# Probabilistic Response Time Bound for CAN Messages with Arbitrary Deadlines

Philip Axer, Maurice Sebastian, Rolf Ernst
TU Braunschweig
Germany
{axer, sebastian, ernst}@ida.ing.tu-bs.de

***Abstract*—The controller area network (CAN) is widely used in industrial and the automotive domain and in this context often for hard real-time applications. Formal methods guide the designer to give worst-case guarantees on timing. However, due to bit errors on the communication channel response times can be delayed due to retransmissions. Some methods exist to cover these effects, but are limited e.g. (support only periodic real-time traffic). In this paper we generalize existing methods to support arbitrary deadlines, and derive a probabilistic response time bound which is especially useful with the emergence of the new automotive safety standard ISO 26262.**

## I. Introduction

The Controller Area Network (CAN) is one of the most eminent buses used in various fields (e.g. automotive, industrial, aerospace) today. Although it has already been introduced in the 80's, it is still in use today due to its cost advantage, versatility and robustness against errors. Due to its simplistic nature, CAN is a priority driven serial bus, it is often used for real-time system, where the worst-case timing delays of transmissions must be predictable. For example, today's cars feature a rich set of distributed control algorithms which are mapped to Electronic Control Units (ECUs) connected via CAN. Formal methods known from the real-time analysis allow the prediction of such networks.

As specified by the CAN standard, an off-the-shelf controller includes an error detection mechanism based on a cyclic redundancy check and automatic retransmission of messages, thus due to its error robustness, CAN is frequently used in safety critical application such as active-steering or for controlling heavy industrial machinery.

Even though the CAN protocol will most likely detect transmission errors and schedule retransmissions until the data is transmitted correctly, the transmission latency is severely affected compared to an error-free transmission. Thus, for hard-real time systems which operate in an environment under electromagnetic interference (EMI) such as electric cars, the transmission latency which is predicted by formal real-time analysis which are based on the absence of errors does not apply anymore.

Deployment in safety-critical domains makes a strongly safety oriented product life-cycle necessary which qualifies a product for deployment in safety-critical missions. This is not only crucial in order to minimize the risk of casualties in case of system failure but also ensures product liability. To unify safety requirements, safety standards such as the industrial-oriented IEC-61508 [6] or the automotive domain specific ISO-26262 [7] specify a safety certification process. In this context it is especially important to consider CAN communications under errors.

**978-3-9810801-8-6/DATE12/©2012 EDAA**

Thus, the goal of this paper is to derive probabilistic scheduling guarantees for CAN communication under the presence of errors, which can be used in a certification process. The rest of the paper is structured as follows: First, we summarize related work in Section II, then we describe the CAN protocol and the generic response time analysis for the error-free and error case in Section III. In Section IV we compute probabilistic bounds on the response time considering the error-case. After we apply the presented method to an automotive benchmark in Section V, we conclude the work in Section VI.

## II. Related Work on CAN

Formal response time analyses are available from real-time research for a large variety of different scheduling policies, which can be directly applied to fault-free analysis. For example, when computing a task's worst-case response time, which is the time from its release until completion, on a single-resource (i.e processor or bus) under static priority preemptive scheduling, one can rely on the *busy window* technique [8], [13]. The busy window of a task is defined as the maximal time interval for which a resource executes only tasks of priority greater than or equal to the priority of the task under analysis and during which the resource is never idle [13]. The maximum response time for a CAN frame can then be derived from the busy window [4].

In order to include effects of errors (e.g. retransmission overhead) different approaches were introduced.

In [11], an approach is presented to tightly bound the reliability for periodic, synchronized messages. Therefore, a reliability metric $\mathcal{R}(t)$ is defined which denotes the probability that CAN communication survives time $t$ without a deadline miss. The reliability is calculated based on the hyperperiod, which is the time when the activation pattern of a periodic message set repeats itself. It is defined by the least common multiple over all periods. Hence, the complexity of the algorithm depends on the amount of activations in the hyperperiod. This algorithm is suitable for automotive message sets in which periods are typically multiples of 10ms. However, if messages are not synchronized, or the relative phasing is unknown the approach is not applicable.

In [3], the busy-window approach is used and a tree-based approach is presented, where different error scenarios are

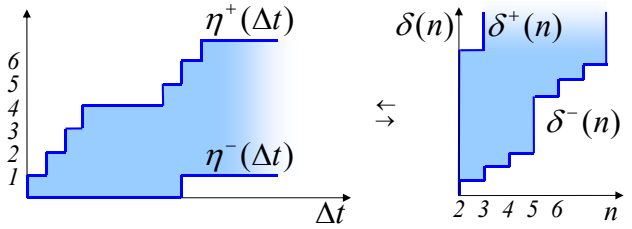


Fig. 1. Event Model Representation

evaluated iteratively. In a second step, these scenarios are translated to probabilities and a *worst case deadline failure probability* is calculated. The approach was extended in [2], and the tree-based was superseded by a simpler, more accurate approach. However, both methods [2], [3] allow only deadlines smaller than the periods, which is a limit for practical use since bursty CAN traffic is not supported.

## III. CAN Protocol and Timing Analysis

The CAN protocol is a multi-master, differential, serial bus. On the physical layer the CAN transceiver output is an open-collector or "wired and" circuit. Data is transmitted in frame entities which are non-preemptable. The arbitration scheme uses carrier sense multiple access/bitwise arbitration (CSMA/BA) and is based on the fact that dominant bits "win" the access to the physical medium. Thus, the smaller the CAN bus identifier, the higher the priority of the frame. In 1991, CAN 2.0 introduced extended-frames which effectively increase the set of possible CAN identifiers. An exact protocol description can be found in the official specification [1].

### A. CAN Error Handing

All nodes check for protocol consistency during the transmission of frames. If framing rules are violated (e.g. missing acknowledgement), or the CRC field does not match the payload an error can be signaled by all bus subscribers. The CAN standard defines special error frames for this purpose.

When an error frame is transmitted, other nodes drop the recent frame and a retransmission is triggered. The worst-case overhead for an error-frame can be given as

$$F = 31\ t_{bit} \tag{1}$$

After an error-frame has been transmitted, the re-transmission has to compete in a new arbitration phase.

The protocol guarantees [10] that the residual error probability for an undetected corrupted message is smaller than $BER \cdot 4.7 \cdot 10^{-11}$ where BER denotes the bit error rate. This is sufficiently high for safety critical applications thus we neglect undetected CRC errors in the following analysis.

### B. Event Models

Throughout the paper we use *event models* as an abstract model [9] for the activation of CAN message frames. An event model describes the maximum and minimum amount of events $\eta^+$,$\eta^-$ which arrive during a given time window $\Delta t$ at the CAN controller and are queued for transmission. Figure 1 shows $\eta^+$ and $\eta^-$ on the left. An alternative representation is the notion of a minimum and maximum time window which covers at least and at most $n$ subsequent events $\delta^-(n)$, $\delta^+(n)$. We can interpret $\delta^-(n)$, $\delta^+(n)$ as the distance from the start of the busy window until the earliest and latest arrival of the n-th event. Both representations $\eta$ and $\delta$ are pseudoinverse and can be converted to each other:

$$\delta^-(n) = \min_{\Delta t \geq 0, \Delta t \in \mathbb{R}} \left\{ \Delta t | \eta^+(\Delta t) \geq n \right\} \tag{2}$$

$$\eta^+(\Delta t) = \max_{n \in \mathbb{N}, n \geq 1} \left\{ n | \delta^-(n) \leq \Delta t \right\} \tag{3}$$

For compact representation, standard event models in [9] use three parameters, event model period $\mathcal{P}$, event model jitter $\mathcal{J}$ and the minimum distance between two events $d^{min}$. The $\eta^+$ function for a bursty input is then defined as:

$$\forall \Delta t > 0 : \eta^+(\Delta t) = \min \left( \left\lceil \frac{\Delta t}{d^{min}} \right\rceil, \left\lceil \frac{\Delta t + \mathcal{J}}{\mathcal{P}} \right\rceil \right) \tag{4}$$

The standard event models are applicable to many typical real-time setups, for instance in the automotive domain where periodic systems are predominant.

### C. Response Time Analysis in the Error-Free Case

The response time of a message is the latency from message activation until it is fully transmitted over the bus. In hard real-time systems, the response times from all activation of a task $\tau_i$ must be smaller than a given deadline $D_i$. In order to show that all observed response times are actually smaller than $D_i$, it is necessary to derive the worst-case response time (WCRT).

For the timing analysis, the concept of the *busy window* or *busy period* [8] is used. Similar to [4], we define the level-$i$ busy window as the time the bus is busy transmitting messages of priority $i$ or higher. The longest level-$i$ busy window can be constructed under the following conditions: No messages of priority $i$ or higher are queued right before the beginning of the level-$i$ busy window. The level-$i$ busy window is initiated with the *critical instant*, that is all tasks are released simultaneously and thus create the highest load possible. All following activations are then released as early as possible according to $\eta^+$.

The worst case queuing delay for message $\tau_i$ happens when the longest message of lower priority with the according transmission time $B_i$ was admitted to the bus right before the start of the critical instant, so that all higher priority activations are delayed by $B_i$ at most.

$$B_i = \max_{\forall k \in lp(\tau_i)} (C_k) \tag{5}$$

Under these assumptions, the busy window of a frame $\tau_i$ is then given by the following recursive equation.

$$w_i = B_i + \sum_{j \in hp(\tau_i) \cup \tau_i} C_j \cdot \eta^+(w_i) \tag{6}$$

Here the busy window is the sum of the blocker and all messages of higher or the same priority of $\tau_i$ which are released in the busy window $w_i$. This fixed point problem can be solved by the following recurrence relation:

$$w_i^{n+1} = B_i + \sum_{j \in hp(\tau_i) \cup \tau_i} C_j \cdot \eta^+(w_i^n) \tag{7}$$

starting with $w_i^0 = B_i$. The iteration can be stopped once the smallest fixed-point $w^{n+1} = w^n$ is found.

As shown in [4], any instance of message $\tau_i$ released in the level-i busy window can potentially lead to the worst response

time. Thus, it is necessary to check all $q_{max}$ activations in the busy window for their response times.

$$q_{max}=\eta_i^+(w_i) \tag{8}$$

The first release of a frame $\tau_i$ in the busy window corresponds to $q=1$, and the last message in the busy window is $q=q_{max}$. The finishing time of the q-th activation can be calculated similarly to the busy window by forming a recurrence relation. An activation of task $\tau_i$ can start transmission when all higher priority messages and all previous queued activations of $\tau_i$ have been transmitted. Thus, the waiting time until the q-th activation starts transmission is given by:

$$w_i(q)=B_i+(q-1)C_i+\sum_{j\in hp(\tau_i)} C_j\cdot\eta^+(w_i(q)) \tag{9}$$

Similarly to the level-i busy window, this fixed-point problem can be iteratively solved.

The finishing time is then the waiting time plus the transmission time: $w_i(q)+C_i$. The response time of the q-th event is the relative time from the release of the q-th event until it arrives at the receiver.

$$r_i(q)=w_i(q)+C_i-\delta^-(q) \tag{10}$$

Therefore, the greatest response time is the worst-case response time for the message $\tau_i$.

$$R_i=\max_{1\le q\le q_{max}} r_i(q) \tag{11}$$

### D. Response Time Analysis in the Presence of Errors

The analysis as presented does not cover the effect of transmission errors. Obviously, detected errors trigger the transmission of an error frame as well as a retransmission which increases the busy window and therefore the response time. On the other hand a longer busy window might increase the probability that successive errors might affect the busy window.

We model the occurrence of errors by using a Poisson model as used in previous work (e.g. in [2]). Practically, a Poisson process models independent single bit errors (without bursts), where $\lambda$ specifies the bit error rate. The probability for the occurrence of $m$ error-events in the time window $\Delta t$ is:

$$p(m,\Delta t)n=\frac{e^{-\lambda\Delta t}(\lambda\Delta t)^m}{m!} \tag{12}$$

As discussed before, the error penalty which affects $\tau_i$ in case of one error event is comprised of the protocol overhead of error signaling $F$ plus one retransmission of the longest frame of equal or higher priority:

$$E_i=F+\max_{\forall j\in hp(\tau_i)\cup\tau_i} C_j \tag{13}$$

Consequentially, the worst-case overhead for K errors can be bounded to:

$$E_{i|K}=K\cdot E_i \tag{14}$$

Given $K$ errors occur during the transmission of frame $\tau_i$ with a corresponding error overhead of $E_{i|K}$, the formula for the level-i busy window can easily be adapted by including the error as an additional blocker term. The k-error, level-i busy window $w_{i|K}$ is than defined as:

$$w_{i|K}=E_{i|K}+B_i+\sum_{j\in hp(\tau_i)\cup\tau_i} C_j\cdot\eta^+(w_i) \tag{15}$$

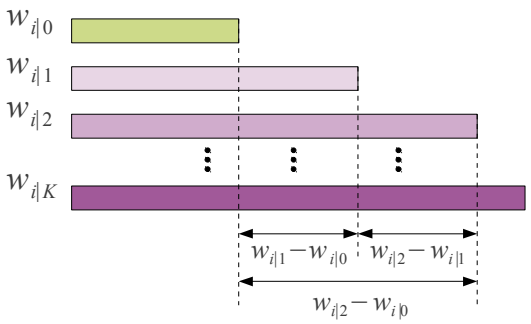


Fig. 2. Possible level-i, k-error busy windows

For each k-error scenario, it is necessary to evaluate which activation leads to the worst case response time. Similarly to the error-free case, we add the $E_{i|K}$ to equation 9 to obtain the waiting time for the q-th activation in the k-error case.

$$w_{i|K}(q)=E_{i|K}+B_i+(q-1)C_i+\sum_{j\in hp(\tau_i)} C_j\cdot\eta^+(w_{i|}(q)) \tag{16}$$

The worst response time $R_{i|K}$ in case of $K$ errors is calculated similarly to the error-free case:

$$r_{i|K}(q)=w_{i|K}(q)+C_i-\delta^-(q) \tag{17}$$

$$R_{i|K}=\max_{1\le q\le q_{max}} r_{i|K}(q) \tag{18}$$

## IV. Probabilistic Timing Analysis Revised

It is then possible to precompute all k-error, level-i busy windows until a threshold criterion is reached (e.g. until the deadline is exceeded). This stopping condition will eventually be reached in a finite amount of analysis steps, since the sequence of $w_{i|K}$ is strictly increasing function in $K$ as shown in Figure 2.

Up to this point we have calculated the response times $R_{i|K}$ and the level-i busy window for each k-error scenario $w_{i|K}$. The remaining problem is to calculate the probability that a busy window of length $w_{i|K}$ actually occurs.

Now we revise the method to derive the probabilities for the busy-window probabilities as presented in [2]. For the following argumentation it is important to note the difference between error-events (the actual bit error) and a retransmission event. It is possible that a message of length $C$ is hit by multiple error-events and only one retransmission occurs (e.g. after reception when the CRC is checked), but it is assumed, that in the worst-case condition, each error-event will lead to exactly one retransmission. Thus, we can directly use equation 12 to obtain the probability that K error-events occur during a given time window and the probability for the error-free case is:

$$P(w_{i|0})=p(0,w_{i|0})=e^{-\lambda w_{i|0}} \tag{19}$$

For $K>0$ it is not enough to just calculate $p(K,w_{i|K})$, because error-events have to occur in certain segments of the busy window. For instance, $w_{i|1}$ will only occur if exactly one retransmission occurs during the time interval $(0,w_{i|0})$ (c.f. Figure 2). Similarly, $w_{i|2}$ will only occur, in either of the following two scenarios: two retransmissions occur in the interval $(0,w_{i|0})$ or, one retransmission in $(0,w_{i|0})$ and one in $(w_{i|0},w_{i|1})$. In [2] it was shown, that the amount of these

combinations is given by the Catalan Series which grows rapidly with $K$, thus exhaustive enumeration is not possible. Also, a more efficient technique was proposed, which can be applied for the general case in which a busy-window includes multiple queued activations which can be affected by errors.

The approach works as follows: One error-event in the entire busy-window $w_{i|1}$ can happen in two ways. The error may actually lead to an $w_{i|1}$ busy window with the probability $P(w_{i|1})$. Or, we face a busy window of length $w_{i|0}$ and the error event occurs in the interval $(w_{i|0}, w_{i|1})$ These intervals are also highlighted in Figure 2.

$$p(1, w_{i|1}) = P(w_{i|1}) + P(w_{i|0})p(1, w_{i|1} - w_{i|0}) \quad (20)$$

The value of $P(w_{i|1})$ can then be obtained by rearranging the equation. Similarly we can apply this idea to $K=2$. Two errors in the time window $w_{i|2}$ may occur in the following mutually exclusive ways. i) in a way that a busy window of length $w_{i|2}$ actually occurs assuming two error-events with the probability $P(w_{i|2})$. ii) $w_{i|1}$ occurred which implies exactly one error in $w_{i|1}$ and the second error must then happen in the interval $(w_{i|1}, w_{i|2})$. iii) $w_{i|0}$ occurred which implies no error in $w_{i|0}$ and exactly two errors must be in the interval $(w_{i|0}, w_{i|2})$.

$$p(2, w_{i|2}) = P(w_{i|2}) + P(w_{i|1})p(1, w_{i|2} - w_{i|1}) + P(w_{i|0})p(2, w_{i|2} - w_{i|0}) \quad (21)$$

By rearranging the equation for $P(w_{i|2})$ we get the probability for a $K=2$ busy window. The same argument is valid for the following k-error busy windows and Equation 21 is generalized into the following form:

$$P(w_{i|K}) = p(K, w_{i|K}) - \sum_{j=0}^{K-1} P(w_{i|j})p(K-j, w_{i|K} - w_{i|j}) \quad (22)$$

The worst-case response time exceedance function can be calculated as:

$$P^+[R_i > r] = 1 - \sum_{\forall K | R_{i|K} < r} P(w_{i|K}) \quad (23)$$

Practically, this function denotes a bound for the probability that a response time exceeds a certain threshold and the probability that a deadline is exceeded can be bounded to $P^+[R_i > D_i]$.

## V. Experiments

To evaluate the presented algorithm we use a modified version of the 17-messages SAE benchmark as presented in [12]. The benchmark includes sporadic messages, as well as periodic messages. Sporadic messages are modeled by assuming a minimum interarrival time, also we assume that the CAN bus is part of a larger distributed, automotive network and the data which ought to be transmitted has been processed on different ECUs which results in an increased released jitter (e.g. due to scheduling on upstream ECUs). Thus, the used message set covers a broader spectrum and may be more applicable to today's automotive networks. This is, for some frames we increased the jitter to 1ms. Besides from that, the benchmark was used as it is. The analysis result is shown in Figure 3 as exceedance functions which are stepped due to the nature of the response time analysis. The experiment was carried out using a 125 kbit/s CAN bus and a bit error rate of $10^{-7}$, which was measured by [5] in an aggressive environment.

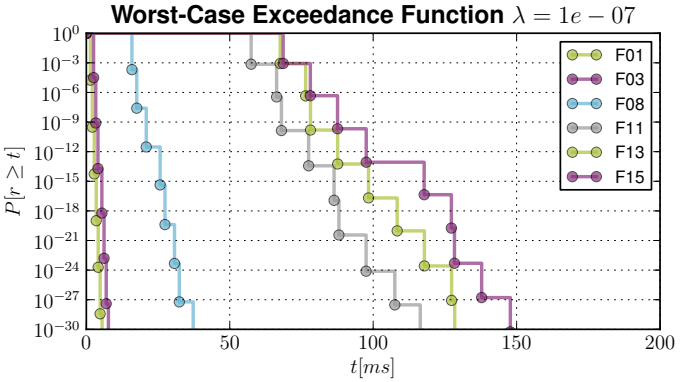


Fig. 3. Possible level-i, k-error busy windows

## VI. Conclusion

In this paper, we generalized methods to compute probabilistic bounds on hard real-time CAN messages for response times greater than deadlines. Therefore, we extended the general CAN worst-case response time analysis and included the error case. In a second step, we calculated the probability of different error cases and derived a probabilistic response time bound.

We showed the applicability by analyzing an industry (SAE) reference frameset and derived the corresponding response time exceedance functions.

## References


[1] ISO 11898-1:2003 - Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling, 2003.

[2] I. Broster and A. Burns. Comparing real-time communication under electromagnetic interference. In *Proc. of Euromicro Conference on Real-Time Systems*, pages 45–52, 2004.

[3] I. Broster, A. Burns, and G. Rodriguez-Navas. Probabilistic analysis of can with faults. In *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*, pages 269 – 278, 2002.

[4] R. Davis, A. Burns, R. Bril, and J. Lukkien. Controller area network (can) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems*, 35(3):239–272, 2007.

[5] J. Ferreira, A. Oliveira, P. Fonseca, and J. Fonseca. An experiment to assess bit error rate in can. *Proc. of RTN*, 2004.

[6] International Electrotechnical Commission (IEC). Functional safety of electrical / electronic / programmable electronic safety-related systems, 1998.

[7] International Organization for Standardization (ISO). Iso/fdis 26262: Road vehicles – functional safety, 2000.

[8] J. Lehoczky. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. *Proc. 11th Real-Time Systems Symposium*, pages 201–209, Dec 1990.

[9] K. Richter. *Compositional scheduling analysis using standard event models*. PhD thesis, TU Braunschweig, 2005.

[10] Robert Bosch GmbH, Postfach 30 02 40, D-70442 Stuttgart. CAN Specification version 2.0, 1991.

[11] M. Sebastian and R. Ernst. Reliability analysis of single bus communication with real-time requirements. In *Proc. 15th IEEE Pacific Rim Int. Symp. Dependable Computing PRDC '09*, pages 3–10, 2009.

[12] K. Tindell and A. Burns. Guaranteeing message latencies on control area network (can). In *Proceedings of the 1st International CAN Conference*. Citeseer, 1994.

[13] K. W. Tindell, A. Burns, and A. J. Wellings. An Extendible Approach for Analyzing Fixed Priority Hard Real-Time Tasks. *Real-Time Systems*, 6(2):133–151, 1994.