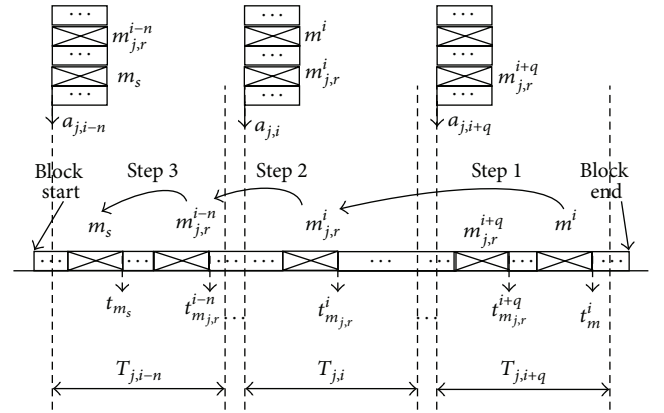


FIGURE 4: The general case of the message traces in a message block.

analysis, the source node and destination node represent the source node and destination node of message m unless specifically noted. Besides, the sporadic messages are not considered in the analysis but in the MRT estimation.

Based on the given system model, the periodic messages sent in the same task cycle by the source node will be transmitted in the same message block (e.g., m_1 , m_3 , and m_5 in Figure 3). However, in a message block, there may be messages sent by the source node in different task cycles. The general case of message traces in a message block is given in Figure 4, where the periodic messages sent during the task cycles from $T_{j,i-n}$ to $T_{j,i+q}$ by the source node are transmitted in the same message block that also contains the messages sent by other nodes.

In Figure 4, the messages with thick outline represent the messages sent by the source node. Message instance m_s is the most front one of the message instances sent by the source node in the message block. Message $m_{j,r}$ denotes a *base rate message* (based on the fifth assumption). $m_{j,r}^{i-n}$, $m_{j,r}^i$, and $m_{j,r}^{i+q}$ are the instances of $m_{j,r}$. Message instance m_i is the instance

of message m sent in $T_{j,i}$ and received q ($q = 0, 1, 2, \dots$) task cycles later. The terms $t_{m_s}, t_{m_{j,r}}^{i-n}, t_{m_{j,r}}^i, t_{m_{j,r}}^{i+q}$, and t_m^i are the timestamps of their receiving completion, respectively.

As shown in Figure 4, before m^i , there are message instances from the same node with it, such as $m_{j,r}^{i-n}, m_{j,r}^i$, and $m_{j,r}^{i+q}$. In order to identify whether these message instances have competitive relationships with m^i , the arbitration situation in the source node is required. However, on the destination node, the arbitration situation in the source node is hard to identify. Therefore, it is difficult to analyze the competitive relationships between m^i and the message instances before it one by one. In order to simplify the analysis, we use m_s as a *reference message instance* and transfer the MRT estimation from m^i to m_s since there is no message instance from the same node before m_s . During the transfer, the *base rate message* $m_{j,r}$ is used as a bridge since its instances will be received by the destination node in the same task cycles as they are sent. The MRT correlation between m^i and m_s is derived as follows.

Step 1. Since m^i is sent in the same task cycle as $m_{j,r}^i$, the start moments of their MRT are the same (based on the fourth assumption). Hence, we can get the following correlation:

$$R_m^i = R_{m_{j,r}}^i + t_m^i - t_{m_{j,r}}^i, \quad (2)$$

where R_m^i and $R_{m_{j,r}}^i$ are the MRTs of m^i and $m_{j,r}^i$, respectively. In order to determine the message instance $m_{j,r}^i$ corresponding to m^i automatically, we define a variable *Index*. The *Index* is set to be zero when a message block that contains message m has only one instance of $m_{j,r}$. When receiving a new message instance of $m_{j,r}$, the *Index* increases by 1 and is assigned to the instance of $m_{j,r}$. Then, the instance of $m_{j,r}$ corresponding to the z th received instance of message m is the instance with *Index* equal to $z \cdot g$, where $z \in N^+$ and $g = T_m/T_j$. If there is no instance of $m_{j,r}$ before m^i in the message block, this step can be omitted.

Step 2. If there are multiple instances of $m_{j,r}$ before m^i , we can get the following correlation (use the sixth assumption):

$$R_{m_{j,r}}^i = R_{m_{j,r}}^{i-n} + t_{m_{j,r}}^i - t_{m_{j,r}}^{i-n} - n \cdot T_j, \quad (3)$$

where $R_{m_{j,r}}^{i-n}$ is the MRT of $m_{j,r}^{i-n}$ and n is the number of instances of $m_{j,r}$ before $m_{j,r}^i$ in the message block. Parameter n can be determined by the *Index* value. If there are no multiple instances of $m_{j,r}$ before m^i , this step can be omitted.

Step 3. Since m_s is sent in the same task cycle as $m_{j,r}^{i-n}$, we can get the following correlation (use the fourth assumption):

$$R_{m_{j,r}}^{i-n} = R_{m_s} + t_{m_{j,r}}^{i-n} - t_{m_s}, \quad (4)$$

where R_{m_s} is the MRT of m_s . Based on the above three steps, we can get the MRT correlation between m^i and m_s :

$$R_m^i = R_{m_s} + t_m^i - t_{m_s} - n \cdot T_j. \quad (5)$$

TABLE 1: Possible congestion situations of the *reference message instance*.

Congestion situations	Location situation	Priority comparison ^a	B_{m_s}	\emptyset_{m_s}
1	Scenario 1	—	0	0
2	Scenario 2	—	$[0, C_a)$	0
3	Scenario 3	$P_{m_a} < P_{m_s}$	$[0, C_a)$	0
4	Scenario 3	$P_{m_a} > P_{m_s}$	Unknown	Unknown

a: $P_{m_a} < P_{m_s}$ and $P_{m_a} > P_{m_s}$ mean the priority of message m_a is lower or higher than message m_s , respectively.

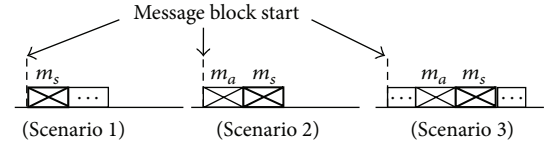


FIGURE 5: Possible location situations of the *reference message instance* m_s in a message block.

Since (5) is derived only based on the message traces monitored before the reception of m^i , it can be obtained in real time. In (5), R_{m_s} is unknown. In order to estimate it, we identify the congestion situation of m_s by analyzing the possible location situations of m_s in a message block, as shown in Figure 5. The scenarios are described as follows.

Scenario 1. m_s is the first message in the message block. In this scenario, it will not be delayed by any other messages.

Scenario 2. The message instance transmitted adjacently before m_s , denoted by m_a , is the first message instance in the message block. In this scenario, m_s is at most delayed by m_a .

Scenario 3. There are more than one message instance transmitted before m_s in the message block. In this scenario, if the priority of m_a is lower than m_s , m_s is at most delayed by m_a when m_s is ready for the transmission.

According to the above scenarios, we can get the possible congestion situations of m_s , as shown in Table 1. The term C_a is the *transmission time* of m_a . The terms B_{m_s} and \emptyset_{m_s} are the *blocking time* and *interference* of m_s , respectively. According to Table 1, if m_s is in *congestion situation* 1, 2, or 3, its MRT can be estimated, and then the MRT of m^i can be estimated according to (5). However, if m_s is in *congestion situation* 4, its MRT cannot be estimated. We use an incremental estimation to estimate the MRT of m^i in this situation. The details of MRT estimation are given in the next section.

4. Proposed Methodology

Based on the previous analysis, we propose a method to estimate the MRT of message m in two ways according to the actual congestion situation of the *reference message instance*, as shown in Figure 6. If the *reference message instance* is not in *congestion situation* 4, the MRT of message m is estimated by

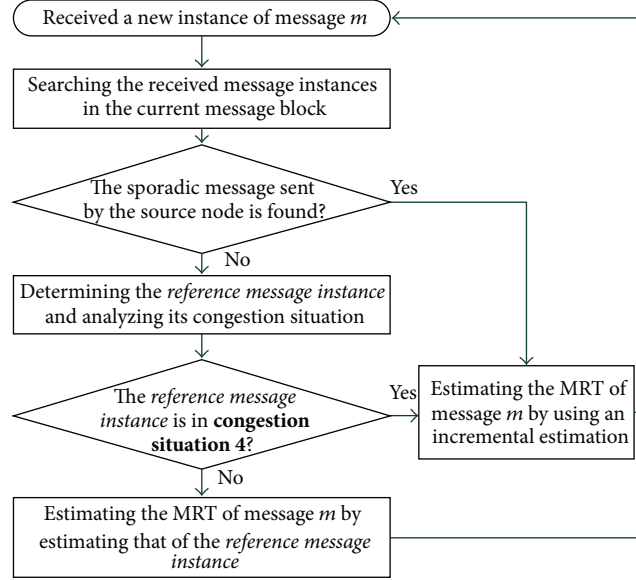


FIGURE 6: Flow chart of the online real-time MRT estimation method.

estimating that of the *reference message instance*. Otherwise, it is estimated by using an incremental estimation. It is noted that if the sporadic messages sent by the source node are received before message m in the message block, the MRT of message m is also estimated by the incremental estimation. The details are given in the following.

4.1. Estimating the MRT of Message m by Estimating That of the Reference Message Instance. In this way, the MRT of message m is estimated according to (5). The estimation formula is

$$\hat{R}_m^i = \hat{R}_{m_s} + t_m^i - t_{m_s} - nT_j \quad (6)$$

with $\hat{R}_{m_s} = \hat{\tau}_p + \hat{B}_{m_s} + \emptyset_{m_s} + C_s$,

where C_s is the transmission time of m_s . In (6), only \hat{R}_{m_s} needs to be estimated because the reception timestamps, t_m^i and t_{m_s} , and the parameter n can be directly acquired from the message traces. Furthermore, since the *interference* \emptyset_{m_s} equals zero and the *transmission time* C_s can be accurately calculated according to the definition of data frame and stuffing mechanism given in [29] by using the contents in m_s , only the *processing time* $\hat{\tau}_p$ and the *blocking time* \hat{B}_{m_s} need to be estimated. The details are given in the following.

(i) Estimating the Processing Time $\hat{\tau}_p$. In our method, the *processing time* $\hat{\tau}_p$ is preestimated and set to be constant in the MRT estimation. First, we conduct an experiment where only the source node and the destination node are connected to the CAN bus. The source node periodically sends two consecutive messages with random data by using the same sending buffer. The destination node records the message traces. By subtracting the *transmission time* of the second message from the time interval between the reception timestamps of these two messages in each period, we can get

some samples of τ_p . Then, we use the finite Gaussian Mixture model to fit the probability distribution function (pdf) of τ_p :

$$p(\tau_p) = \sum_{i=1}^K \rho_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(\tau_p - \mu_i)^2}{2\sigma_i^2}\right), \quad (7)$$

where $\sum_{i=1}^K \rho_i = 1$, the parameter K is determined by k -means clustering algorithm, and the parameters (μ_i, σ_i) are learned by using Expectation Maximization algorithm. Finally, we use the expectation given in (8) to estimate τ_p :

$$\hat{\tau}_p = \int_{\gamma_1}^{\gamma_2} p(\tau_p) d\tau_p. \quad (8)$$

The lower boundary γ_1 and upper boundary γ_2 are set to be the minimum and maximum sample values.

(ii) Estimating the Blocking Time \hat{B}_{m_s} . Since the exact ready moment of the *reference message instance* for the transmission cannot be identified in *congestion situations* 2 and 3, we assume that it is ready at any moment during the current transmission with the same probability. Then, we can estimate \hat{B}_{m_s} by using the uniform distribution to describe the pdf of ready moment:

$$\hat{B}_{m_s} = \frac{1}{C_a} \int_0^{C_a} (C_a - t) dt = \frac{C_a}{2}. \quad (9)$$

It is noted that (9) may cause an estimation error up to $\pm C_a/2$.

4.2. Estimating the MRT of Message m by Using the Incremental Estimation. The time interval between two consecutively received instances of message m , called *measured period*, can

reflect the MRT variation of message m [30]. So, the MRT can be estimated by the incremental estimation as follows:

$$\begin{aligned} \hat{R}_m^i &= \hat{R}_m^{\text{lst}} + (T_m^i - T_m) \\ \text{with } T_m^i &= t_m^i - t_m^{\text{lst}} \text{ cs. } \hat{R}_m^i = \begin{cases} R_m^{i.\text{min}}, & \text{if } \hat{R}_m^i < R_m^{i.\text{min}} \\ R_m^{i.\text{max}}, & \text{if } \hat{R}_m^i > R_m^{i.\text{max}} \end{cases} \end{aligned} \quad (10)$$

where \hat{R}_m^{lst} and t_m^{lst} are the MRT and the receiving completion timestamp of the instance of message m received last time, respectively. The term T_m^i is the *measured period* of m^i . The lower boundary $R_m^{i.\text{min}}$ and upper boundary $R_m^{i.\text{max}}$ are used to constrain the cumulative estimation error when the incremental estimation is continuously used for a long time. They are determined according to the actual message traces:

$$\begin{aligned} R_m^{i.\text{min}} &= \hat{\tau}_p + C_m, \\ R_m^{i.\text{max}} &= \max(R_{m_s}) + t_m^i - t_{m_s} - n \cdot T_j \\ &= [\hat{\tau}_p + C_c + t_{m_s} - t_{m_c}] + t_m^i - t_{m_s} - n \cdot T_j \\ &= \hat{\tau}_p + C_c + t_m^i - t_{m_c} - n \cdot T_j. \end{aligned} \quad (11)$$

The term C_c is the *transmission time* of message m_c which is the first one of the messages that has lower priority than m_s in the direction from m_s to the first message in the message block. If all the messages before m_s in the message block have higher priority than m_s , message m_c denotes the first message in the message block. The term t_{m_c} is the reception timestamp of message m_c . The minimum MRT $R_m^{i.\text{min}}$ occurs in the scenario where message m is not delayed by other messages. The maximum MRT $R_m^{i.\text{max}}$ occurs in the scenario where the messages between m_c and m_s all delay m_s .

4.3. Estimation Error Analysis. When the MRT of message m is estimated by estimating that of the *reference message instance*, the estimation error E_m^i is

$$\begin{aligned} E_m^i &= \hat{R}_m^i - R_m^i = \hat{R}_{m_s} - R_{m_s} + 2 \cdot e_t + n \cdot e_T \\ &= e_\tau + e_B + 2 \cdot e_t + n \cdot e_T, \end{aligned} \quad (12)$$

where e_t and e_T are the resolution of the time stamping clock on the destination node and the timer clock on the source node, respectively. The terms e_τ and e_B denote the estimation errors of $\hat{\tau}_p$ and \hat{B}_{m_s} , respectively. In (12), e_τ is caused by the randomness of node's properties, and e_B is caused by the unknown ready moment of m_s . Usually, the resolutions of clocks are small and the randomness of node's properties is not obvious. So the estimation error E_m^i is mainly caused by e_B . According to (9), the value range of the estimation error E_m^i is

$$\begin{aligned} |E_m^i| &\leq \frac{1}{2}C_a + \delta \\ \text{with } \delta &= \max(|e_\tau|) + 2 \cdot \max(|e_t|) + n \cdot \max(|e_T|). \end{aligned} \quad (13)$$

Since C_a is related to baud rate, the boundary of estimation error E_m^i in this way will decrease as baud rate increases.

When the MRT of message m is estimated by using the incremental estimation, the estimation error E_m^i is

$$E_m^i = \hat{R}_m^i - R_m^i = E_m^{\text{lst}} + 2 \cdot e_t + e_T, \quad (14)$$

where E_m^{lst} is the estimation error of the instance of message m received last time. From (14), we can see that cumulative estimation error may be caused by the continuous use of the incremental estimation for a long time. The worst estimation error in this way is

$$|E_m^i| \leq R_m^{i.\text{max}} - R_m^{i.\text{min}}. \quad (15)$$

However, in practical applications, the incremental estimation will not be continuously used for a long time because of varies of random factors. So the estimation error E_m^i is mainly affected by the estimation error of the *blocking time* \hat{B}_{m_s} .

5. Experiments and Results

In order to validate the proposed method, experiments are carried out in this section. Firstly, by comparing the MRT estimated by our method and the worst-case MRT with the actual MRT, the estimation accuracy of our method is analyzed. Then, in order to assess the general applicability of our method, we apply it to the messages in the CAN systems with different baud rates and bus loads. Finally, we conduct a case study on condition monitoring to see whether the estimation error is acceptable or not.

5.1. Experimental Platform. The topology of the experimental platform is shown in Figure 7. The CAN system used in the platform is constructed according to the actual CAN system in a hybrid excavator. Table 2 gives part of the total 41 messages in the CAN system. The baud rate is 250 Kbps. The message sending order and processing time are preidentified. To obtain the actual MRT as a reference to evaluate the estimation accuracy, we add a wire between the test node and "monitor A" and send a sync pulse before each instance of the message, which is to be analyzed, through the wire. The sync pulse is solely for result comparison with actual MRT, and it will not be needed in practical applications. In the test, "monitor A" measures the actual MRT by calculating the time interval between the arrivals of sync pulse and message instance and sends the actual MRT to the PC through RS232 immediately. At the same time, the PC records the message traces on CAN bus by "monitor B." A software application developed by us is installed on the PC to receive the actual MRT and estimate the MRT in real time according to the monitored message traces. The actual MRT and the estimated MRT are saved in pairs. Each test lasts for 5 minutes.

5.2. Evaluation of Estimation Accuracy. In order to evaluate the estimation accuracy of our method, we apply it to an arbitrary message. In the test, the message with priority

TABLE 2: Part of the messages in the experimental platform.

Priority (ID)	Period (ms)	Source node	Sending order	Processing time (ms)
1	10	2	2	0.183
2	10	1	2	0.171
3	10	1	3	0.171
4	10	1	1	0.171
5	50	3	2	0.185
6	20	3	1	0.185
7	10	2	3	0.183
8	10	2	1	0.183
9	50	4	4	0.194
10	50	4	5	0.194
11	50	4	2	0.194
12	50	4	3	0.194
13	50	4	1	0.194

TABLE 3: Evaluation parameters of estimation accuracy.

Symbol	Description
e_{mp}	The maximum positive estimation error which is used to assess the worst underestimation of our method
e_{mn}	The maximum negative estimation error which is used to assess the worst overestimation of our method
e_{exp}	The expectation of estimation error which is used to assess the average estimation accuracy
e_{va}	The variance of estimation error which is used to assess the stability of estimation accuracy

TABLE 4: Evaluation parameter values of estimation accuracy.

Method	e_{mp} (ms)	e_{mn} (ms)	e_{exp} (ms)	e_{va} (ms ²)
Worst case	1.971	—	1.873	3.623
Our method	0.347	-0.270	0.031	0.002

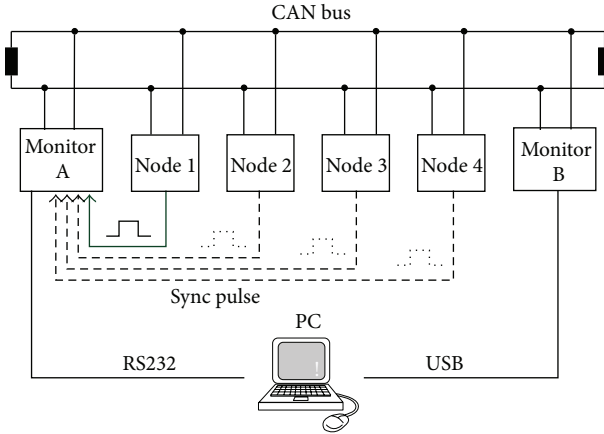


FIGURE 7: Topology of the experimental platform.

4 is used as an example. In Figure 8, the MRT estimated by our method (represented as eMRT) and the worst-case MRT obtained according to [18] (represented as WCMRT) are compared with the actual MRT. The instances shown in Figure 8 are part of the acquired data in the experiment. From Figure 8(a), the actual MRT of different message instances varies obviously at random receptions. The WCMRT cannot reflect the variation of the actual MRT and will cause large error in most cases (up to 1.971 ms). However, if the eMRT is used to represent the actual MRT, the variation of MRT will be detected with certain accuracy (smaller than 0.4 ms). According to the estimation error shown in Figure 8(b), the absolute estimation error of eMRT is small and bounded, and the estimation error of eMRT is much smaller than that of WCMRT. For most samples, the estimation error is close to the average error which is about tens of microseconds. Only few samples will have an estimation error much larger than the average error, such as the 41st instance in Figure 8(b). These *imperfect samples* (for the sake of simplicity, we call the samples that have obvious larger error *imperfect samples*)

are always caused by the obvious variation of congestion situation. Under such conditions, the *blocking time* is likely to be close to the maximum value or minimum value. When we use the intermediate value to estimate it, the estimation error can be up to half the *transmission time* which is about 0.32 ms in the 250 Kbps system. This is why the estimation errors of the *imperfect samples* are usually close to half the *transmission time*. The quantitative values (acquired from the total experiment data set) of the evaluation parameters, which are described in Table 3, are provided in Table 4 from which we can see that the average estimation error is at about tens of microseconds and the error boundaries are at about ± 0.35 ms. Moreover, the small variance of estimation error indicates that the stability of estimation accuracy is good.

5.3. Evaluation of the General Applicability. In order to check whether the proposed method is applicable for different messages in a CAN system, we apply it to the messages with different priorities and different sending orders. When checking the effects of message priority, we apply it to the messages with priorities 4, 6, and 13, which have the same message sending order. When checking the effects of message sending order, we apply it to the message with priority 9 in different tests where it is sent by node 4 with sending orders 1, 2, and 4. The probability distributions of the estimation error of each message are given in Figures 9(a) and 9(b), from which we can see that the main estimation error ranges of each message are the same although the probability of each range has some differences. The estimation error of most samples is in the range from -0.05 ms to 0.05 ms. The probabilities of this range for each message are close to each other. According to Table 5, the worst estimation errors of each message are smaller than the boundaries (about ± 0.35 ms). The average estimation error and the variance of estimation error are also very small. So the proposed method is effective to estimate the MRT of any periodic message in a specific CAN system.

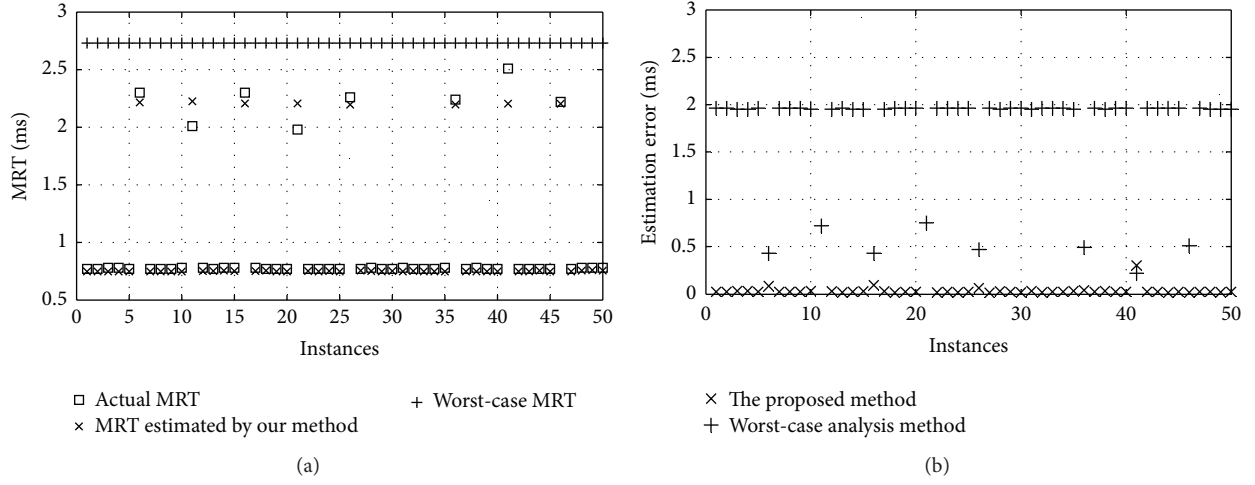


FIGURE 8: Comparison of the (a) MRTs and the (b) estimation errors obtained by our method and the worst-case analysis method [18] (where message priority = 4).

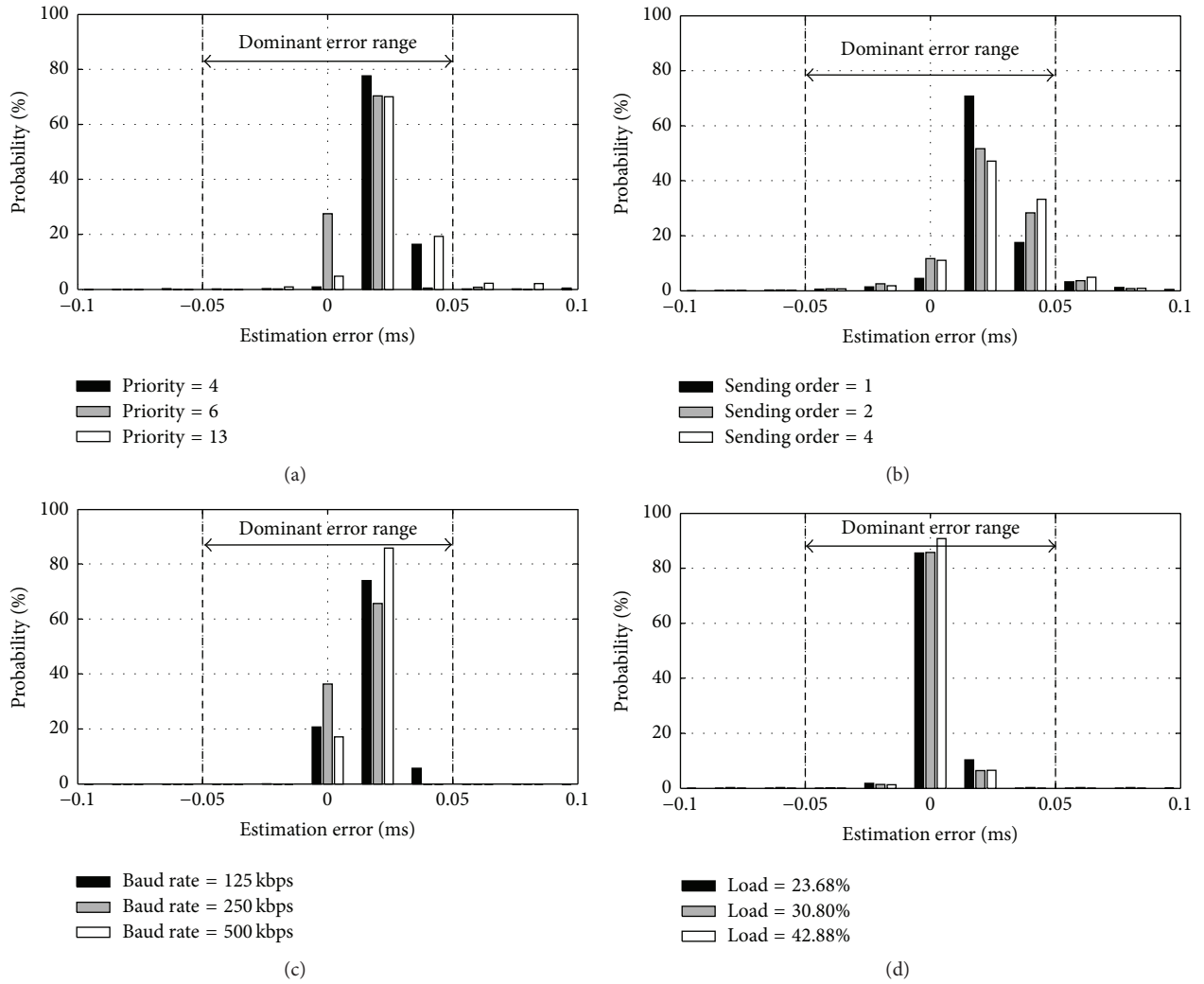


FIGURE 9: The effects of (a) message priority, (b) message sending order, (c) baud rate, and (d) bus load on the estimation accuracy.

TABLE 5: Quantitative values of the evaluation parameters used to assess the effects of message priority, message sending order, baud rate, and bus load on the estimation accuracy.

Parameter	Message priority				Message sending order				Baud rate (Kbps)				Bus load		
	4	6	13	1	2	4	125	250	500	23.68%	30.80%	42.88%			
e_{mp} (ms)	0.316	0.061	0.132	0.179	0.121	0.135	0.044	0.028	0.026	0.022	0.273	0.018			
e_{mn} (ms)	-0.220	-0.269	-0.008	-0.089	-0.073	-0.109	-0.010	-0.006	0	-0.020	-0.256	-0.016			
e_{exp} (ms)	0.025	0.014	0.025	0.025	0.026	0.024	0.017	0.011	0.014	1.086e - 3	1.627e - 3	9.260e - 4			
e_{va} (ms ²)	1.550e - 3	5.417e - 4	1.999e - 4	2.511e - 4	3.096e - 4	2.989e - 4	6.850e - 5	2.786e - 5	1.696e - 5	3.371e - 5	1.505e - 3	2.742e - 5			

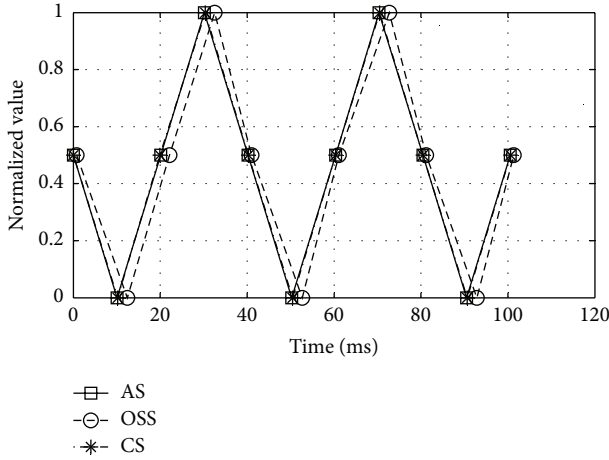


FIGURE 10: The ability of our method to eliminate the signal distortion and latency caused by MRT.

In order to check the general applicability of the proposed method to different applications, we apply it to the messages obtained from the CAN systems which have different baud rates and bus loads. When checking the effects of baud rate, we set the baud rate of the test CAN to be 125 Kbps, 250 Kbps, and 500 Kbps in three tests by modifying the software in each node, respectively. In each test, we apply the proposed method to the message with priority 4. When checking the effects of bus load, we change the bus load by removing some nodes from the test CAN. The average bus load is set to be 23.68%, 30.80%, and 42.88% in three tests, correspondingly. The corresponding transient peak bus loads are 31.77%, 63.84%, and 50.82%, respectively. According to the analysis of estimation error in Section 4, the maximum estimation error will reduce as the baud rate increases. However, according to the experimental results given in Figure 9(c) and Table 5, the average estimation error and overall distribution of estimation error are not affected by baud rate, that is, because the *imperfect samples* do not occur in the tests. From Figures 9(c) and 9(d) and Table 5, the average error and error distribution are all at the same level, and the maximum errors are all smaller than the boundary (about ± 0.35 ms). So we can conclude that the proposed method can be used in different applications.

5.4. Case Study on Condition Monitoring. In order to check whether the estimation accuracy of our method is acceptable in practical applications, we conduct a case study in this section. In the test, a triangular wave with unit amplitude and 25 Hz frequency is carried by the message to be analyzed which has priority 4. The original sampled signal (OSS) and the corrected signal (CS) are compared with the actual signal (AS) in Figure 10. The AS is the signal sent by the source node with the designed period, 10 ms. The OSS is the signal which uses the reception timestamp as the sending moment for each sample value. The CS is the signal which uses the moment obtained by subtracting the eMRT from the reception timestamp as the sending moment for each sample value. According to the recorded message traces,

the message instances that carry the sample data shown at 0 ms, 40 ms, 60 ms, 80 ms, and 100 ms in Figure 10 are not preempted by other message instances, while the others are preempted by two message instances. Due to the notable *interference* differences, the OSS is obviously distorted, as shown in Figure 10. However, by using the eMRT to correct the sampled signal, the distortion and the latency of sampled signal are greatly reduced. By comparing the CS with the AS, we can see that the distortion caused by the estimation errors is minor and can be ignored in practical applications.

6. Discussion

Compared with the MRT acquisition method given in chapter 7 of [26], our method is essentially different although both of these approaches estimate the MRT of CAN messages based on the analysis of message traces on CAN bus. Due to the different estimation ideas and analysis methods of message traces, our method can estimate the MRT in real time while that of [26] cannot. In [26], the MRT is estimated by reconstructing the arrival time (the start of MRT). In the reconstruction of arrival time, two message instances, which satisfy certain conditions and are separated by a period of time, are used as the references. Based on the reference message instances, the arrival time of message instances is reconstructed. Since the determination of two reference message instances is a prerequisite, the arrival time of the message instances between the reference message instances cannot be reconstructed at their receiving completion moments because the second reference message instance is unavailable (since it has not been received) at that time. Hence, the MRT cannot be estimated in real time by the method given in [26]. For our method, as the MRT is estimated by estimating each component of it, the arrival time needs not to be identified. The focus of message traces analysis changes from the reconstruction of arrival time to the identification of traffic congestion on CAN bus. As the message instances that may cause delay to message m must be transmitted before message m , we only need to analyze the message instances received before message m . Therefore, the MRT of a message instance can be estimated at its receiving completion moment by our method. Hence, our method can meet the real-time requirement of delay acquisition while that of [26] cannot.

In our method, the message traces analysis based on message block is the base. In practice, the constitution of message block may have various forms due to random traffic congestion and different sending model. In this paper, due to the limited space, we only discussed the general case in the case of making some assumptions. Nevertheless, our method can also be extended to the other cases. For example, in the message traces analysis given in Section 3, we assume that the periodic messages to be sent by a node during the same task cycle are sent without offset and the source node has at least one *base rate message*. If these assumptions are not satisfied, few changes can be made to validate our method. Specifically, if the periodic messages to be sent by a node during the same task cycle are sent with offset, the sending offset between m^i and $m^j_{j,r}$ and that between m_s and $m^{i-n}_{j,r}$

can be added to (2) and (4), respectively. If the source node has no *base rate message*, the message that has the minimum period in the source node can be used as a bridge and the MRT correlation can be derived by three similar steps where the corresponding sending offsets between message instances are added to (2) and (4), and the minimum message period in the source node is used in (3) instead of T_j . As the sending offsets between the periodic messages from the same node are always deterministic, they can be identified online. Therefore, the changes can be achieved.

In addition to the above assumptions, we also ignore the jitter of task cycle and the clock drift that will occur in practical applications. In well-designed NCSs, as the jitter of task cycle and the clock drift are usually strictly suppressed to make the message period as accurate as possible, they are unlikely to cause large error to the message period. Since the number of multiple instances in a message block is unlikely to be very big and the incremental estimation is unlikely to be continuously used for a long time in most cases, the jitter of task cycle and the clock drift are unlikely to cause large cumulative error to the estimated MRT in our method. If the estimation accuracy needs to be improved further, one can take the clock drift into consideration by using the actual task cycle, which can be obtained in a similar way as the acquisition of actual message period in [26], in (3) instead of T_j .

7. Conclusions

In this paper, we proposed a novel online real-time MRT estimation method for periodic CAN messages in order to help the real-time delay compensation in CAN-based NCSs where the exact network delay should be obtained in real time without increasing the bus load. Firstly, by analyzing the composition of MRT, we figured out that the MRT can be estimated in real time by estimating its components instead of reconstructing its start moment as the existing literature does. Then, by analyzing the message traces from the perspective of online real-time implementation, a reference message instance, based on which the idea of MRT estimation was put forward, was derived. The possible congestion situations of the reference message instance were also discussed. Then, the online real-time MRT estimation method, which estimated the MRT in two ways according to the actual congestion situation of the reference message instance, was proposed based on the previous analysis. Finally, experiments were conducted on a platform constructed according to the CAN system of a hybrid excavator to validate the proposed method. The estimation accuracy and the general applicability of the proposed method were checked. The experimental results showed that the MRT estimated by our method agreed well with the actual MRT. Future work includes applying the proposed method to practical CAN-based NCSs to improve the system performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 51475414, by the NSFC under Grant 51475422, and by the Science Fund for Creative Research Groups of NSFC under Grant 51221004.

References

- [1] R. A. Gupta and M.-Y. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
- [2] L. Zhang, H. J. Gao, and O. Kaynak, "Network-induced constraints in networked control systems—a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.
- [3] M. Tenruh, "Message scheduling with reduced matrix cycle and evenly distributed sparse allocation for time-triggered CAN," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1240–1251, 2011.
- [4] I. Park and M. Sunwoo, "FlexRay network parameter optimization method for automotive applications," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 4, pp. 1449–1459, 2011.
- [5] G. Cena, I. Cibrario Bertolotti, T. Hu, and A. Valenzano, "Fixed-length payload encoding for low-jitter controller area network communication," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2155–2164, 2013.
- [6] Z. Shuai, H. Zhang, J. Wang, J. Li, and M. Ouyang, "Lateral motion control for four-wheel-independent-drive electric vehicles using optimal torque allocation and dynamic message priority scheduling," *Control Engineering Practice*, vol. 24, no. 1, pp. 55–66, 2014.
- [7] L. Repele, R. Muradore, D. Quaglia, and P. Fiorini, "Improving performance of networked control systems by using adaptive buffering," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4847–4856, 2014.
- [8] Y. Shi and B. Yu, "Robust mixed H_2/H_∞ control of networked control systems with random time delays in both forward and backward communication links," *Automatica*, vol. 47, no. 4, pp. 754–760, 2011.
- [9] W. P. M. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nešić, "Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, 2010.
- [10] R. Yang, G.-P. Liu, P. Shi, C. Thomas, and M. V. Basin, "Predictive output feedback control for networked control systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 512–520, 2014.
- [11] R. Wang, G.-P. Liu, W. Wang, D. Rees, and Y. B. Zhao, "Guaranteed cost control for networked control systems based on an improved predictive control method," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1226–1232, 2010.
- [12] H. Xu, S. Jagannathan, and F. L. Lewis, "Stochastic optimal control of unknown linear networked control system in the presence of random delays and packet losses," *Automatica*, vol. 48, no. 6, pp. 1017–1030, 2012.
- [13] A. Kruszewski, W.-J. Jiang, E. Fridman, J. P. Richard, and A. Toguyeni, "A switched system approach to exponential stabilization through communication network," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 887–900, 2012.

- [14] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2011.
- [15] G. Cena, I. Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "Synchronize your watches. Part I. General-purpose solutions for distributed real-time control," *IEEE Industrial Electronics Magazine*, vol. 7, no. 1, pp. 18–29, 2013.
- [16] Z.-H. Pang and G.-P. Liu, "Design and implementation of secure networked predictive control systems under deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1334–1342, 2012.
- [17] Á. Cuenca, J. Salt, A. Sala, and R. Pizá, "A delay-dependent dual-rate PID controller over an ethernet network," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 1, pp. 18–29, 2011.
- [18] 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.
- [19] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "MPS-CAN analyzer: integrated implementation of response-time analyses for controller area network," *Journal of Systems Architecture*, vol. 60, no. 10, pp. 828–841, 2014.
- [20] H. B. Zeng, M. Di Natale, P. Giusto, and A. Sangiovanni-Vincentelli, "Stochastic analysis of CAN-based real-time automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 4, pp. 388–401, 2009.
- [21] H. B. Zeng, M. D. Natale, P. Giusto, and A. Sangiovanni-Vincentelli, "Using statistical methods to compute the probability distribution of message response time in controller area network," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 4, pp. 678–691, 2010.
- [22] P. M. Yomsi, D. Bertrand, N. Navet, and R. I. Davis, "Controller Area Network (CAN): response time analysis with offsets," in *Proceedings of the 9th IEEE International Workshop on Factory Communication Systems (WFCS '12)*, pp. 43–52, IEEE, Lemgo, Germany, May 2012.
- [23] H. Aysan, R. Dobrin, S. Punnekkat, and J. Proenza, "Probabilistic scheduling guarantees in distributed real-time systems under error bursts," in *Proceedings of the IEEE 17th International Conference on Emerging Technologies and Factory Automation (ETFA '12)*, vol. 50, pp. 1–9, Krakow, Poland, September 2012.
- [24] R. I. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Schedulability analysis for Controller Area Network (CAN) with FIFO queues priority queues and gateways," *Real-Time Systems*, vol. 49, no. 1, pp. 73–116, 2013.
- [25] I. Horvath and Z. Matyas, "Controller Area Network (CAN) worst-case message latency with priority inversion," U.S. Patent 0063120, March 5, 2015.
- [26] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol*, Springer New York, New York, NY, USA, 2012.
- [27] D. A. Khan, R. I. Davis, and N. Navet, "Schedulability analysis of CAN with non-abortable transmission requests," in *Proceedings of the IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA '11)*, pp. 1–8, IEEE, Toulouse, France, September 2011.
- [28] D. Wang, B. Yang, C. Zhu, J. Qi, and G. Ni, "Experimental study on communication delay of powertrain system of plug-in hybrid electric vehicles," *Mathematical Problems in Engineering*, vol. 2013, Article ID 758961, 13 pages, 2013.
- [29] Robert Bosch GmbH, *CAN Specification Version 2.0*, Robert Bosch GmbH, 1991.
- [30] D. Gao and Q. Wang, "Health monitoring of controller area network in hybrid excavator based on the message response time," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '14)*, pp. 1634–1639, IEEE, July 2014.

