Integrating Reliability and Timing Analysis of CAN-Based Systems

Hans A. Hansson, *Associate Member, IEEE*, Thomas Nolte, *Student Member, IEEE*, Christer Norström, *Associate Member, IEEE*, and Sasikumar Punnekkat, *Member, IEEE*

***Abstract*—This paper presents and illustrates a reliability analysis method developed with a focus on Controller-Area-Network-based automotive systems. The method considers the effect of faults on schedulability analysis and its impact on the reliability estimation of the system, and attempts to integrate both to aid system developers. We illustrate the method by modeling a simple distributed antilock braking system, and showing that even in cases where the worst case analysis deems the system unschedulable, it may be proven to satisfy its timing requirements with a sufficiently high probability. From a reliability and cost perspective, this paper underlines the tradeoffs between timing guarantees, the level of hardware and software faults, and per-unit cost.**

***Index Terms*—Automotive system, Controller Area Network (CAN), distributed system, fieldbus, real-time analysis, reliability analysis, tradeoff analysis.**

I. Introduction

THIS PAPER extends and improves preliminary results presented in [1] and [2], by refining the fault model and providing a more rigorous treatment. During the last decade or so, real-time researchers have extended schedulability analysis to a mature technique which for nontrivial systems can be used to determine whether a set of tasks executing on a single CPU or in a distributed system will meet their deadlines or not [3]–[5]. The main focus of the real-time research community is on hard real-time systems, and the essence of analyzing such systems is to investigate if deadlines are met in a worst case scenario. Whether this worst case actually will occur during execution, or if it is likely to occur, is not normally considered.

Reliability modeling, on the other hand, involves study of fault models, characterization of distribution functions of faults and development of methods and tools for composing these distributions and models in estimating an overall reliability figure for the system.

In [6], we presented a model for calculating worst case latencies of messages under error assumptions for the Controller Area Network (CAN). In many situations, this analysis might infer that a given message set is not feasible under worst case fault interference. Such a result, though correct, is of limited help to the system designers except to prompt them to overdesign the system and waste resources to tackle a situation, which might never happen during the life time of the system.

When performing schedulability analysis (or any other type of formal analysis) it is important to keep in mind that the analysis is only valid under some specific model assumptions, typically under some assumed "normal condition," e.g., no hardware failures and a "friendly" environment. The "abnormal" situations are typically catered for in the reliability analysis, where probabilities for failing hardware and environmental interference are combined into a system reliability measure. This separation of deterministic (0/1) schedulability analysis and stochastic reliability analysis is a natural simplification of the total analysis, which unfortunately is quite pessimistic, since it assumes that the "abnormal" is equivalent to failure; in particular, for transient errors/failures, this may not at all be the case.

Consider, for instance, occasional external interference on a communication link. The interference will lead to transmission errors and subsequent retransmission of messages. The effect will be increased message latencies which may lead to missed deadlines, especially if the interference coincides with the worst case message transmission scenario considered when performing schedulability analysis. In other scenarios, the interference will not increase the worst case message latency, as illustrated in Fig. 1. The figure shows a system with three periodic messages, M_1, M_2, and M_3, with the parameters shown in Table I. Assuming an overhead, $O = 1$ for error signaling and recovery (but not including retransmission of the corrupted message), we have shown the effects of three different scenarios, corresponding to an external interference hitting the system at different points in time. In the first case, the error caused by the interference results in a retransmission of M_1, causing M_2 and M_3 to miss their deadlines. In the second case, though a retransmission is necessitated, still the message set meets its deadlines, whereas in the third scenario, the error has no effect at all since it falls in a period of inactivity of bus.

This simple example shows that there are situations (scenarios) when system requirements (e.g., deadlines) are not violated by the "abnormal." Hence, there is a potential for obtaining a more accurate and tight reliability analysis by considering the likelihood of the "abnormal" actually causing a deadline vio-

Manuscript received April 4, 2001; revised September 24, 2001 and October 30, 2001. Abstract published on the Internet September 13, 2002. This work was supported by the Swedish Foundation for Strategic Research via the program ARTES, by the Knowledge and Competence Foundation via the profile grant MRTC, and by Mälardalen University.

H. A. Hansson and T. Nolte are with the Mälardalen Real-Time Research Centre, Department of Computer Engineering, Mälardalen University, SE-721 23 Västerås, Sweden.

C. Norström is with ABB Technology Partners/Robotics, SE-721 67 Västerås, Sweden, and also with the Mälardalen Real-Time Research Centre, Department of Computer Engineering, Mälardalen University, SE-721 23 Västerås, Sweden.

S. Punnekkat is with the Vikram Sarabhai Space Centre, Trivandrum 695010, India, on leave from Mälardalen University, SE-721 23 Västerås, Sweden.

Digital Object Identifier 10.1109/TIE.2002.804970

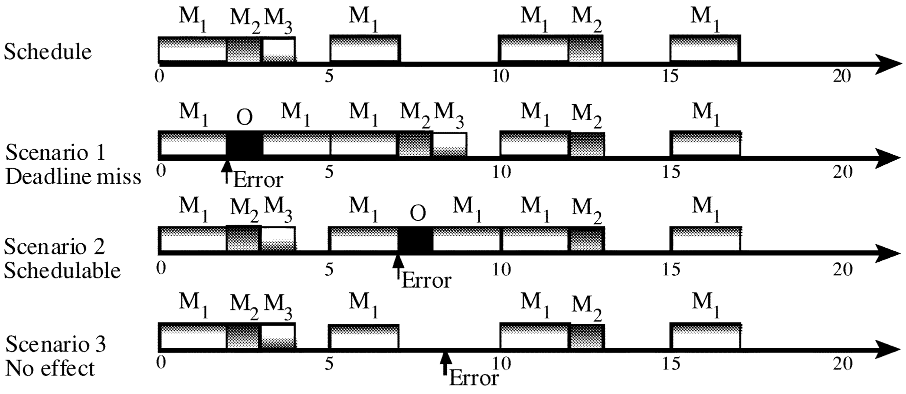

Fig. 1. Dependency of effects of faults on phasings.

TABLE I
MESSAGE SET PARAMETERS

Msg ID	Period	Deadline	Transmission time	Priority
M_1	5	5	2	High
M_2	10	7	1	Medium
M_3	20	8	1	Low

lation. The basic argument of our work, is that for any system (even the most safety critical one) the analysis is only valid as long as the underlying assumptions hold. A system can only be guaranteed up to some level, after which we must resort to reliability analysis. The main contribution of this paper is in providing a methodology to calculate such an estimate by way of integrating schedulability and reliability analysis.

Although we use automotive examples throughout the paper, it should be pointed out that our results have substantially wider applicability, both in the sense that CAN is used in other application domains, such as factory automation, and that the general approach is applicable also to other communication systems. Furthermore, our type of reasoning is especially pertinent considering the growing trend of using wireless networks in factory automation and elsewhere. The error behavior of such systems will most likely require reliability to be incorporated into the analysis of timing guarantees.

The outline of the paper is as follows. Section II presents general reliability modeling for distributed real-time systems and introduces our approach. Section III specifically discusses the scheduling of message sets in CANs under a general fault model, presents schedulability analysis for the model, and introduces a simulation-based approach for analysis of arbitrary samples of phasings and interference. Section IV illustrates our results, with a sample message set used in a distributed computer network inside passenger cars. Section V discusses possible extensions and presents some conclusions.

II. RELIABILITY MODELING

Reliability is defined as the probability that a system can perform its intended function, under given conditions, for a given time interval. In the context of an automobile its intended function is to provide reliable and cost-effective transport of men and material. At a subsystem level, such as for an Antilock Braking System (ABS) for automobiles, this boils down to performing the tasks (mainly input_sensors, compute_control, and output_actuators, etc.,) as per the specifications. Being part of a real-time system, the specifications for ABS imply the necessity for the results to be both functionally correct and within timing specifications.

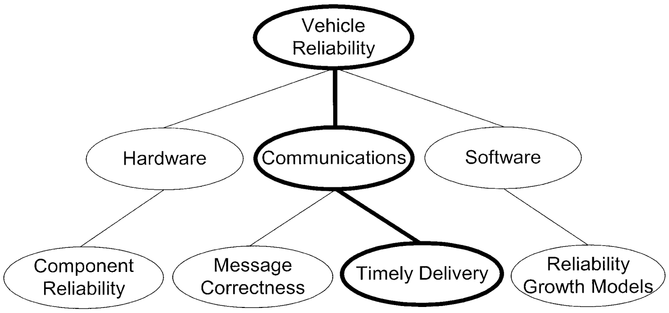

Fig. 2. System reliability: a top-down view.

A major issue here is how to compose hardware reliability, software reliability, environment model, and timing correctness to arrive at reasonable estimates of overall system reliability (Fig. 2).

Let us define

$$\begin{aligned} p_{HF}(t) &= \text{Probability (Hardware failure at } t) \\ p_{SF}(t) &= \text{Probability (Software failure at } t) \\ p_{F}(t) &= \text{Probability (Communication failure at } t). \end{aligned}$$

The reliability of the system $R(t)$ is the probability that the system performs all its intended functions correctly for a period t. This is given by the product of cumulative probabilities that there are no failures in hardware, software, and communication subsystem during the period $(0, t)$. That is,

$$R(t) = \left(1 - \int_0^t p_{HF}(t)\right) \times \left(1 - \int_0^t p_{SF}(t)\right) \times \left(1 - \int_0^t p_F(t)\right). \quad (1)$$

In this paper, we concentrate only on the final term in (1), i.e., the probability that no errors occur in the communication subsystem. Please note that, when we talk about communication subsystem, we are not concentrating on the faults in the hardware (as in [7]) or in the software of such a system. Instead, we look at it from a system level and treat its correctness as the probability of correct and timely delivery of message sets. Since the main cause for an incorrect (corrupted, missing, or delayed) message delivery is environmental interference, an appropriate modeling of such factors in the context of the environment in which the system will operate is essential for performing reliability analysis. A completely accurate modeling of the probability of timely delivery of messages is far from being trivial and hence we have made certain simplifying assumptions in order to divide the problem into manageable proportions.

A. Problem Statement

The premises of our problem (from a designer's perspective) are as follows.

- We are given the overall reliability requirements of the system (for example, a vehicle), from which we derive the reliability requirement of a particular subsystem (for example, the computer system controlling pedal brake and ABS), which in turn defines a requirement on the reliability of its communication subsystem.
- The given requirements of the subsystem, after a series of design steps, get converted to a set of relevant tasks along with their timing properties. In our example of cars, the overall vehicle level safety and performance requirements, in conjunction with the vehicle dynamics and properties of subsystems allow us to have separation of concerns between hydraulic/mechanical systems and the electrical system. In the next step, we convert these requirements and constraints into the timing specifications for the individual tasks and messages.
- We have an environment model and know how to perform response time analysis of CAN messages under normal and error conditions [6].
- The problem analyzed is, given the above information, how to find a suitable way of predicting the reliability of the communication subsystem. The simple approach will be to give a 0/1 weight to the schedulability aspect in evaluating the system "correctness" and calculate the reliability. However, the environment model provides worst case scenarios, which may not occur in practice and its impact may depend on the actual phasing of messages and the way in which they interact with the environment/fault model. Therefore, the major issues are as follows.
- Can we partition the schedulability analysis under faults by considering a set of scenarios corresponding to different message and fault model phasings? (as illustrated in Fig. 1, and where the worst case considered in our previous analysis [6] is only one of several scenarios.)
- How can we use such an analysis to obtain a more accurate reliability estimate?

B. Reliability Estimation

By definition, reliability is specified over a mission time. Normally, we can assume a repetitive pattern of messages (over the least common multiple (LCM) of the individual message periods). Each LCM is typically a very small fraction of the mission time.

We will now outline a methodology for estimating communication failures due to external interference in a CAN bus. It should be noted that, the methodology presented in this section can easily be applied to other communication models as well.

Let t represent an arbitrary time point when the external interference hits the bus and causes an error. If we can assume zero error latency and instantaneous error detection then t becomes the time point of detection of an error in the bus due to external interference.

We now define the following probabilities:

$$p_I(t) = \text{Probability (Interference at } t)$$
$$p_C(t) = \text{Probability (Msg corruption} | \text{Interference at } t)$$
$$p_M(t) = \text{Probability (Deadline miss} | \text{Interference at } t).$$

By relying on the extensive error detection and handling features available in the CAN, we can safely assume that an error in message corruption is either detected and corrected by retransmission or will ultimately result in a timing error. This allows us to ignore $p_C(t)$ and define in our context, the probability of communication failure due to an interference starting at t as

$$p_F(t) = p_M(t) \times p_I(t). \qquad (2)$$

In the environment model in [6], we have assumed the possibility of an interference I_1, having a certain pattern hitting the message transmission. Let $p_{I_1}(t)$ be the probability of such an event occurring at time t. We also assume that another interference I_2, having a different pattern, can hit the system at time t with a probability, for example, $p_{I_2}(t)$. In [6], we assumed that both cases of interference hit the message transmission in a worst case manner and looked at their impact on the schedulability. In this article, we will increase the realism by relaxing the requirement on the worst case phasing between schedule and interference. It should be noted that there is an implicit assumption that these cases of interference are independent.

C. Failure Semantics

In the above presentation, we assume that a single deadline miss causes a failure. This may be true for many systems, but in general, the *failure semantics* does not have to be restricted to this simple case. For instance, most control engineers would require the system to be more robust, i.e., the system should not loose stability if single deadlines, or even multiple deadlines, are missed. A tolerable failure semantics for such a system [8] could for instance be "three consecutive deadlines missed or five out of 50 deadlines missed." Such a definition of failure is more realistic and also leads to a substantial increase in reliability estimates, as compared to the single deadline miss case. To simplify the presentation we will, however, stick to the simple "single missed deadline" failure semantics for the time being, but return to this issue in the reliability analysis in Section IV-C.

D. Calculating Failure Probabilities

To calculate the subsystem reliability, first we need to calculate the failure probability (in our case of the communication

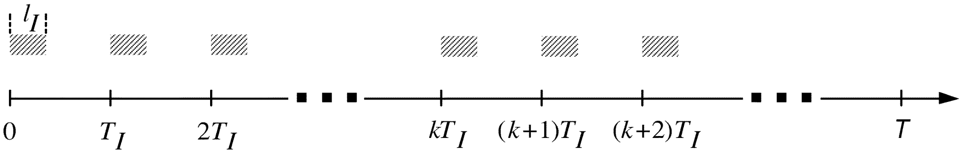

Fig. 3. Pattern of interference from an intermittent source, with burst length l_I and period T_I.

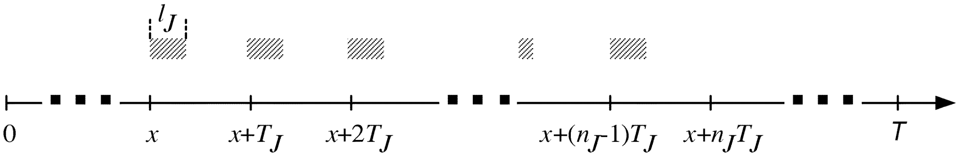

Fig. 4. Pattern of interference from a transient source, hitting the system at time x, with burst length l_J, period T_J, and consisting of n_J bursts.

subsystem subject to possible external interference), i.e., the probability of at least one failure (defined as a missed deadline) during the mission time. In doing this we assume that the interference free system is schedulable, i.e., that it meets all deadlines with probability 1. This can for a CAN bus be verified by using the analysis presented in Section III-A. Furthermore, due to the bit-wise behavior of the CAN bus, we can with respect to the external interference make a discretization of the time scale, with the time unit corresponding to a single bit time τ_{bit} (1 μs for a 1-Mb/s bus), i.e., we make no distinction between an interference hitting the entire bit or only a fraction of it. In either case, the corruption will both occur and be detected.

We will distinguish between two types of interference sources.

- *Intermittent sources,* which are bursty sources that interfere during the entire mission time $\mathcal{T}$, and are for an interference source I characterized by a period T_I and a burst length l_I (where $l_I < T_I$), as illustrated in Fig. 3.
- *Transient sources,* which are bursty sources of limited duration. These occur at most once during a mission time $\mathcal{T}$, and are for an interference source J characterized by a period T_J, a burst length l_J, and a number of bursts n_J (where $l_J < T_J$ and $T_J * n_J < \mathcal{T}$), as illustrated in Fig. 4.

For a single intermittent source I we define the probability of a communication failure during the mission time as follows:

$$P_F^{\mathcal{T}}(I) = \sum_{t \in [0,\, T_I - 1]} p_I(t) \times P_F^{\mathcal{T}}(I | h(I) = t) \tag{3}$$

where $P_F^{\mathcal{T}}(I | h(I) = t)$ denotes the conditional probability of a communication failure, given that the system was hit by interference from source I at time t, denoted by $h(I) = t$. It follows, since interference and bus communication are independent, that $p_I(t) = 1/T_I$. To calculate $P_F^{\mathcal{T}}(I | h(I) = t)$ we will use simulation, i.e., we will simulate the bus traffic during a mission, including the effects of interference, to determine if any communication failure (deadline violation) occurs. This will for each t result in either 0, if no deadline is missed, or 1, if a deadline miss is detected.

For a single transient source J we define the probability of a communication failure during the mission time as follows:

$$P_F^{\mathcal{T}}(J) = \sum_{t \in [-n_J T_J,\, \mathcal{T} - 1]} p_J(t) \times P_F^{\mathcal{T}}(J | s(J) = t) \tag{4}$$

where $p_J(t)$ denotes the probability that the first transient interference hits the system at time t, and $P_F^{\mathcal{T}}(J | s(J) = t)$ denotes the conditional probability of a communication failure during $\mathcal{T}$ due to interference from J, given that the interference J starts at time t. In the above summation all possible full or partial interference from the transient source during mission time are considered. The interval $[-n_J T_J,\, 0]$ specifically captures initial partial interference, starting before mission time, but ending during mission. It follows, since interference and bus communication are independent, that $p_J(t) = 1/(\mathcal{T} + n_J T_J)$. To calculate $P_F^{\mathcal{T}}(J | s(J) = t)$ we will use simulation, just as above. The number of scenarios to simulate here is potentially much larger than for an intermittent source, since typically $\mathcal{T} \gg T_I$. However, the number of simulations can be reduced, since the probability of failure is independent of the LCM, in which the interference hits the system. We can prove that

$$\begin{aligned} P_F^{\mathcal{T}}(J) &\leq \frac{\mathcal{T} + n_J T_J}{LCM} \times \sum_{t \in [0,\, LCM-1]} p_J(t) \times P_F^{\mathcal{T}}(J | s(J) = t) \\ &= \frac{\sum_{t \in [0,\, LCM-1]} P_F^{\mathcal{T}}(J | s(J) = t)}{LCM}. \end{aligned} \tag{5}$$

It should be noted that we introduce a slight pessimism (which is the reason for the $\leq$) since the probability for failure in $\mathcal{T}$ is lower toward the end, when the interference bursts are extending past the end of $\mathcal{T}$. However, since we assume $n_J \times T_J \ll \mathcal{T}$, the introduced pessimism can be considered negligible.

Also, note that in (5) the effects of the interference starting before the LCM is covered by the interference on the subsequent LCM, which is the reason for starting the summation with $t = 0$ [rather than $t = -n_j T_j$ as in (4)].

Finally, note that by not counting the interference starting outside the very first LCM (at the beginning of the mission time), we introduce some optimism, but since the assumption of $n_J \times T_J \ll \mathcal{T}$ this optimism is insignificant. To be more precise, a fraction of (4), $\sum_{t \in [-n_J T_J,\, 0]} p_J(t) \times P_F^{\mathcal{T}}(J | s(J) = t)$, could be added to (5) in order to cover for the pre-mission time-triggered transient faults.

We now extend the above basic analysis to the analysis of multiple sources of interference. First, we consider the case of two independent sources of interference. There are three cases to consider.

1) Two intermittent sources I_1 and I_2

$$P_F^{\mathcal{T}}(I_1, I_2) = \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ t_2 \in [0, T_{I_2}-1]}} p_{I_1}(t_1) \times p_{I_2}(t_2) \times P_F^{\mathcal{T}}(I_1, I_2 | h(I_1) = t_1 \wedge h(I_2) = t_2). \quad (6)$$

2) Two transient sources J_1 and J_2

$$P_F^{\mathcal{T}}(J_1, J_2) = \sum_{\substack{t_1 \in [-n_{J_1} T_{J_1}, \mathcal{T}-1] \\ t_2 \in [-n_{J_2} T_{J_2}, \mathcal{T}-1]}} p_{J_1}(t_1) \times p_{J_2}(t_2) \times P_F^{\mathcal{T}}(J_1, J_2 | s(J_1) = t_1 \wedge s(J_2) = t_2). \quad (7)$$

3) One intermittent and one transient source

$$P_F^{\mathcal{T}}(I_1, J_2) = \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ t_2 \in [-n_{J_2} T_{J_2}, \mathcal{T}-1]}} p_{I_1}(t_1) \times p_{J_2}(t_2) \times P_F^{\mathcal{T}}(I_1, J_2 | h(I_1) = t_1 \wedge s(J_2) = t_2). \quad (8)$$

The number of scenarios to simulate for the above three cases are in the order of $T_{I_1} \times T_{I_2}$, $\mathcal{T}^2$, and $T_{I_1} \times \mathcal{T}$, respectively. This may be rather large, especially for the two latter cases. By observing symmetries in the formulas we can however reduce the number of scenarios. For case 3), consider the following two mutually exclusive situations: 1) $LCM \geq T_{I_1}$, which leads to the following reduced formula:

$$P_F^{\mathcal{T}}(I_1, J_2) = \frac{\mathcal{T} + n_{J_2} T_{J_2}}{LCM} \times \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ t_2 \in [0, LCM-1]}} p_{I_1}(t_1) \times p_{J_2}(t_2) \times P_F^{\mathcal{T}}(I_1, J_2 | h(I_1) = t_1 \wedge s(J_2) = t_2). \quad (9)$$

and 2) $LCM < T_{I_1}$, which leads to the following reduced formula:

$$P_F^{\mathcal{T}}(I_1, J_2) = \frac{\mathcal{T} + n_{J_2} T_{J_2}}{T_{I_1}} \times \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ t_2 \in [0, T_{I_1}-1]}} p_{I_1}(t_1) \times p_{J_2}(t_2) \times P_F^{\mathcal{T}}(I_1, J_2 | h(I_1) = t_1 \wedge s(J_2) = t_2). \quad (10)$$

The above two equations can be combined into

$$P_F^{\mathcal{T}}(I_1, J_2) = \frac{\mathcal{T} + n_{J_2} T_{J_2}}{\max(LCM, T_{I_1})} \times \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ t_2 \in [0, \max(LCM, T_{I_1})-1]}} p_{I_1}(t_1) \times p_{J_2}(t_2) \times P_F^{\mathcal{T}}(I_1, J_2 | h(I_1) = t_1 \wedge s(J_2) = t_2) \quad (11)$$

and, thus, we have reduced the number of scenarios from the order of $T_{I_1} \times \mathcal{T}$ to the order of $T_{I_1} \times \max(LCM, T_{I_1})$.

Finally, we present the general formula for an arbitrary number of interference sources of either type (n intermittent and m transient sources of interference)

$$P_F^{\mathcal{T}}(I_1, \ldots, I_n, J_1, \ldots, J_m) = \sum_{\substack{t_1 \in [0, T_{I_1}-1] \\ \cdots \\ t_n \in [0, T_{I_n}-1] \\ t'_1 \in [-n_{J_1} T_{J_1}, \mathcal{T}-1] \\ \cdots \\ t'_m \in [-n_{J_m} T_{J_m}, \mathcal{T}-1]}} \begin{pmatrix} p_{I_1}(t_1) \\ \times \cdots \\ \times p_{I_n}(t_n) \\ \times p_{J_1}(t'_1) \\ \times \cdots \\ \times p_{J_m}(t'_m) \\ \times P_F^{\mathcal{T}} \begin{pmatrix} I_1, \ldots, I_n, \\ J_1, \ldots, J_m \\ | h(I_1) = t_1 \\ \wedge \cdots \\ \wedge h(I_n) = t_n \\ \wedge s(J_1) = t'_1 \\ \wedge \cdots \\ \wedge s(J_m) = t'_m \end{pmatrix} \end{pmatrix}. \quad (12)$$

E. Approach to Analysis

The above equations define the probability of communication failure given a set of sources interfering with the communication according to specified patterns. In our modeling, we will additionally specify the probability that an interference source actually is active during the mission. This gives us the following.

- We have a set of external sources of interference $\mathcal{K} = \mathcal{I} \cup \mathcal{J}$, where $\mathcal{I}$ is a set of intermittent sources of interference and $\mathcal{J}$ is a set of transient sources, as defined above.
- Each interference source $k \in \mathcal{K}$ is either active or inactive during a mission. The probability for the source to be active is P_k^{act}, and the probability for inactivity is consequently $1 - P_k^{\text{act}}$.

For a scenario with a single intermittent interference source I_1 and a single transient interference source J_2, we can now (with a slight generalization of the notation) define the probability of a communication failure in a mission as follows:

$$\begin{aligned} Q_F^{\mathcal{T}}(\{I_1, J_2\}) = {} & P_{I_1}^{\text{act}} \times P_{J_2}^{\text{act}} \times P_F^{\mathcal{T}}(I_1, J_2) \\ & + P_{I_1}^{\text{act}} \times (1 - P_{J_2}^{\text{act}}) \times P_F^{\mathcal{T}}(I_1) \\ & + (1 - P_{I_1}^{\text{act}}) \times P_{J_2}^{\text{act}} \times P_F^{\mathcal{T}}(J_2) \quad (13) \end{aligned}$$

which for an arbitrary set $\mathcal{I}$ of intermittent sources and an arbitrary set $\mathcal{J}$ of transient interference sources can be generalized to

$$Q_F^{\mathcal{T}}(\mathcal{I} \cup \mathcal{J}) = \sum_{A \subseteq \mathcal{I} \cup \mathcal{J}} \left(\prod_{a \in A} P_a^{\text{act}} \right) \times \left(\prod_{b \in \overline{A}} (1 - P_b^{\text{act}}) \right) \times P_F^{\mathcal{T}}(A) \quad (14)$$

where $\overline{A} = (\mathcal{I} \cup \mathcal{J}) \setminus A$. Intuitively, (14) defines the failure probability for a system that is potentially subjected to interference from a set $\mathcal{I} \cup \mathcal{J}$ of interference sources as the sum of weighted probabilities that a failure occurs in each of the possible combination of sources.

The value of $P_F^{\mathcal{T}}(A)$ is, as mentioned above, obtained by simulation. This amounts to the following.

- For each intermittent interference source $I \in A$ make a selection from the set of discrete time points $[0, T_I - 1]$. Each picked sample s indicates the phasing of the corresponding source at time 0. The selected phasing will give a scenario with interference from source I starting at s, $s + T_I$, $s + 2T_I, \ldots, s + nT_I$, where $n = \lceil \mathcal{T}/T_I \rceil$.
- For each transient interference source $J \in A$ we make a selection from the set of discrete time points $[-n_J T_J, \mathcal{T} - 1]$. Each picked sample indicates the time when the interference from the corresponding source hits the system.
- Perform the simulation (as detailed in Section III) to obtain $P_F^{\mathcal{T}}$ ($A|$selected phasings of interference). The result will be either 1 or 0, corresponding to no communication failures or communication failure detected, respectively.

We can then calculate $Q_F^{\mathcal{T}}(A)$ using (14).

For most realistic systems with multiple sources of interference, complete analysis involving simulation of all combinations of phasings and interference sources will not be computationally feasible. An alternative approach can then be to use a statistical-sampling-based method in which only a restricted set of phasings is simulated. That approach is further elaborated upon in Section III-C.

III. Schedulability Analysis of CAN Messages

The CAN is a broadcast bus designed to operate at speeds of up to 1 Mb/s. Data are transmitted in messages containing between 0–8 B of data. An 11-bit identifier is associated with each message. The identifier is required to be unique, in the sense that two simultaneously active messages originating from different sources must have distinct identifiers. The identifier serves two purposes: 1) assigning a priority to the message, and 2) enabling receivers to filter messages.

The CAN is a collision-detect broadcast bus, which uses deterministic collision resolution to control access to the bus. The basis for the access mechanism is the electrical characteristics of a CAN bus: if multiple stations are transmitting concurrently and one station transmits a "0," then all stations monitoring the bus will see a "0." Conversely, only if all stations transmit a "1" will all processors monitoring the bus see a "1." During arbitration, competing stations are simultaneously putting their identifiers, one bit at the time, on the bus. By monitoring the resulting bus value, a station detects if there is a competing higher priority message and stops transmission if this is the case. Because identifiers are unique within the system, a station transmitting the last bit of the identifier without detecting a higher priority message must be transmitting the highest priority queued message and, hence, can start transmitting the body of the message.

A. *Classical CAN Bus Analysis*

Tindell *et al.* [9], [10] present an analysis to calculate the worst case latencies of CAN messages. This analysis is based on the standard fixed-priority response time analysis for CPU scheduling [3].

Calculating the response times requires a bounded worst case queuing pattern of messages. The standard way of expressing this is to assume a set of traffic streams $\mathcal{S}$ (corresponding to CPU tasks), each generating messages with a fixed priority. The worst case behavior of each stream is to periodically queue messages. Each $S_i \in \mathcal{S}$ is a triple $\langle P_i, T_i, C_i \rangle$, where P_i is the priority (defined by the message identifier), T_i is the period and C_i the worst case transmission time of messages sent on stream S_i. The worst case latency R_i of a CAN message sent on stream S_i is defined by

$$R_i = J_i + q_i + C_i \tag{15}$$

where J_i is the queuing jitter of the message, i.e., the maximum variation in queuing time relative T_i, inherited from the sender task which queues the message, and q_i represents the effective queuing time, given by

$$q_i = B_i + \sum_{j \in HP(i)} \left\lceil \frac{q_i + J_j + \tau_{\text{bit}}}{T_j} \right\rceil C_j + E(q_i + C_i) \tag{16}$$

where the term B_i is the worst case blocking time of messages sent on S_i, $HP(i)$ is the set of streams with priority higher than S_i, τ_{bit} (the bit time) caters for the difference in arbitration start times at the different nodes due to propagation delays and protocol tolerances, and $E(q_i + C_i)$ is an error term denoting the time required for error signaling and recovery. The reason for the blocking factor is that transmissions are nonpreemptive, i.e., after a bus arbitration has started the message with the highest priority among competing messages will be transmitted until completion, even if a message with higher priority gets queued before the transmission is completed. However, in case of errors, a message can be interrupted/preempted during transmission, requiring a complete retransmission of the entire message. The extra cost for this is catered for in the error term E above.

B. *Our Previous Generalization*

In [6], we presented a generalization of the relatively simplistic error model by Tindell and Burns [9]. Our error model specifically addresses the following.

- *Multiple sources of errors:* Handling of each source separately is not sufficient; instead, they have to be composed into a worst case interference with respect to the latency on the bus.
- *Signaling pattern of individual sources:* Each source can typically be characterized by a pattern of shorter or longer bursts, during which the bus is unavailable, i.e., no signaling will be possible on the bus.

The above model is, just as Tindell and Burns' model, deterministic in that it models specific fixed patterns of interference. An alternative is to use a stochastic model with interference distributions. Such a model is proposed by Navet *et al.* [11], who use a Generalized Poisson Process to model the frequency of interference, as well as their duration (single errors and error bursts).

In this paper, we will use the following deterministic error model, which is a simplified version of the model introduced in [6].

- There is a set $\mathcal{K}$ of sources of interference, with each source $k_i \in \mathcal{K}$ contributing an error term $E_{k_i}(t)$. Their combined effect $E(t)$ can be defined as,

$$E(t) = E_{k_1}(t)|E_{k_2}(t)|\cdots|E_{k_{|\mathcal{K}|}}(t) \tag{17}$$

  where | denotes composition of error terms.
- Each source $k_i \in \mathcal{K}$ interferes by inducing an undefined bus value during a characteristic time period T_{k_i}. Each such interference will (if it coincides with a transmission) lead to a transmission error. If T_{k_i} is larger than τ_{bit}, then the error recovery will be delayed accordingly.
- Patterns of interference for each source $k_i \in \mathcal{K}$ can independently be specified as a sequence of n_{k_i} bursts of length l_{k_i} with period T_{k_i}.

From the generic parameters it is possible to model both intermittent and transient sources of interference, as introduced in Section II-D. An intermittent source k is defined by letting $n_k > \mathcal{T}/T_k$. Any interference source with smaller n_k is a transient source. See Figs. 3 and 4 for illustrations.

Using such a model we can see that

$$E_{k_i}(t) = B_{k_i}(t) \times (O + \max(0,\, l_{k_i} - \tau_{\text{bit}})) \tag{18}$$

where the number of interference until t, $B_{k_i}(t)$, is given by

$$B_{k_i}(t) = \min\left(n_{k_i},\, \left\lceil \frac{t}{T_{k_i}} \right\rceil\right). \tag{19}$$

Note that, $\max(0,\, l_{k_i} - \tau_{\text{bit}})$ defines the length of l_{k_i} exceeding τ_{bit}, whereas $\lceil t/T_{k_i} \rceil$ is the number of initiated bursts until t.

We assume that the overhead O_i is given by

$$O_i = 31 \times \tau_{\text{bit}} + \max_{l \in HP(i) \cup \{i\}} (C_l) \tag{20}$$

where $31 \times \tau_{\text{bit}}$ is the time required for error signaling in CAN and the $\max$ term denotes the worst case retransmission time as the largest transmission time of any message of higher priority ($HP(i)$) or the considered message (i). Retransmissions of corrupted messages with lower priority will not interfere, since the considered message will, due to the priority-based arbitration, be transmitted before any such message.

C. Analysis With Random Phasings of Interference

The analysis in Section III-B above assumes worst case phasings of queuings and interference. In combining timing and reliability modeling, we will use a relaxed model which considers the probability of different interference scenarios, not only the extreme worst case. Our relaxed model will be based on the following.

1) *Worst-case phasings of message queuings at time 0 in the LCM (actually this could be at any time, so why not choose 0?):* This introduces some pessimism, since the worst case may not occur in every LCM, but is consistent with the assumed traffic in the interference-free model.

2) *Random phasings of interference:* This can be expressed as an offset from the beginning of the mission time to when the first interference hits. For each interference source I_i that hits, such an offset should be "sampled" (as outlined in Section II-E). This can be expressed as an offset from the beginning of the mission time to when the first interference hits. For each interference source I_i that hits, such an offset should be "sampled" (as outlined in Section II-E).

To calculate the probability of a deadline miss, we perform a simulation of the message transfer and interference during the mission time $\mathcal{T}$, as described in Section II-E.

The analysis we make is based on either exhaustive simulation or by sampling. If the LCM is small and the number of combined interference sources to be analyzed are few, then exhaustive simulation is recommended. Algorithms for both exhaustive and sampling-based simulation are presented in the Appendix.

In calculating the final failure probability $Q_F^{\mathcal{T}}$ (14) we can use either of the algorithms in the Appendix which can be used to calculate $P_F^{\mathcal{T}}$. In general, the total number of required simulations in the exhaustive case is rather large, since there are $2^{|\mathcal{K}|} - 1$ combinations of interference sources to consider, and for each combination A, there are

$$\prod_{a \in A} \begin{cases} T_a, & \text{if } a \in \mathcal{I} \\ \mathcal{T}, & \text{if } a \in \mathcal{J} \end{cases}$$

phasings to consider. The actual situation is however not as bad as this may indicate, since the algorithms can be optimized for special cases (e.g., using (11) instead of (12)) and using $\mathit{Rnd\_sim}$ rather than $\mathit{Ex\_sim}$. Due to the regular pattern of many scenarios and that we immediately can conclude "communication failure" if a single failure is detected, the time to perform simulations can be substantially reduced as well. The details of this are, however, outside the scope of this paper.

We have implemented the analysis using a simulator developed by Lindgren *et al.* [12].

IV. Example: A Distributed Braking System

We now present a case study of a simplified intelligent ABS, where each separate brake is controlled by a computer. Furthermore, there is one computer that controls the brake pedal. All nodes are connected by a CAN bus (see Fig. 5). The application is a distributed control algorithm, which calculates the brake force for each wheel depending on the brake pressure achieved from the driver. Therefore, each wheel computer has to receive information about the state of the other wheels, to be able to make correct calculation and actuation. Thus each wheel is equipped with a sensor that monitors the rotation of the wheel. Each node sends the monitored values periodically.

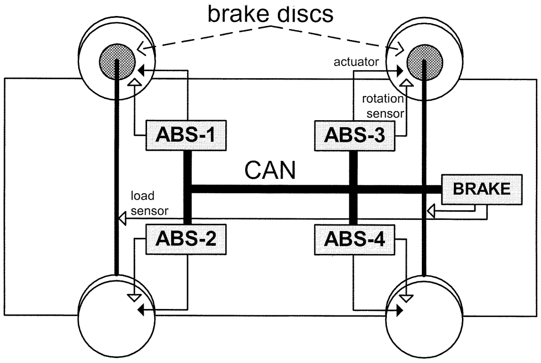

Fig. 5. Typical computer network in a car with ABS.

TABLE II
TYPICAL MESSAGE SET IN CAR

Msg ID	Priority	T_i	D_i	C_i	Sender
OPERATOR-1	1	8	8	0.54	BRAKE
ABS-1	2	4	4	0.54	ABS-1
ABS-2	3	4	4	0.54	ABS-2
ABS-3	4	4	4	0.54	ABS-3
ABS-4	5	4	4	0.54	ABS-4
OPERATOR-2	6	15	15	0.54	BRAKE

Our task is to implement a “reliable” ABS for vehicles. Since this is a subsystem of the entire vehicle system, we assume an appropriate reliability figure (for example, less than 10^{-9} faults/hour) to be attained by the ABS. This figure is, in fact, mandated by an assumed overall system reliability requirement of less than 10^{-7} faults/hour.

Table II specifies a typical subset of messages sent in this simplified ABS and the timing details (in milliseconds) of these messages sent via a CAN in a car. The timing parameters are typically requirements derived from the vehicle dynamics by a control engineer. Priority 1 is assumed to be the highest and 6 is the lowest. We assume a maximum blocking time of $135\tau_{\rm bit}$ due to background message traffic. We also assume that the CAN bus operates at 250 kb/s.

Please note that the task set above is a quite simplified one from those used in practice. Our aim here is, however, not to present an accurate model of an ABS system, but to illustrate our methodology and indicate its usefulness.

A. Interference Characteristics

In our example, we will consider two sources of interference, *viz.*, a mobile phone lying inside the vehicle and radar transmissions from ships while the vehicle crosses bridges. These are considered to represent typical sources of interference. We characterize them in the following manner.

Mobile phones typically operate at 900–1800 MHz frequencies. The carrier when a call is active or being activated is for a period of about 500 μs duration out of a 4-ms cycle (since each frequency can be shared by up to eight phones). When inactive, the mobile phone will send signals to the base station once in every half-hour. In addition, on a moving vehicle extra signals are sent when the phone switches between base stations. It should be noted that these cases of interference may have impact only when the mobile phone is lying close to the network cables. We assume a typical interval between bursts to be 30 s, i.e., $T_{\rm phone} = 30$ s and $l_{\rm phone} = 500\ \mu$s.

For our second interference source, *viz.*, radar transmissions from a ship, we assume the duration of an interference burst to be 1 ms with a single burst in each interference, i.e., $l_{\rm radar} =$ 1 ms and $n_{\rm radar} = 1$.

B. Experiments

To illustrate our method, we will provide the following analysis of our simple ABS system:

- classical worst-case analysis, without considering any faults, using the analysis of Section III-A;
- worst case analysis under worst case fault phasings, using the analysis of Section III-B;
- worst-case analysis under arbitrary fault phasings, using the analysis of Section III-C.

Since we are considering two independent sources of interference, three cases will be considered: 1) interference from the mobile phone only; 2) interference from the radar only; and 3) interference from both the sources.

In combining the obtained results to an overall reliability estimate for the considered mission time of 8 h, we will make the following assumptions.

- If a mobile phone is lying too close to a network cable, it will remain there for the entire mission.
- The probability that a mobile phone is lying too close to a network cable is 10^{-4}.
- The probability of passing a bridge (under which a ship equipped with a powerful radar may pass) is for each mission 0.5.
- The probability that a ship equipped with a powerful radar is passing under a bridge when a specific car (which is known to pass such a bridge during its current mission) is passing is 10^{-3}.
- The probability of a ship having activated its radar while passing a bridge is 0.7.

From the above information, we derive: $P^{\rm act}_{\rm phone} = 10^{-4}$ and $P^{\rm act}_{\rm radar} = 0.5 \times 10^{-3} \times 0.7 = 3.5 \times 10^{-4}$. Note that we have no strong basis for the assumed values above. They merely represent intelligent guesses indicating the type of parameters that need to be identified.

C. Reliability Analysis Results

We have performed a reliability analysis on our example message set. First, we calculated the response time without interference using (15) in Section III-A. The resulting values (in milliseconds) in column (0) in Table III show that the message set is schedulable under interference-free conditions.

Next, we analyzed the system under worst case interference phasings using the analysis from Section III-B. The results, presented in columns (1)–(3) in Table III, show that some messages miss deadlines (indicated by a “*”) in all three cases: interference from phone only [column (1)], radar only [column (2)], and both sources [column (3)].

TABLE III
RESPONSE TIME ANALYSIS—NORMAL AND UNDER FAULTS

Msg ID	Priority	T_i	D_i	C_i	Response Time			
					(0) no interf.	(1)phone	(2)radar	(3) phone&radar
OPERATOR-1	1	8	8	0.54	1.08	2.24	2.74	3.549
ABS-1	2	4	4	0.54	1.64	2.78	3.28	* 4.374
ABS-2	3	4	4	0.54	2.16	3.32	3.82	* 6.131
ABS-3	4	4	4	0.54	2.70	3.86	* 4.36	* 7.339
ABS-4	5	4	4	0.54	3.24	* 4.40	* 6.52	* 8.419
OPERATOR-2	6	15	15	0.54	3.78	7.86	7.60	*16.384

TABLE IV
SIMULATION RESULTS: SIMPLE FAILURE SEMANTICS

Interference Sources	Average Case in Simulation			
	Total Messages	Missed Deadlines	Failure Probability	99.9% Confidence Interval
phone only	4530000	4385	96.8×10^{-5}	$\pm 4.2 \times 10^{-5}$
radar only	4530000	5691	125.6×10^{-5}	$\pm 5.3 \times 10^{-5}$
phone and radar	4530000	12331	272.2×10^{-5}	$\pm 8.3 \times 10^{-5}$

TABLE V
SIMULATION RESULTS: REFINED FAILURE SEMANTICS

Interference Sources	Average Case in Simulation			
	Total Messages	Number of Failures	Failure Probability	99.9% Confidence Interval
phone only	4530000	0	0	0
radar only	4530000	0	0	0
phone and radar	4530000	1015	22.4×10^{-5}	$\pm 6.6 \times 10^{-5}$

According to the analysis under worst case assumptions, we can see that one out of six and two out of six messages miss their deadlines considering individual interference from phone only and radar only, respectively, while five out of six messages miss their deadlines under combined interference from both the sources.

We finally conducted simulation runs by varying the points of start of interference according to Algorithm 2 in the Appendix. The results are shown in Table IV. Combining the obtained failure probabilities, using the probabilities for the different interference scenarios derived from the assumptions in the previous section, we get the following failure probability formula [derived from (14)]:

$$\begin{aligned} Q_F^{\mathcal{T}}(\{\text{phone}, \text{radar}\}) = {} & P_{\text{phone}}^{\text{act}}(1 - P_{\text{radar}}^{\text{act}}) \times 0.000\,968 \\ & + P_{\text{radar}}^{\text{act}}(1 - P_{\text{phone}}^{\text{act}}) \times 0.001\,256 \\ & + P_{\text{phone}}^{\text{act}} \times P_{\text{radar}}^{\text{act}} \times 0.002\,722 \end{aligned} \tag{21}$$

which evaluates to 5.36 × 10^{-7}.

Unfortunately, this is an unacceptable high failure probability compared to the admissible failure probability of 10^{-9}. However, returning to the discussion on failure semantics at the end of Section II-C, we note that systems that fail due to a single deadline miss should be avoided, since they are extremely sensitive. In particular, for critical systems, the designer has an obligation to make the system more robust. For instance, for our simple system a more reasonable failure semantics would be: "a failure occurs if more than two out of ten deadlines are missed." It should be noted that changing failure semantics may have implications for the design, since we have to make sure that the system appropriately can handle the new situation, e.g., in our case that a few deadline misses can be tolerated.

Table V reports the results from a simulation of exactly the same system as above, except that we now use the new failure semantics.

Recalculating the overall mission failure probability $Q_F^{\mathcal{T}}$ using the failure probabilities from Table V gives $Q_F^{\mathcal{T}}(\{\text{phone}, \text{radar}\}) = 7.84 \times 10^{-12}$, which is quite negligible in relation to the required failure probability of 10^{-9}. This means that the system has sufficiently high reliability to be acceptable. Hence, our analysis has relieved the designer from the costly redesign process which would have been chosen if only the analysis with worst case phasing was considered.

V. CONCLUSIONS

We have presented a method that allows controlled relaxation of the timing requirements of safety-critical hard real-time systems. The underlying rational is that no real system is (or can ever be) hard real time, since the behavior of neither the design nor the hardware components can be completely guaranteed. By integrating hard real-time schedulability with the reliability analysis normally used to estimate the imperfection of reality, we obtain a more accurate reliability analysis framework with high potential for providing solid arguments for making design tradeoffs, e.g., that allow a designer to choose a slower (and less expensive) bus or CPU, even though the timing requirements are violated in some rare worst case scenario.

Using traditional schedulability analysis techniques, the designer will in many cases have no other choice than to redesign the system (in hardware, software or both). However, by resorting to our new analysis, we may see that the probability of an extreme error situation arising is very low and thus the designer may not need to perform a costly redesign.

It is well known [8] that a control system that fails due to a single deadline miss is not robust enough to be of much practical use. Rather, the system should tolerate single deadline misses, or even multiple deadline misses or more complex requirements on the acceptable pattern of deadline misses. These requirements should of course be derived from the requirements on stability in the control of the external process. The possibility to handle such requirements in the analysis can makes the use of the resources even more efficient, i.e., we achieve a tradeoff situation between algorithmic fault tolerance and resource usage. By considering each message separately in our example we could increase the reliability by incorporating algorithmic fault tolerance for functions which are dependent on a message that has the lowest reliability.

The presented method is tailored for analysis of the effects of external interference on CAN bus communication. The method could be extended in various directions, such as including stochastic modeling of external interference, distributions of transmission times due to bit stuffing, distributions of actual queuing times, distributions of queuing jitter, as well as applying the framework to CPU scheduling, including variations in execution times of tasks, jitter, periods for sporadic tasks, etc. Some of these extensions require dependency issues to be carefully considered. For instance, message queuing jitter may for all messages on the same node be dependent on interrupt frequencies. Assuming independence in such a situation may lead to highly inaccurate results. Another critical issue which should be given further attention is the sensitivity of the analysis to variations in model parameters and assumptions, such as the assumed probabilities for the interference sources to be active in our example.

We are convinced that a successful development of a holistic analysis framework, taking both reliability and schedulability (as well as other pertinent issues) into account, will be of immense value for the development of resource constrained safety-critical real-time systems. The method presented here is an important step in that direction.

Appendix
Simulation Algorithms

The following are high-level pseudocode descriptions of the algorithms for the simulation based analysis in Section III-C.

Algorithm 1 (Exhaustive Simulation)
The algorithm takes as input a list of interference sources A and returns the probability that an interference scenario causes a communication failure.

```
Algorithm Ex_sim($A$: “list of
  interference sources”):
probability
sample: array [“A”] of int;
Function startsample($k$: interference
  source): Int
  If $k \in \mathcal{I}$ {intermittent source}
  then startsample := $0$
  else {transient source}
    startsample := $-n_k T_k$;
end.func
Function maxsample($k$: interference
  source): Int
  If $k \in \mathcal{I}$ {intermittent source}
  then maxsample := $T_k - 1$
  else {transient source}
    maxsample:= $\mathcal{T} - 1$;
end.func
Function simulate(“list of phasings”):
  1/0
  “Simulate behavior during $\mathcal{T}$”
  If deadline_violation
  then simulate := $1$
  else simulate := $0$;
end.func
begin.alg
  scenarios := $0$;
  fail := $0$;
  For sample[First(A)]
    := startsample(First(A)) to
      maxsample(First(A)) do
    For sample[next(First(A))] :=
        startsample(next(First(A))) to
        maxsample(next(First(A))) do
    ⋮
      For sample[Last(A)] :=
        startsample(Last(A)) to
        maxsample(Last(A)) do
          scenarios := scenarios + 1;
          fail := fail +
            simulate(sample);
      od
    ⋮
    od
  od
  Ex_sim := fail/scenarios;
end.alg
```

The algorithm Ex_sim investigates for each combination of phasings if the interference sources cause a communication failure or not.

Algorithm 2 (Random Simulation)
The algorithm takes as input a list of interference sources A and returns the probability that an interference scenario causes a communication failure.

```
Algorithm Rnd_sim($A$: “list of interfer-
ence sources”):
probability
sample: array [“A”] of int;
begin.alg
  scenarios := $0$;
  fail := $0$;
  Do until confidence $\geq$ required_confi-
dence
    For a = First(A) to Last(A) do
      sample[a] :=
        pick_a_sample(startsample(a),
        maxsample(a))
    od
    scenarios := scenarios + 1;
    fail := fail + simulate(sample);
  od
  Rnd_sim := fail/scenarios;
end.alg
```

```
The algorithm $Rnd\_sim$ randomly selects
  phasings of the interference sources and
  investigates if the selected combination
  of interference causes a communication
  failure. The procedure is repeated until
  required level of confidence is reached.
```

Acknowledgment

The authors wish to express their gratitude to R. Elvsén for useful discussions and to the anonymous reviewers for their helpful comments.

References

[1] H. Hansson, C. Norström, and S. Punnekkat, "Integrating reliability and timing analysis of CAN-based systems," in *Proc. 2000 IEEE Int. Workshop Factory Communication Systems (WFCS'2000)*, Porto, Portugal, Sept. 2000, pp. 165–172.

[2] ——, "Reliability modeling of time-critical distributed systems," in *Formal Techniques in Real-Time and Fault-Tolerant Systems*, M. Joseph, Ed. Pune, India: 6th Int. Symp., FTRTFT 2000, Springer-Verlag, Sept. 2000, vol. 1926, Lecture Notes in Computer Science (LNCS).

[3] N. C. Audsley, A. Burns, M. F. Richardson, K. Tindell, and A. J. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Software Eng. J.*, vol. 8, no. 5, pp. 284–292, Sept. 1993.

[4] L. Sha, R. Rajkumar, and J. P. Lehoczky, "Priority inheritance protocols: An approach to real-time synchronization," *IEEE Trans. Comput.*, vol. 39, pp. 1175–1185, Sept. 1990.

[5] J. Xu and D. L. Parnas, "Priority scheduling versus pre-run-time scheduling," *Real-Time Syst. J.*, vol. 18, no. 1, pp. 7–23, Jan. 2000.

[6] S. Punnekkat, H. Hansson, and C. Norström, "Response time analysis under errors for CAN," in *Proc. IEEE Real-Time Technology and Applications Symp. (RTAS 2000)*, June 2000, pp. 258–265.

[7] L. M. Pinho, F. Vasques, and E. Tovar, "Integrating inaccessibility in response time analysis of can networks," in *Proc. 2000 IEEE Int. Workshop Factory Communication Systems (WFCS'2000)* Porto, Portugal, Sept. 2000, pp. 77–84.

[8] M. Törngren, "Fundamentals of implementing real-time control applications in distributed computer systems," *Real-Time Syst. J.*, vol. 14, no. 3, pp. 219–250, May 1998.

[9] K. W. Tindell and A. Burns, "Guaranteed message latencies for distributed safety-critical hard real-time control networks," Dept. Comput. Sci., Univ. York, York, U.K., Tech. Rep. YCS229, June 1994.

[10] K. W. Tindell, H. Hansson, and A. J. Wellings, "Analysing real-time communications: Controller area network (CAN)," in *Proc. 15th IEEE Real-Time Systems Symp.*, Dec. 1994, pp. 259–265.

[11] N. Navet, Y.-Q. Song, and F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over controller area network," *J. Syst. Architect.*, vol. 7, no. 46, pp. 607–617, Sept. 2000.

[12] M. Lindgren, H. Hansson, C. Norström, and S. Punnekkat, "Deriving reliability estimates of distributed real-time systems," in *Proc. 7th Int. Conf. Real-Time Computing Systems and Applications (RTCSA 2000)* Cheju Island, South Korea, Dec. 2000, pp. 279–287.

Hans A. Hansson (A'01) received the M.Sc. degree in engineering physics, the Licentiate degree in computer systems, the B.A. degree in business administration, and the Doctor of Technology degree in computer systems from Uppsala University, Uppsala, Sweden, in 1981, 1984, 1984, and 1992, respectively.

He is currently a Professor of Computer Engineering at Mälardalen University, Västerås, Sweden. He is also Director of the Mälardalen Real-Time Research Centre and Programme Director for the ARTES, a national Swedish real-time systems initiative. He was previously Department Chairman and Senior Lecturer in the Department of Computer Systems, Uppsala University, Chairman of the Swedish National Real-Time Association (SNART), and a Researcher at the Swedish Institute of Computer Science, Stockholm, Sweden. His research interests include real-time system design, reliability and safety, timed and probabilistic modeling of distributed systems, scheduling theory, distributed real-time systems, and real-time communications networks.

Prof. Hansson is a member of the Association for Computing Machinery and SNART.

Thomas Nolte (S'01) received the B.Eng. degree in 2000 from the Department of Computer Engineering, Mälardalen University, Västerås, Sweden, where he is currently working toward the Ph.D. degree in the Systems Design Laboratory.

His current research interests are reliability and timing analysis of distributed systems. He has also done some work on modeling and analysis of the bit-stuffing mechanism used in the Controller Area Network.

Christer Norström (A'01) received the Ph.D. degree from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 1997.

He is currently Manager for Motion Control and Applications at ABB Technology Partners/Robotics, Västerås, Sweden. He is also a part-time Senior Lecturer in the Department of Computer Engineering, Mälardalen University, Västerås, Sweden, and one of the founding members of the department. He has been a Consultant, in particular, for the automotive industry. He has taught numerous courses on real-time systems for industry both in Sweden and in Europe. His research interests are design of real-time systems, reliability and safety methods, software engineering, and architectures for real-time systems. He is very interested in technology transfer from academia to industry and he has manifested this through several successful transfers to the automotive industry. He was previously Chairman of the Department of Computer Engineering, Mälardalen University.

Dr. Norström was recognized as best teacher at Mälardalen University in 2001. He is a member of the Swedish National Real-Time Association.

Sasikumar Punnekkat (M'94) received the Master of Statistics degree and the Master of Technology in Computer Science degree with honors from the Indian Statistical Institute, New Delhi, India, in 1982 and 1984, respectively, and the Doctor of Philosophy degree in computer science from the University of York, York, U.K., in 1997.

He is a Senior Scientist in the Software Quality Assurance Division, Vikram Sarabhai Space Centre, Trivandrum, India. During 1999–2000, he was a Research Fellow in the Department of Computer Engineering, Mälardalen University, Västerås, Sweden. He joined the Indian Space Research Organization in 1984 and was involved in the design, development, and testing of software for satellite launch vehicles. His research interests span various aspects of real-time systems, fault-tolerant computing, and software engineering.

Dr. Punnekkat was a recipient of the prestigious Commonwealth Scholarship during 1993–1997.