



Message delay time distribution analysis for controller area network under errors^{*#}

Lei-ming ZHANG, Yi-chao SUN, Yong LEI[‡]

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

E-mail: lmzhang@zju.edu.cn; syc_best@163.com; ylel@zju.edu.cn

Received Dec. 8, 2017; Revision accepted May 13, 2018; Crosschecked June 11, 2019

Abstract: Controller area network (CAN) is a widely used fieldbus protocol in various industrial applications. To understand the network behavior under errors for the optimal design of networked control systems, the message response time of the CAN network needs to be analyzed. In this study, a novel delay time distribution analysis method for the response messages is proposed when considering errors. In this method the complex message queues are decomposed into typical message patterns and cases. First, a stochastic fault model is developed, and the probability factor is defined to calculate the error distribution. Then the message delay time distribution for the single slave node configuration is analyzed based on the error distribution. Next, based on the delay time distribution analysis of typical patterns and cases, an analysis framework of message delay time distribution for the master/slave configuration is developed. The testbed is constructed and case studies are conducted to demonstrate the proposed methodology under different network configurations. Experimental results show that the delay time distributions calculated by the proposed method agree well with the actual observations.

Key words: Controller area network; Message delay; Probability distribution; Errors

<https://doi.org/10.1631/FITEE.1700815>

CLC number: TP277

1 Introduction

Since its standardization in the 1980s, the controller area network (CAN) protocol has gained wide acceptance across various applications, including vehicle systems, networked automation systems, and more recently, airplane sensor-actuator systems (Farsi et al., 1999). However, the probability of a CAN system suffering from faults increases as the complexity of the network topologies and system functions grow rapidly. Errors in the system contributed by factors such as vibration, electromagnetic interference, and the effects of aging on the

cables can cause delay when transmitting messages, which will result in the degradation of the network performance and a decrease in the stability of the overall system. To achieve a robust design of the CAN network for an optimal bandwidth usage and evaluate the real-time performance of the networked control systems, the research on network behaviors in terms of message queues and message response time distributions when considering errors is of great importance.

In the literature, message response time analysis for a CAN network has drawn attention since the early 1990s. Since it was difficult to exactly calculate the message response time, the worst-case response time (WCRT) was studied extensively. Tindell et al. (1995b) proposed a method to estimate the bound of WCRT of a given message. The WCRT analysis method has been extended to the conditions of considering errors, fixed priority, different controllers,

[‡] Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 51475422 and 51521064)

A preliminary version was presented at the Chinese Automation Congress, China, November 27–29, 2015

ORCID: Yong LEI, <http://orcid.org/0000-0003-0235-5203>

© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2019

and other circumstances (Tindell and Burns, 1994; Tindell et al., 1994, 1995a). Navet et al. (2000) calculated the worst-case deadline failure probability (WCDFP) considering random errors other than deterministic errors for the CAN network. Davis et al. (2007) improved the WCRT analysis method by resolving the flaws with the original schedulability analysis for the CAN messages. The WCRT analysis and optimal priority assignment policies have been introduced (Davis et al., 2011, 2013; Davis and Navet, 2012), under different constraints such as first in first out (FIFO) queueing policy and arbitrary deadlines. Broster et al. (2005) demonstrated the unreasonable and conservative aspects of the error model proposed in Navet et al. (2000), and conducted the WCRT analysis of messages under a Poisson distribution fault model. Mubeen et al. (2014, 2015) calculated the WCRT for periodic, sporadic, and mixed messages in CAN networks by integrating the effect of hardware and software limitations in the CAN controller. Yomsi et al. (2012) proposed an extendible framework built upon the transaction model to analyze the WCRT of non-preemptive CAN frames with offsets. However, the existing WCRT method is too conservative in practice such that it will result in over-design and under-utilization of network resources and bandwidth.

The message response time analysis from the perspective of probability and statistics for the CAN network has been studied in the literature. Kumar et al. (2009) proposed a response time distribution analysis method for CAN networks based on the deterministic stochastic petri net (DSPN), where the message response time distributions in different priorities were analyzed and the results were compared with that of worst-case analysis. Chen et al. (2012) developed a non-preemptive priority M/G/1 model for the response time of CAN messages, and calculated the distribution function of bit-stuffing and the mean value of the response time. Zeng et al. (2010) proposed a method to compute message response time probability distribution using statistical analysis when only partial information was available on the bus and the assigned message priorities were given.

As can be seen from the literature, although existing probability distribution approaches have been developed to analyze the response time of CAN messages, the analysis of the CAN message response time

distribution considering errors on the bus has not been conducted. In practical industrial systems, the queueing of the messages and the error interruptions are inevitable in the CAN network, which will in turn make the design of a CAN network complex. To take full advantage of the bus bandwidth and ensure that the message transmissions satisfy the real-time requirements of a system, it is needed to develop a new message response time analysis methodology that considers both message arbitrations and error interruptions.

The purpose of this study is to develop a delay time distribution analysis method for the response messages in the CAN network under errors. The advantages of this study are as follows: First, the proposed method considers the interaction of both message arbitrations and error interruptions on the message delay time analysis, which provides a deep insight into the CAN behaviors. Second, in this framework, the complex message queues are decomposed into typical message patterns and cases, which simplifies the analysis procedure and makes it straightforward to calculate the delay time distribution of messages for the master/slave configuration based on the observations of a practical single-node network. The results of this work will enable system engineers to predict and estimate the performance of a network system in its design stage based on the message response time analysis of several basic nodes, and to better understand the network behaviors with stochastic errors, which will ultimately lead to optimal design of the networked control systems.

2 Problem definition

In a polling based CAN network, the transmission time of a message M from a slave node depends on the message queue on the bus (Sun et al., 2015), as shown in Fig. 1.

The top panel of Fig. 1 shows the scenario where only one master device and one slave node are on the bus and the bus is error-free. M_P denotes the request frame sent from the master device to the slave node, and M denotes the response message sent from the slave node to the master device without error interruptions. In this scenario, the response time of message M , R_M , is calculated as

$$R_M = J_M + C_M, \quad (1)$$

where C_M and J_M respectively denote the transmission time and the jitter time of message M when the bus is error-free. Since the variation of the jitter time is considerably small compared with the response time R_M , the expectation of J_M is adopted in this study.

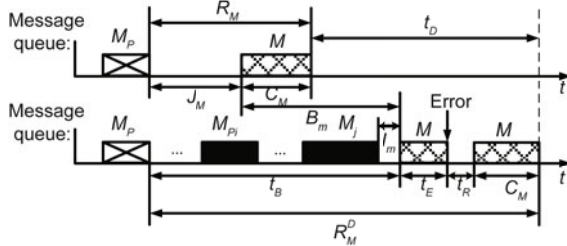


Fig. 1 Analysis of the delay time for a message M

The bottom panel of Fig. 1 shows the scenario where the transmission of message M is delayed by the blocking of other messages and the error interruptions on the bus (Hansson et al., 2002). M_{Pi} and M_j are the examples of other messages that block the transmission of M , where M_{Pi} denotes the message sent from the master device to other slave nodes, and M_j denotes the message sent from other slave nodes. I_m is the shortest frame interval defined in the CAN protocol (Bosch, 1991) that distinguishes two consecutive messages, which is equal to the transmission time of seven consecutive recessive bits in this work. In this scenario, the response time of message M , R_M^D , is calculated as

$$\begin{aligned} R_M^D &= t_B + t_E + t_R + C_M \\ &= t_B + E_m + C_M, \end{aligned} \quad (2)$$

where t_B is the time interval between M_P successfully transmitted and message M starting to transmit, and E_m is the delay time caused by error interruptions. E_m contains two parts: The first part t_E is the already transmitted time of the uncompleted message when error interruption occurs, and the second part t_R is the bus recovery time which is equal to the transmission time of 24 bits (Bosch, 1991).

Then the delay time contributed by message blocking and error interruptions can be calculated as

$$\begin{aligned} t_D &= R_M^D - R_M \\ &= \begin{cases} E_m, & t_B \leq J_M, \\ t_B - J_m + E_m = B_m + E_m, & t_B > J_m, \end{cases} \end{aligned} \quad (3)$$

where B_m denotes the blocking time caused by other messages.

Therefore, to determine the delay time distribution of message M , the following challenges must be addressed:

1. How should the error interruptions occurring on the bus be modeled, and how is the delay time distribution of message calculated when error interruptions occur at different parts of the message?

2. How is the delay time distribution of message calculated when both message blocking and error interruptions on the bus are considered?

The assumptions in this study are as follows: The communication mode of the CAN bus is polling with one master device, and the information about message sequences without errors can be obtained by practical observation or simulation.

3 Analysis methodology

3.1 Overall analysis framework

The basic idea of the proposed method is that by analyzing the message sequences, one can decompose complex message arbitrations induced and error interruptions induced message delays into simple patterns and cases. The overall procedure for the proposed method is shown in Fig. 2.

First, a stochastic fault model is developed to describe the faults occurring in a CAN network. Then we introduce the probability factor to express the causal relationship between the intermittent connection (IC) fault and the resulting error interruptions on the bus. Third, we conduct a detailed analysis of the message delay for the single slave node configuration, which includes calculating the probability of different numbers of error interruptions occurring during the message transmission, obtaining the probability density functions (PDFs) of the delay time corresponding to the different numbers of error interruptions, and determining the delay time distribution by calculating the expectation of all these PDFs. Finally, we introduce typical patterns and cases of the message sequence for the master/slave configuration, and present a framework to analyze the delay time distribution for the master/slave configuration considering both message blocking and error interruptions. Details of the proposed method are introduced in the following subsections.

3.2 Stochastic fault model

During the virtual industrial production and control process, the faults that a CAN system suffers from the surrounding environment are complex and stochastic. Among the various kinds of faults, we consider the IC fault in this study. An IC fault causes short-term abnormal or chronically instable connection between the node module and network bus. If the IC fault occurs during the transmission of a dominant bit, this dominant bit will turn into a recessive bit, which causes error interruption on the CAN system (Lei et al., 2015).

The arrivals of the IC fault follow a Poisson process (Lei et al., 2014), and the arrival rate is λ . Therefore, the probability that the number of IC faults arriving in any interval of length t is equal to

n can be obtained as follows:

$$P(t, n) = P\{N(t) = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, n = 0, 1, \dots \quad (4)$$

3.3 Error handling of CAN and probability factor calculation

3.3.1 CAN error handling mechanisms

There are five different error types that can be detected in the CAN network: bit error, stuff error, cyclic redundancy check (CRC) error, format error, and acknowledgement (ACK) error. Any node that detects an error condition will send an error flag to interrupt the transmission on the bus. Table 1 shows the cause of each type of error, and the output time sequence of the error flag defined in the CAN protocol for each type of error. For example, if an IC fault affects one dominant bit that leads to a CRC validation error, the error flag will occur and generate an error interruption after the ACK delimiter, rather than after this affected dominant bit.

3.3.2 Calculation of probability factor α

Although the arrivals of an IC fault have been modeled, the distribution of the error interruptions on the bus is difficult to determine. The reasons are twofold: First, the IC fault can affect only the dominant bits of the frame, and it cannot lead to an error interruption when the bus is idle or when the bus is transmitting the recessive bits. Therefore, an arrival of an IC fault may not result in an error interruption on the bus. Second, when the dominant bits are affected by the IC fault, it may not immediately result in an error interruption.

Considering the aforementioned two reasons, a probability factor α is developed to describe the relationship between the IC fault arrivals and the resulting error interruptions on the bus. According to

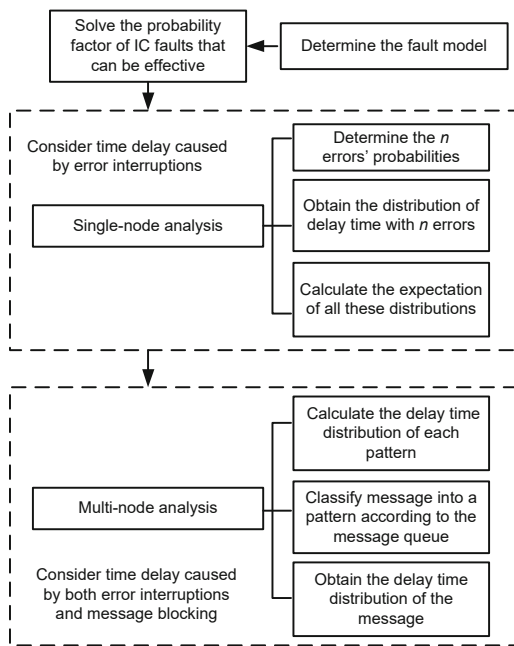


Fig. 2 Overall framework of the delay time analysis procedure

Table 1 Output time sequences of error flag for different error types in the CAN protocol

Error type	Cause of the error	Time sequence of error flag
Bit error	Output level does not agree with the bus monitoring level	Output error flag next to the bit after the error is detected
Stuff error	Six consecutive equal bit levels are detected	
Format error	Level detected does not agree with the fixed bit format	
ACK error	Level in ACK slot of sending unit is recessive	
CRC error	CRC calculated does not agree with that received	Output error flag next to the bit after ACK delimiter

the CAN protocol, a message M can be divided into several segments (Fig. 3).

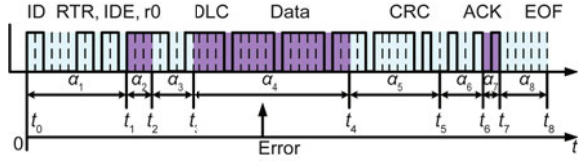


Fig. 3 Probability factor α for different frame segments of message M

In every segment of M , the probability factor α_i represents the probability that the arrival of an IC fault can effectively cause an error interruption, which can be obtained by

$$\alpha_i(M) = \frac{n_{di}}{n_i} + \sum_{\forall j \in E(ji)} \frac{n_{dj}}{n_j}, \quad (5)$$

where n_i or n_j is the number of all bits on each corresponding segment, and n_{di} or n_{dj} is the number of dominant bits that can be effective on each corresponding segment. The set $E(ji)$ represents all scenarios in which the IC fault arrives at segment j but causes an error interruption on segment i .

The calculation of $\alpha_i(M)$ includes two parts: the first part denotes the probability that the IC fault arrival is effective and causes an error interruption in segment i , and the second part denotes the probability that the IC fault arrival is effective on segment j but causes an error interruption in segment i . The expectation of $\alpha_i(M)$ can be calculated by

$$\bar{\alpha}(M) = E[\alpha_i(M)] = \sum_{i=1}^8 \frac{n_i}{n_{\text{total}}} \alpha_i(M), \quad (6)$$

where n_{total} is the total number of bits of message M .

3.4 Message delay analysis for the single slave node configuration

In this subsection, we introduce the message delay analysis for the single slave node configuration, which serves as the fundamental building block of this work. In the single slave node configuration, only one master device and one slave node are on the bus, the slave node transmits the response message as long as the master device sends the request message, and there are no other messages affecting the transmission. Hence, only the error interruptions can cause delay time for the transmission of the response message sent from the slave node.

Let M_{Pk} denote the request message sent from the master device to node k , and M_{Nk} the response message sent from slave node k . Thus, the only message sequence observed on the bus is (M_{Pk}, M_{Nk}) . As the IC faults may arrive continuously in time, different numbers of error interruptions should be analyzed separately. Two scenarios for different numbers of error interruptions on the bus are shown in Fig. 4, where the top panel and bottom panel are the message sent by the slave node interrupted by one error and two successive errors, respectively. The scenario that more error interruptions on the bus can be analyzed similarly.

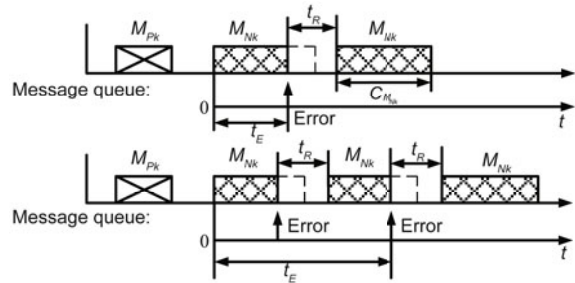


Fig. 4 Message queues under different numbers of error interruptions

The probability that different numbers of IC faults arrive during the transmission of M_{Nk} can be calculated by Eq. (4), and the probability factor for message M_{Nk} , $\bar{\alpha}(M_{Nk})$, can be calculated by Eq. (6). Then the probability that n error interruptions occur during the transmission interval of M_{Nk} can be calculated by

$$\begin{aligned} P_{M_{Nk}}^{(n)} &= P\{n \text{ error interruptions during } C_{M_{Nk}}\} \\ &= [(1 - P(C_{M_{Nk}}, 0))\bar{\alpha}(M_{Nk})]^n \cdot [P(C_{M_{Nk}}, 0) \\ &\quad + (1 - P(C_{M_{Nk}}, 0))(1 - \bar{\alpha}(M_{Nk}))], \end{aligned} \quad (7)$$

where $C_{M_{Nk}}$ denotes the transmission time of M_{Nk} when the bus is error-free.

According to the Poisson process theory, the PDF for the n^{th} IC fault arrival is given as follows:

$$f_{S_n}(t) = \lambda e^{-\lambda t} \frac{(\lambda t)^{n-1}}{(n-1)!}. \quad (8)$$

Then the probability density function of t_E corresponding to the IC fault arriving at different frame

segments can be obtained:

$$f_{M_{Nk}}^{(n)}(t_E) = \alpha_i(M_{Nk})f_{S_n}(t_E) = \begin{cases} \alpha_1(M_{Nk})f_{S_n}(t_E), & t_0 \leq t_E < t_1, \\ \alpha_2(M_{Nk})f_{S_n}(t_E), & t_1 \leq t_E < t_2, \\ \vdots \\ \alpha_8(M_{Nk})f_{S_n}(t_E), & t_7 \leq t_E \leq t_8. \end{cases} \quad (9)$$

Finally, the delay time PDF of M_{Nk} can be obtained by calculating the expectation of all these PDFs (note that B_m is zero since there is no competition in the single node case):

$$f_{M_{Nk}}(t_D) = f_{M_{Nk}}(E_m) = f_{M_{Nk}}(t_E + t_R) = \sum_{i=1}^n P_{M_{Nk}}^{(i)} f_{M_{Nk}}^{(i)}(t_D - t_R). \quad (10)$$

Furthermore, the cumulative distribution function (CDF) of delay time t_D is obtained:

$$F_{M_{Nk}}(t) = \int_{-\infty}^t f_{M_{Nk}}(t_D) dt_D. \quad (11)$$

According to the stochastic Poisson fault model, the number of IC fault arrivals in any time interval can be infinite theoretically; thus, the number of error interruptions during $C_{M_{Nk}}$ can be infinite. However, the scenarios in which more than two consecutive error interruptions occur are rare according to actual observations. Therefore, a probability threshold value ψ can be used to limit the number of error interruptions which need to be considered. If $P_i < \psi$, then such an i -error-interruption case can be ignored.

4 Message delay analysis for the master/slave configuration

When the single node case is extended to the master/slave case, the contention between messages from multiple nodes is inevitable. Thus, the delay time is caused not only by error interruptions, but also by the blocking time of other messages. Since there are various message sequences on the bus in the master/slave configuration, the complexity of the analysis for multiple slave nodes is obviously increased. In this section, as shown in Table 2, we introduce the typical patterns and cases of the message sequence in the master/slave configuration, based on which we can analyze the message delay time for the master/slave configuration, and the details of the patterns and cases are introduced in the following subsections.

4.1 Pattern I: no arbitration contention between messages

Similar to the single slave node configuration, let M_{Pk} denote the request message sent from the master device to node k , and M_{Nk} the response message sent from node k . In pattern I, message M_{Nk} has no arbitration contention with other messages, but may still be influenced by the transmission delay of the messages ahead of M_{Nk} . As analyzed in the single slave node configuration, there are at most two consecutive error interruptions that are statistically significant. Therefore, the maximum delay time caused by single node message under errors, $\max(E_m)$, is given by

$$\max(E_m) = \max(t_E) + t_R = 2(C_M + t_R), \quad (12)$$

where M is the message ahead of M_{Nk} and C_M is the transmission time of M when the bus is error-free. Then by comparing the interval time between

Table 2 Typical patterns and cases of the message sequence for the master/slave configuration

Pattern	Case(s) and scenario(s)
I: no arbitration contention	Case A: $t_{\text{interval}} \gg \max(E_m)$
	Case B: $t_{\text{interval}} \ll \max(E_m)$ Scenario 1: M_{Nk} is interrupted Scenario 2: M is interrupted
	Case C: $t_{\text{interval}} \approx \max(E_m)$ Scenario 1: M_{Nk} is interrupted Scenario 2: M is interrupted
II: existing arbitration contention	Scenario a: M_{Nk} wins the arbitration Scenario b: M_{Nk} loses the arbitration

M and M_{Nk} , i.e., t_{interval} with $\max(E_m)$, we can obtain three cases in pattern I.

4.1.1 Case A of pattern I

In this case, t_{interval} is considerably greater than $\max(E_m)$, where M_{Nk} is not affected by M even if M is interrupted by error interruptions. Fig. 5 shows the message queue and the error interruptions in case A.

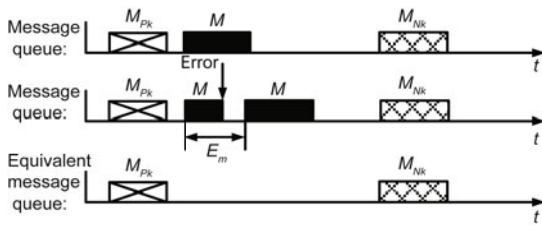


Fig. 5 Delay time analysis of M_{Nk} in case A of pattern I

As shown in the bottom panel of Fig. 5, the delay time distribution of M_{Nk} in case A is equivalent to that in the single slave node configuration. Therefore, the delay time PDF considering error interruptions of M_{Nk} in case A is

$$f_{M_{Nk}}^{\text{case A}}(t_D) = f_{M_{Nk}}(E_m) = f_{M_{Nk}}(t_E + t_R) = \sum_{i=1}^n P_{M_{Nk}}^{(i)} f_{M_{Nk}}^{(i)}(t_D - t_R). \quad (13)$$

4.1.2 Case B of pattern I

In this case, t_{interval} is equal to the shortest frame interval I_m , which is far smaller than $\max(E_m)$. In case B, if the error interruptions occur during the transmission of M and cause delay time E_m for M , then the delay time E_m will affect the subsequent message M_{Nk} and make the delay time of M_{Nk} equal to E_m . The message queue and the error interruptions in case B are shown in Fig. 6.

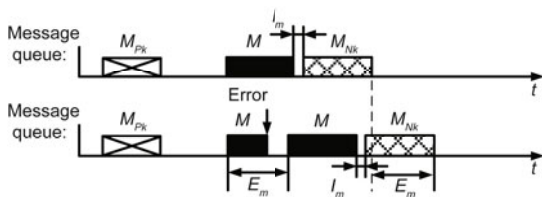


Fig. 6 Delay time analysis of M_{Nk} in case B of pattern I

Based on the locations where the error interruptions occur, case B can be decomposed into two scenarios:

1. Scenario 1 in case B of pattern I

In this scenario, the error interruptions occur during the transmission of M_{Nk} ; thus, the delay time PDF of M_{Nk} is the same as the result obtained in the single slave node configuration, as shown below:

$$f_{M_{Nk}}^{(s1)}(t_D) = f_{M_{Nk}}^{\text{case A}}(t_D). \quad (14)$$

2. Scenario 2 in case B of pattern I

In this scenario, the error interruptions occur during the transmission of M and cause delay time E_m for M . Then the transmission of M_{Nk} is blocked by the delay of M , and the blocking time B_m is equal to E_m . Therefore, the delay time PDF of M_{Nk} in this scenario can be described by

$$f_{M_{Nk}}^{(s2)}(t_D) = f_{M_{Nk}}(B_m) = f_M(E_m) = \sum_{i=1}^n P_M^{(i)} f_M^{(i)}(t_D - t_R). \quad (15)$$

To calculate the delay time distribution of M_{Nk} in case B, the percentage of each scenario appearing on the bus should be determined. Since the two scenarios are caused by the different locations where error interruptions occur, and the arrivals of an IC fault are assumed to follow a Poisson process, the percentage for each scenario can be obtained by combining the length of each message and the corresponding probability factor. The probability of scenario 1 appearing on the bus is given by

$$\beta = \frac{C_{M_{Nk}} \bar{\alpha}(M_{Nk})}{C_{M_{Nk}} \bar{\alpha}(M_{Nk}) + C_M \bar{\alpha}(M)}, \quad (16)$$

where $C_{M_{Nk}}$ and C_M denote the transmission time of M_{Nk} and M respectively, and $\bar{\alpha}(M_{Nk})$ and $\bar{\alpha}(M)$ denote the probability factors of M_{Nk} and M respectively.

Thus, the probability of scenario 2 appearing on the bus is $(1 - \beta)$. By combining scenarios 1 and 2, the delay time PDF of M_{Nk} in case B can be obtained:

$$f_{M_{Nk}}^{\text{case B}}(t_D) = \beta f_{M_{Nk}}^{(s1)}(t_D) + (1 - \beta) f_{M_{Nk}}^{(s2)}(t_D). \quad (17)$$

4.1.3 Case C of pattern I

In this case, t_{interval} is in the range of $\max(E_m)$. Thus, if the error interruptions occur during the

transmission of M and cause delay time E_m for M , the delay time E_m will affect the subsequent message M_{Nk} , but the resulting delay time of M_{Nk} is different from that in case B. Fig. 7 shows the message queue and the error interruptions in case C.

Similar to the analysis in case B, case C can be decomposed into two scenarios based on the locations where the error interruptions occur.

1. Scenario 1 in case C of pattern I

In this scenario, the error interruptions occur during the transmission of M_{Nk} ; thus, the delay time PDF of M_{Nk} is the same as the result obtained in the single slave node configuration, shown as follows:

$$f_{M_{Nk}}^{(s1)}(t_D) = f_{M_{Nk}}^{\text{case A}}(t_D). \quad (18)$$

2. Scenario 2 in case C of pattern I

In this scenario, the error interruptions occur during the transmission of M and cause delay time E_m for M . As shown in the middle panel of Fig. 7, the location of error interruption on M is on the front of the whole message, then the retransmission of M has no influence on the transmission of M_{Nk} , and the block time B_m is zero. On the other hand, as shown in the bottom panel of Fig. 7, the location of error interruption on M is at the rear of the whole message, then the repeated transmission of M blocks the transmission of M_{Nk} , and the resulting block time for M_{Nk} is $B_m = E_m - (T_k - I_m)$, where T_k denotes the interval time between M_{Nk} and M , which can be obtained by analyzing the message response time of different single nodes when the bus is error-free. Therefore, the delay time PDF of M_{Nk} in this scenario can be obtained by

$$f_{M_{Nk}}^{(s2)}(t_D) = f_{M_{Nk}}(B_m) = f_M(E_m - T_k + I_m),$$

$$= \begin{cases} 0, & t_D \leq 0, \\ \sum_{i=1}^n P_M^{(i)} \cdot f_M^{(i)}(t_D - t_R + T_k - I_m), & t_D > 0. \end{cases} \quad (19)$$

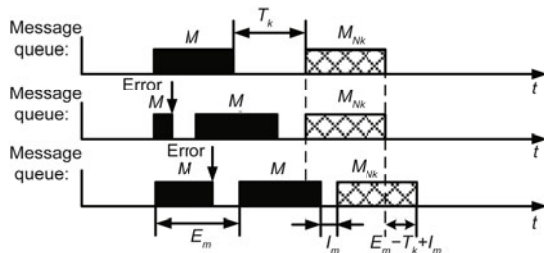


Fig. 7 Delay time analysis of M_{Nk} in case C of pattern I

In case C, the percentage of each type of the two scenarios appearing on the bus can also be calculated by Eq. (16). Therefore, the delay time PDF of M_{Nk} in case C can be obtained by combining the two scenarios:

$$f_{M_{Nk}}^{\text{case C}}(t_D) = \beta f_{M_{Nk}}^{(s1)}(t_D) + (1 - \beta) f_{M_{Nk}}^{(s2)}(t_D). \quad (20)$$

4.2 Pattern II: arbitration contention existing between messages

In pattern II, there are arbitration contentions between message M_{Nk} and other messages. In this subsection, we present an analysis procedure for the situation where there is an arbitration contention between M_{Nk} and message M . Thus, the situation where there are arbitration contentions between M_{Nk} and multiple messages can be analyzed in a similar way.

Fig. 8 shows two different message queues on the bus where a different message wins the arbitration and the interval time between M_{Nk} and M in both of the message queues is the shortest frame interval I_m . To analyze the delay time distribution of M_{Nk} in pattern II, the message queues shown in Fig. 8 can be decomposed into two scenarios based on the arbitration results of the messages.

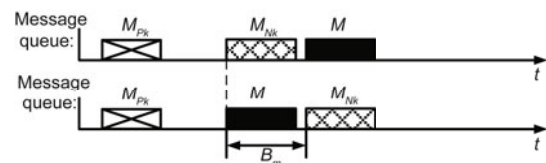


Fig. 8 Message analysis of M_{Nk} in pattern II

4.2.1 Scenario a of pattern II

As shown in the top panel of Fig. 8, message M_{Nk} wins the arbitration and starts the transmission first. In this scenario, message M has no influence on the transmission of M_{Nk} , and the delay time distribution of M_{Nk} is up to the message sequence ahead of M_{Nk} , which is one of the three cases shown in pattern I. As long as the message sequence ahead of M_{Nk} is determined, the delay time PDF of M_{Nk} considering error interruptions in scenario a can be obtained:

$$f_{M_{Nk}}^{(sa)}(t_D) = f_{M_{Nk}}^{\text{case } u}(t_D), \quad u \in \{A, B, C\}. \quad (21)$$

4.2.2 Scenario b of pattern II

As shown in the bottom panel of Fig. 8, message M_{Nk} loses the arbitration and is blocked by message M . In this scenario, the block time B_m caused by M is the sum of the transmission time C_M and the jitter time J_M of message M .

If the error interruptions occur during the transmission of M_{Nk} , then the delay time PDF of M_{Nk} is given by

$$\begin{aligned} f_{M_{Nk}}^{(sb1)}(t_D) &= f_{M_{Nk}}(E_m + B_m) \\ &= \sum_{i=1}^n P_{M_{Nk}}^{(i)} f_{M_{Nk}}^{(i)}(t_D - t_R - B_m). \end{aligned} \quad (22)$$

On the other hand, if the error interruptions occur during the transmission of M , similar to the analysis in case B of pattern I, the delay time PDF of message M is $f_M^{\text{case B}}$. Since M_{Nk} is blocked by message M with block time B_m , the delay time PDF of M_{Nk} can be calculated by

$$f_{M_{Nk}}^{(sb2)}(t_D) = f_M^{\text{case B}}(t_D - B_m). \quad (23)$$

Then the delay time PDF of M_{Nk} in scenario b can be obtained:

$$f_{M_{Nk}}^{(sb)}(t_D) = \beta f_{M_{Nk}}^{(sb1)}(t_D) + (1 - \beta) f_{M_{Nk}}^{(sb2)}(t_D), \quad (24)$$

where β and $(1 - \beta)$ denote the percentages of error interruptions occurring during the transmission of M_{Nk} and M , respectively, which can be calculated by Eq. (16).

To calculate the delay time distribution of M_{Nk} in pattern II, the probability of each scenario appearing on the bus should be determined. The probability of scenario a in which message M_{Nk} wins the arbitration appearing on the bus is given by

$$\gamma = \sum_{t_k \leq t_v} \phi_N(t_k) \phi_M(t_v), \quad (25)$$

where $\phi_N(t_k) = P\{t = t_k\}$ ($k \in \{1, 2, \dots\}$) denotes the discrete time distribution of the time interval between M_{Pk} and M_{Nk} , and $\phi_M(t_v) = P\{t = t_v\}$ ($v \in \{1, 2, \dots\}$) denotes the discrete time distribution of the time interval between M_{Pk} and M . $\phi_N(t_k)$ and $\phi_M(t_v)$ can be measured from practical observations when the bus is error-free. Therefore, the probability of scenario b in which message M_{Nk} loses the arbitration appearing on the bus is $(1 - \gamma)$.

By combining scenarios a and b, the delay time PDF of M_{Nk} of pattern II can be obtained:

$$f_{M_{Nk}}(t_D) = \gamma f_{M_{Nk}}^{(sa)}(t_D) + (1 - \gamma) f_{M_{Nk}}^{(sb)}(t_D). \quad (26)$$

4.3 Message delay analysis procedure for the master/slave configuration

Based on the two typical patterns presented above, the message delay time distribution for the master/slave configuration can be analyzed in the following steps:

Step 1: Classify the message of interest into the corresponding pattern (pattern I or II) and case (case A, B, or C of pattern I) based on its practical location in the message queue when the bus is error-free.

Step 2: Analyze the delay time distribution of the message of interest according to the corresponding pattern and case methods shown in Sections 4.1 and 4.2, and obtain the delay time PDF of the message of interest.

Step 3: Based on the delay time PDF, the CDF of the delay time for the message of interest can be obtained by

$$F_{M_{Nk}}(t) = \int_{-\infty}^t f_{M_{Nk}}(t_D) dt_D. \quad (27)$$

5 Testbed setup and case studies

To illustrate the procedure for analyzing message delay time distribution, a testbed was constructed and two case studies were conducted. In case study 1, we analyzed the delay time distribution for the single slave node configuration, and compared it with practical observations. Furthermore, in case study 2, we analyzed the delay time distribution of each response message for the master/slave configuration, and compared it with the corresponding practical observations.

5.1 Testbed setup

The schematic layout of the experimental setup is illustrated in Fig. 9, and the constructed testbed, which consists of three modules, for the case studies is shown in Fig. 10. The first module is the DeviceNet network, which uses CAN as its physical layer and the data link layer protocol. The communication mode was set as polling and the communication speed is 500 kb/s; thus, the

transmission time of one bit, i.e., τ_{bit} , is 2 μs . Therefore, $t_R = 24\tau_{\text{bit}} = 48 \mu\text{s}$, and $I_m = 7\tau_{\text{bit}} = 14 \mu\text{s}$.

The second module is the in-house developed fault injection system. A controlled high-speed on-off switch was applied to generate the IC faults occurring on the cable. The arrivals of the IC fault follow a Poisson process with the arrival rate λ_{IC} . The third module is the CAN-bus analyzer based data acquisition module, which records the data link layer information on the bus.

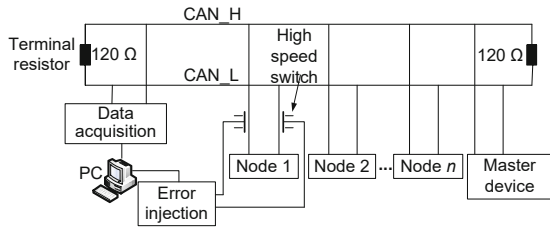


Fig. 9 Schematic layout of the testbed

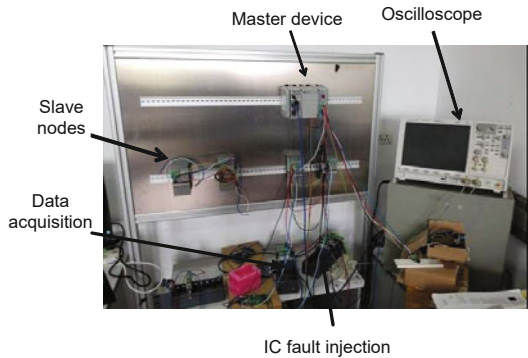


Fig. 10 Constructed testbed for case studies

5.2 Case study 1: delay time analysis for the single slave node configuration

In this case study, there are two nodes on the bus: the master device and the slave node with address 9. The drop cable of node 9 was set to experience the IC faults at an injection rate of $\lambda_{\text{IC}} = 1000$ faults/s. The bit stream information of the response message sent from node 9 when the bus was error-free is shown in Fig. 11, where “0” and “1” represent the dominant bit and recessive bit, respectively, and the bit pointed to by the triangle is the stuffing bit.

Then the probability factor α_i can be obtained by Eq. (5), which is shown in Table 3.

Finally, the delay time cumulative distribution

of the response message sent from node 9 can be obtained, which is plotted by the solid line in Fig. 12. Moreover, the practical observation of the delay time is plotted by the dashed line in Fig. 12.

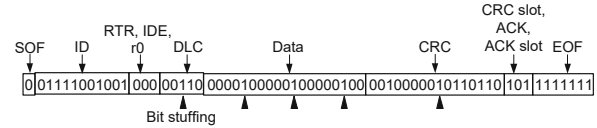


Fig. 11 Bit stream information of the message sent from node 9 when the bus is error-free

Table 3 Probability factor α_i for the single node case

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8
0.033	0.046	0.085	0.097	0.207	0.346	0.751	0.267

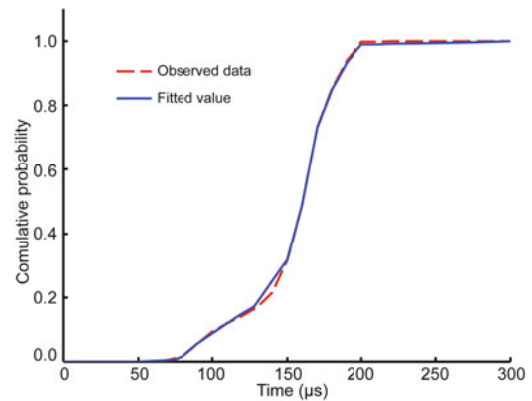


Fig. 12 Delay time comparison between the fitted value and practical observation in case study 1 (root mean square error: 0.0083; maximum absolute error: 0.0361)

As can be seen from Fig. 12, the delay probability distribution calculated by the proposed method agrees well with the practical observation. The curve begins after 50 μs due to the existence of the bus recovery time, and becomes steep after 150 μs because most error interruptions occur near the CRC frame check field according to the CAN protocol.

5.3 Case study 2: delay time analysis for the master/slave configuration

In this case study, there are 10 nodes on the bus: the master device and the slave nodes whose addresses are from one to nine. The drop cable of node 5 was set to experience the IC faults at an injection rate of $\lambda_{\text{IC}} = 334$ faults/s. The physical

waveform graph of the messages transmitted on the bus is shown in Fig. 13, where the message queue from left to right is (M_{P1} , M_{P2} , M_{N1} , M_{P3} , M_{N2} , M_{P4} , M_{N3} , M_{P5} , M_{N4} , M_{N5} , M_{P6} , M_{P7} , M_{N6} , M_{P8} , M_{N7} , M_{P9} , M_{N8} , M_{N9}).

As analyzed in Section 4.3, the detailed analysis procedure for the master/slave configuration is shown in the following subsections.

5.3.1 Step 1

Classify the response message sent from the slave node into its corresponding pattern (Fig. 13 and Table 4).

The Roman numerals represent types of patterns, and the capital letters in the brackets represent the different cases in pattern I.

5.3.2 Step 2

Calculate the delay time distribution of each response message according to the corresponding pattern's method shown in Table 2. For example, the pattern of message M_{N1} is case A of pattern I; thus, the delay time distribution of M_{N1} can be calculated using the procedure discussed in Section 4.1.1. The calculation of the delay time distribution for other response messages can be conducted in a similar way.

5.3.3 Step 3

Finally, the delay time cumulative distribution of response messages can be obtained. In this subsection, we take the delay time cumulative distribution

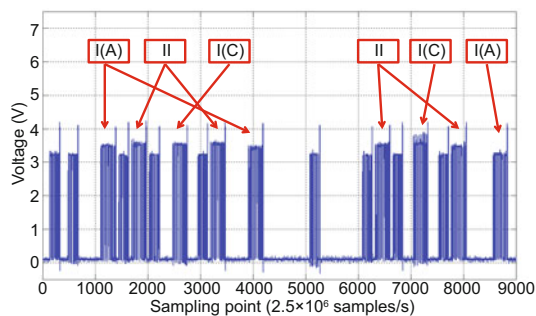


Fig. 13 Physical waveform graph of all the messages in case study 2

of three messages (i.e., M_{N1} , M_{N2} , and M_{N3}) as examples.

Figs. 14–16 show the delay time distribution comparisons between the fitted result and practical observation for response messages M_{N1} , M_{N2} , and M_{N3} , respectively.

As can be seen from Figs. 14–16, the delay time distributions calculated using the proposed method agree well with the practical observations. The calculated delay time distribution in Fig. 15 can be separated into two parts according to its trend: the first part is from 0 μ s to around 200 μ s, and the

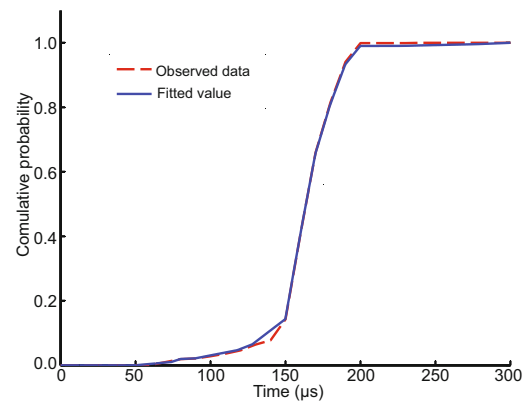


Fig. 14 Delay time comparison between the fitted value and practical observation for M_{N1} in case study 2 (root mean square error: 0.0073; maximum absolute error: 0.0308)

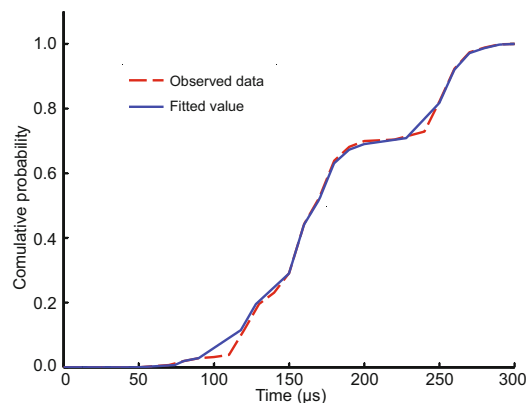


Fig. 15 Delay time comparison between the fitted value and practical observation for M_{N2} in case study 2 (root mean square error: 0.0140; maximum absolute error: 0.0423)

Table 4 Corresponding pattern of each message sent from the slave node in case study 2

Message	M_{N1}	M_{N2}	M_{N3}	M_{N4}	M_{N5}	M_{N6}	M_{N7}	M_{N8}	M_{N9}
Pattern	I(A)	II	I(C)	II	I(A)	II	I(C)	II	I(A)

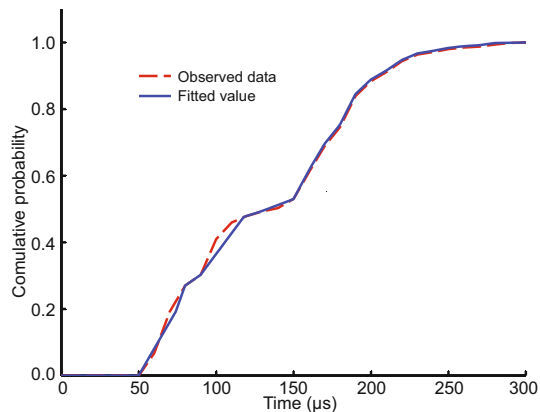


Fig. 16 Delay time comparison between the fitted value and practical observation for M_{N3} in case study 2 (root mean square error: 0.0121; maximum absolute error: 0.0345)

other part is the rest until 300 μs . These two parts are just the two scenarios decomposed in pattern II. Furthermore, the calculated delay time distribution in Fig. 16 can be separated into two parts according to its trend: the first part is from 0 μs to around 120 μs , and the other part is the rest until 300 μs . These two parts are just the two scenarios decomposed in case C of pattern I. The potential explanations for the slight gaps between the fitted result and practical observation around some time intervals in Figs. 14–16 are the unpredictability of the duration of the IC fault arrival, the variation of the sample point, and the different degrees of concentration of the dominant bits of a message.

In conclusion, the method proposed in this study is effective for calculating the delay time distributions of the response messages for both the single slave node configuration and the master/slave configuration.

6 Conclusions and future work

In this paper, a novel message delay time distribution analysis method under errors has been proposed for the CAN network. The arrivals of an IC fault were modeled by the Poisson process, and the probability factor was developed to describe the causal relationship between the IC fault arrivals and the error interruptions on the bus. Then the message delay time distribution considering errors for the single slave node configuration was analyzed in detail. After elaborately analyzing the delay time distribution for typical patterns and cases of the message

queues, the analysis framework of the message delay time distribution for the master/slave configuration considering both message blocking and error interruptions was proposed. A testbed was constructed and case studies were carried out to demonstrate and verify the delay time distribution analysis method. As shown in the case studies, the delay time distributions calculated using the proposed method agreed well with the actual observations for both single and master/slave configurations. Future work includes developing a systematic method to estimate delay distributions under complex stochastic errors and developing a model to describe the influence of the degree of concentration of the dominant bits on the probability factor.

Compliance with ethics guidelines

Lei-ming ZHANG, Yi-chao SUN, and Yong LEI declare that they have no conflict of interest.

References

- Bosch R, 1991. CAN Specification Version 2.0. Technical Report, Rober Bousch GmbH. <http://esd.cs.ucr.edu/webres/can20.pdf>
- Broster I, Burns A, Rodríguez-Navas G, 2005. Timing analysis of real-time communication under electromagnetic interference. *Real-Time Syst*, 30(1-2):55-81. <https://doi.org/10.1007/s11241-005-0504-z>
- Chen X, Liu LY, Lü WJ, et al., 2012. Modeling and analysis of response time of CAN bus based on queueing theory. *J Tianjin Univ*, 45(3):228-235 (in Chinese). <https://doi.org/10.3969/j.issn.0493-2137.2012.03.007>
- Davis RI, Navet N, 2012. Controller area network (CAN) schedulability analysis for messages with arbitrary deadlines in FIFO and work-conserving queues. *Proc 9th IEEE Int Workshop on Factory Communication Systems*, p.33-42. <https://doi.org/10.1109/WFCS.2012.6242538>
- Davis RI, Burns A, Bril RJ, et al., 2007. Controller area network (CAN) schedulability analysis: refuted, revisited and revised. *Real-Time Syst*, 35(3):239-272. <https://doi.org/10.1007/s11241-007-9012-7>
- Davis RI, Kollmann S, Pollex V, et al., 2011. Controller area network (CAN) schedulability analysis with FIFO queues. *Proc 23rd Euromicro Conf on Real-Time Systems*, p.45-56. <https://doi.org/10.1109/ECRTS.2011.13>
- Davis RI, Kollmann S, Pollex V, et al., 2013. Schedulability analysis for controller area network (CAN) with FIFO queues priority queues and gateways. *Real-Time Syst*, 49(1):73-116. <https://doi.org/10.1007/s11241-012-9167-8>
- Farsi M, Ratcliff K, Barbosa M, 1999. An overview of controller area network. *Comput Contr Eng J*, 10(3):113-120. <https://doi.org/10.1049/cce:19990304>
- Hansson HA, Nolte T, Norström C, et al., 2002. Integrating reliability and timing analysis of CAN-based systems. *IEEE Trans Ind Electron*, 49(6):1240-1250. <https://doi.org/10.1109/TIE.2002.804970>

- Kumar M, Kumar A, Srividya VA, 2009. Response-time modeling of controller area network (CAN). In: Garg V, Wattenhofer R, Kothapalli K (Eds.), Distributed Computing and Networking, Springer Berlin Heidelberg, p.163-174.
https://doi.org/10.1007/978-3-540-92295-7_20
- Lei Y, Yuan Y, Zhao JZ, 2014. Model-based detection and monitoring of the intermittent connections for CAN networks. *IEEE Trans Ind Electron*, 61(6):2912-2921.
<https://doi.org/10.1109/TIE.2013.2272277>
- Lei Y, Xie H, Yuan Y, et al., 2015. Fault location for the intermittent connection problems on CAN networks. *IEEE Trans Ind Electron*, 62(11):7203-7213.
<https://doi.org/10.1109/TIE.2015.2442518>
- Mubeen S, Mäki-Turja J, Sjödin M, 2014. Extending worst case response-time analysis for mixed messages in controller area network with priority and FIFO queues. *IEEE Access*, 2:365-380.
<https://doi.org/10.1109/ACCESS.2014.2319255>
- Mubeen S, Mäki-Turja J, Sjödin M, 2015. Integrating mixed transmission and practical limitations with the worst-case response-time analysis for controller area network. *J Syst Softw*, 99:66-84.
<https://doi.org/10.1016/j.jss.2014.09.005>
- Navet N, Song YQ, Simonot F, 2000. Worst-case deadline failure probability in real-time applications distributed over controller area network. *J Syst Archit*, 46(7):607-617. [https://doi.org/10.1016/S1383-7621\(99\)00016-8](https://doi.org/10.1016/S1383-7621(99)00016-8)
- Sun YC, Yang F, Lei Y, 2015. Message response time distribution analysis for controller area network containing errors. Chinese Automation Congress, p.1052-1057.
<https://doi.org/10.1109/CAC.2015.7382654>
- Tindell K, Burns A, 1994. Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Control Networks. Technical Report, Real-Time System Research Group, Department of Computer Science, University of York, England.
- Tindell K, Burns A, Wellings AJ, 1994. An extendible approach for analyzing fixed priority hard real-time tasks. *Real-Time Syst*, 6(2):133-151.
<https://doi.org/10.1007/BF01088593>
- Tindell K, Burns A, Wellings AJ, 1995a. Analysis of hard real-time communications. *Real-Time Syst*, 9(2):147-171. <https://doi.org/10.1007/BF01088855>
- Tindell K, Burns A, Wellings AJ, 1995b. Calculating controller area network (CAN) message response times. *Contr Eng Pract*, 3(8):1163-1169.
[https://doi.org/10.1016/0967-0661\(95\)00112-8](https://doi.org/10.1016/0967-0661(95)00112-8)
- Yomsı PM, Bertrand D, Navet N, et al., 2012. Controller area network (CAN): response time analysis with offsets. Proc 9th IEEE Int Workshop on Factory Communication Systems, p.43-52.
<https://doi.org/10.1109/WFCS.2012.6242539>
- Zeng HB, di Natale M, Giusto P, et al., 2010. Using statistical methods to compute the probability distribution of message response time in controller area network. *IEEE Trans Ind Inform*, 6(4):678-691.
<https://doi.org/10.1109/TII.2010.2050143>