Evaluating the temporal behaviour of CAN based systems by means of a cost functional

Roland Rüdiger

In the past decade much work has been dedicated to an analysis of performance characteristics of CAN based systems. The present contribution outlines a project which continues and extends earlier research by other authors. Following an idea borrowed from stochastic control theory, we introduce a cost functional C with the expectation value $\mathbb{E}[C] = \sum_{m=1}^{M} c_m \mathbb{P}[R_m > D_m]$, which serves as a quality measure of *timeliness*. Here, M is the number of priority classes, $\mathbb{P}[R_m > D_m]$ is the probability that a frame of priority class m does not meet its deadline D_m, and the coefficient c_m denotes the cost resulting from this event. Several methods of determining these probabilities are described and compared. A close examination of the expected costs yields useful information about the qualtity of the system's temporal behaviour. In particular, the analysis can be applied if the system is perturbed by stochastic noise. Preliminary results are presented which were obtained by applying recent work of Navet & Song as well as by simulation using the simulator package NetSim of i+ME.

1 Introduction

This paper outlines a project which deals with the temporal behaviour of real time systems under the influence of a noisy environment. As an example of outstanding importance the Controller Area Network (CAN) protocol is considered. Our approach is based on the concept of cost functionals (or simply: cost functions), which is sufficently general to be applicable to other systems in an analogous way. Cost functions, when suitably defined on basis of empirical data, can be regarded as a kind of quality measure or metric. In the context of the present paper, as explained in detail in section 3, we shall consider a quality measure of *timeliness*. A detailed discussion of real time metrics may be found in the book of Nissanke [25].

The approach of this paper is formal. In the research area of high speed networks there exists a countless number of papers applying formal and analytical methods. Presently, in the area of fieldbus systems, there seems to be no comparably widespread use of this techniques. But the situation is changing.

This contribution gives an account of the methods used in the project and presents some preliminary results. In section 2 a brief survey of priority queueing systems is given; in particular, an early application of the classical Cobham formula to an error model of CAN is mentioned. In section 3 the idea of a cost functional is discussed. The traditional approach presumes that *waiting* causes costs. Alternatively, the approach forming basis of the work presented here, presumes that costs arise in the case that messages miss their deadlines. This requires to apply methods for determining probability distribution functions of response times. Therefore, in section 4, several of these methods are considered and compared. One approach uses well known soujourn time distribution of queueing theory. The technical problem consists in numerically inverting Laplace transforms. Another approach relies on recent work of Navet & Song who extend the worst case analysis by Tindell et al. by a stochastic error model. Last but not least, simulation is one approach which might give some insight even in highly complex systems. Finally, in section 5, some preliminary results are presented, which demonstrate the potential and shortcomings of the methods. Presently, none of the methods gives one unique final answer to any question of interest. In a table, the methods are compared with each other, and some of their benefits and shortcomings are shown. Section 6 concludes the paper with a summary and some perspectives.

Research of this kind is, by its very nature, fairly formal. Nobody will expect a practioner to learn probability theory and stochastics before he or she will apply the methods presented here. Therefore, eventually, all of the formalism has to be cast in tools which can be used without knowledge of the formal machinery behind the scenes. This constitutes part of future work.

1.1 Controller Area Network (CAN)

The CAN protocol is a member of a class of protocols usually referred to as "autobus" protocols [18]. Nowadays, applications of CAN exist in many industrial areas. However, the decisive impetus for the growing market of CAN interfaces, which has nowadays established itself as a mass market, came from the decision of the automobile industries to apply CAN. Sales forcast of CAN accumulated production volume by the end of this century prognosticate a total of 140 million units [20], [21, p. 29]. With respect to the work presented here this large number of units is the key argument to apply probabilistic and stochastic methods. Detailed monographs about CAN are e.g. [19], [7], and [21].

1.2 Real time systems

In the past decade there has been some debate about the real time behaviour of the CAN protocol. It was argued that CAN, being part of the class of random access protocols, cannot have deterministic response times. Although this is certainly true, it only goes halfway towards finding a complete answer. Meanwhile, a lot of work has been devoted to this question and matters seem to have been clarified to a large extent.

Usually, two types of real time requirements are considered: soft real time, and hard real time. The essential difference is that, in hard real time systems, timing requirements have to be met strictly and without any exception. However, in ordinary technical systems one cannot be absolutely sure of anything: there might always be some exceptions. Of course, there's no point in forming stochastic models of situations which might occur with some excessively low probability only of, say, 10^{-40}.

If, however, probabilities of any events, which might realistically occur, have a broader range from "technically reasonable" to "unreasonably low" then a stochastic model should be formed and, on a descriptive level, the conceptual difference between hard real time and soft real time disappears.

There exists one common problem with an interpretation of probabilities. We are always faced with the question: "What is a low probability?" One good answer, at least in principle, is to look at the expected cost caused by an event which will harmfully effect a system. In his textbook on real time systems, Kopetz [16, p. 304] writes: *"Critical Failure: A failure is critical if the cost of the failure can be orders of magnitude higher than the utility of the system during normal operation."* Here, *safety critical real-time computer system* is used synonymously for *hard real-time computer system*. Clearly, even if one can be certain that the probability of some disastrous event will be very low this knowledge is of little use if, at the same time, the resulting cost from this event will be exorbitantly high.

It is in fact this point of view which will be made precise and applied in the context of CAN.

2 Priority queueing systems

Collision resolution on a CAN bus is achieved by assigning priorities to the message classes. Therefore, many features of CAN based systems can be modelled using priority queueing systems. Since the pioneering work of Cobham [3, 4] priority queueing systems have extensively been studied.

Table 1: Classification of priority queueing systems

<table>
<tr><td>non-preemptive discipline
(relative priorities)</td><td colspan="2"></td></tr>
<tr><td rowspan="3">preemptive discipline
(absolute priorities)</td><td>preemptive resume
(with work conservation)</td><td></td></tr>
<tr><td rowspan="2">preemptive repeat
(without work conservation)</td><td>without resampling</td></tr>
<tr><td>with resampling</td></tr>
<tr><td>alternating priorities</td><td colspan="2"></td></tr>
</table>

2.1 Classification

Table 1 shows the standard classification of priority queueing systems as it can be found in the literature (see e.g. [9]). If there is no external noise present the familiy of non-preemptive systems reproduces essential properties of CAN adequately. Nevertheless, these systems have also been used to model noisy systems, see [6].

Many generalizations of these classical systems have been discussed in the literature. In particular, queueing systems with preemption-distance priorities as investigated by Herzog [10, 11] as well as queueing systems with server vacations might be useful for forming error models (see the references in [9, p. 274]). These systems are beyond the scope of the present paper.

2.2 Cobham's formula

The classical work by Cobham [3, 4] initiated the investigations on priority queueing systems. For a non-preemptive scheduling strategy the following equation (*Cobham's formula*) gives the expected waiting time for a given message class:

$$\mathbb{E}\left[W_m\right] = \frac{W_0}{\left(1 - \sum_{j=1}^{m} \rho_j\right)\left(1 - \sum_{j=1}^{m-1} \rho_j\right)} \quad \text{for } m = 1, 2, \ldots, M \tag{1}$$

where W_0 is the expected residual service delay:

$$W_0 = \frac{1}{2} \sum_{j=1}^{M} \rho_j \, \mathbb{E}\left[S_j\right] \left(1 + \xi_j^2\right)$$

(ρ_j: load factor, $\mathbb{E}\left[S_j\right]$ and ξ_j: expected value and variational coefficient of transmission/service time of message/job class j)

A comparative analysis of CAN and ABUS by Dudeck et al. [6] has been based on Equation (1). Within their model these authors have also considered the effect of errors on transmission time. The extra delay of messages resulting from bus errors is taken into account by introducing some variation of service time.

3 Cost functional

The notion of a cost function has proven to be useful in many fields, e.g. in the theory of control of stochastic processes, see [23]. Some of the advantages of this conception are the following

- A cost function may be used as target for optimization.

- Quality aspects of systems and configurations of a different kind can be compared on the basis of one quality measure instead of considering a whole bunch of probability distribution functions.
- It is possible to evaluate a system as part of a larger system.

3.1 Basis of evaluation: *expected waiting time*

In the following we briefly describe one of the well-known results of queueing theory based on a cost function. If *waiting* involves charges, we can write

$$\text{cost}_m = \gamma_m W_m$$

for message (job) class m with some constant γ_m $(m = 1, 2, \ldots, M)$. Assigning priorities may be written as a permutation π_i, where $1 \leq i \leq M!$. Then, the (total) cost is a random variable $C^{(i)}$ with mean $\mathbb{E}\left[C^{(i)}\right]$:

$$C^{(i)} = \sum_{m=1}^{M} \gamma_m W_m^{(i)}, \qquad \mathbb{E}\left[C^{(i)}\right] = \sum_{m=1}^{M} \gamma_m \, \mathbb{E}\left[W_m^{(i)}\right]. \tag{2}$$

One of the established results due to Kleinrock [14] states that this expectation value will strictly be minimized in the following way: if the quantities γ_m/ρ_m are sorted in descending order according to

$$\frac{\gamma_{m_1}}{\rho_{m_1}} \geq \frac{\gamma_{m_2}}{\rho_{m_2}} \geq \ldots \geq \frac{\gamma_{m_M}}{\rho_{m_M}}$$

then the assignment of priorities $\pi_{opt} = \binom{m_1, m_2, \ldots, m_M}{1, 2, \ldots, M}$ minimizes the expected cost:

$$\mathbb{E}\left[C^{(opt)}\right] \leq \mathbb{E}\left[C^{(i)}\right] \text{ for } 1 \leq i \leq M!.$$

This procedure reduces the original problem of complexitity $M!$ to the problem of sorting an array.

3.2 Basis of evaluation: *timeliness*

There are situations where *timeliness* is of primary concern rather than *high throughput*. In this case, some cost will incur if deadlines are missed. An approach based on a cost function which models this situation adequately would be the following. In case of violating the timing constraint $R_m \leq D_m$, where R_m and D_m are the response times and deadlines of messages of class m, respectively, some distastrous event might happen with resulting cost c_m. A system designer who wishes to establish a quality measure of this kind has to specify pairs (c_m, D_m) for each message class m. It might of course be a problem to choose reasonable values for c_m and in most cases there will be no unique solution. However, as Kleinrock [14, p. 126] remarks, quite often decisions are being made regarding the relative system performance among classes which imply some form of cost function, perhaps unknown to the user.
So, for a given message class m, the random variable cost_m may be written as:

$$\text{cost}_m = \begin{cases} 0 & \text{if } R_m \leq D_m \\ c_m & \text{if } R_m > D_m \end{cases}$$

and the total cost is

$$C^{(i)} = \sum_{m=1}^{M} c_m \Theta(R_m^{(i)} - D_m)$$

Here $\Theta(x)$ is the unit step function.
For the sequel the quantity of primary interest to us will be the average cost:

$$\mathbb{E}\left[C^{(i)}\right] = \sum_{m=1}^{M} c_m \, \mathbb{P}\left[R_m^{(i)} > D_m\right] \tag{3}$$

It is exactly this quantity which has to be considered by an insurance company in the process of determining a premium.
Unfortunately, there seems to exist no theorem analogous to the algorithm of subsection 3.1, which allows to minimize expression (3). In many situations, however, it will suffice to keep the cost reasonably low without minimizing it in a strict sense. This will be discussed in section 5 in more detail.
Nevertheless, a necessary condition on some threshold of the cost function can be set up. If for any of the message classes stringent timing conditions are important then a system, which is operated in a range where $\mathbb{E}\left[C^{(i)}\right] \approx \min_{1\le m\le M} c_m$, behaves in such a way as if one of the messages would almost always be late.
Therefore, operating the system in a technically reasonable way requires that at least the following holds true:

$$\mathbb{E}\left[C^{(i)}\right] \ll \min_{1\le m\le M} c_m \tag{4}$$

Of course, adopting expression (3) as a basis is just one approach; in particular, it would be possible and meaningful to include other cost factors which ultimately even might dominate the right-hand side of (3). In this case any considerations on optimal priority ordering would be obsolete.

4 Response times

In order to base an evaluation of timeliness on Equation (3) the cumulative complementary response time distribution has to be calculated. Unfortunately, in most cases, this is not easily obtained. In the following we summarize some of the methods which yield information on this distribution.

4.1 Worst case considerations

Models without stochastic noise

Under the assumptions that there is no external noise present and that the messages to be transmitted are periodic, upper bounds on the response times can be given. This situation has been considered and analyzed in all detail in a series of papers by Wang & al. [32] and by Tindell & al. [30, 28, 31], see also [21, p. 253]. At a first glance, this work might appear of no use to the problem of determining probabilities for missing deadlines, as presented in this paper. However, the extension of the worst case analysis as conceived by Navet & Song [24] yields an extremely useful estimate of these probabilities. The following subsection summarizes and reformulates some of their key results.
Following Tindell & al. we characterize a periodic message class m by a quadrupel of the form (C_m, T_m, J_m, D_m). Here, C_m is the transmission time of messages of class m with period T_m, queueing jitter J_m, and deadline D_m. An explicit expression for the transmission time is:

$$C_m = \begin{cases} \left(\lfloor (34 + 8s_m)/4 \rfloor + 47 + 8s_m\right) \tau_{bit} & \text{for standard CAN [2, part A]} \\ \left(\lfloor (54 + 8s_m)/4 \rfloor + 67 + 8s_m\right) \tau_{bit} & \text{for extended CAN [2, part B]} \end{cases} \tag{5}$$

For a detailed justification of a denominator of 4 see [21, p. 255]. Then, the queueing delay (*interference time*) of message class m satisfies

$$I_m = \max_{j \in lp(m)} C_j + \sum_{j \in hp(m)} \lceil (I_m + J_j + \tau_{bit}) / T_j \rceil C_j \quad \text{for } m = 1, 2, \ldots, M. \tag{6}$$

Here, τ_{bit} is the bit time and $hp(m)$ and $lp(m)$ are the sets of priority classes higher and lower than m, respectively.
These equations can numerically be solved for I_m by iteration.
Then, the response time is given by

$$R_m = C_m + J_m + I_m$$

with the additional constraint

$$R_m \leq D_m - J_m. \tag{7}$$

In [29] Tindell & al. extended their work by a deterministic error model.

Models incorporating stochastic noise

A realistic model incorporating external perturbations of a CAN bus has to take account of its stochastic nature. In a remarkable paper Navet & Song [24] (*CiA Research Award 1997*) have extended the worst case analysis by a stochastic error model. Their results yield estimates of the probabilities of missing deadlines and, consequently, estimates of the expected cost as decribed in section 3.2. The idea is to add one term in Equations (6) which introduces an extra delay in the case of bus errors.
If exactly k errors occur the interference time is implicitely determined by

$$I_m = E_m(k) + \max_{j \in lp(m)} C_j + \sum_{j \in hp(m)} \lceil (I_m + J_j + \tau_{bit}) / T_j \rceil C_j \tag{8}$$

where

$$E_m(k) = k \left(23 \cdot \tau_{bit} + \max_{j \in hp(m) \cup \{m\}} C_j \right).$$

Obviously, an increasing number of errors will increase the response time. So, at some value of k, the messages will no longer be scheduable or, more formally, Equation (8) will have no solution. Therefore, one can ask for the maximal number of errors $k_m^{(max)}$ keeping the messages of class m scheduable, i.e. which does not violate the timing constraint (7).
So far, the model is still purely deterministic. If, in addition, a stochastic model for the occurence of errors is adopted then probabilities of missing deadlines can be estimated.
In the following we briefly summarize some of the results relevant to the subject of the present paper. The mathematical arguments of [24] have slightly been straitened.
Let $X(t)$ be the total number of errors during an interval of length t; then the complementary probability distribution function (PDF) of $X(t)$, evaluated for $t = R_m^{(max)}$ at $k = k_m^{(max)}$, yields estimates of the probabilities required in Equation (3):

$$p_m = 1 - \mathbb{P}\left[X(R_m^{(max)}) \leq k_m^{(max)} \right].$$

The error model conceived by Navet and Song presumes that errors are either single errors or error bursts which occur with relative frequencies of $1 - \alpha$ and α, respectively, where $0 \leq \alpha \leq 1$. Let $N(t)$ be the total number of error events and y_i the number of errors involved in event i. Then $X(t)$ can be written as

$$X(t) = \sum_{i=1}^{N(t)} y_i.$$

It is natural to assume that the random variables y_i are independently identically distributed (i.i.d.).
Conditioning the distribution $\mathbb{P}[X(t) \leq k]$ on $N(t)$ yields

$$\mathbb{P}[X(t) \leq k] = \sum_{m=0}^{k} F_{mk} P_m \quad \text{for } k = 0, 1, 2, \ldots \tag{9}$$

where $F_{mk} = \mathbb{P}[S_m \leq k]$ is the PDF of

$$S_m = \sum_{i=1}^{m} y_i \tag{10}$$

and $P_m = \mathbb{P}[N(t) = m]$ is the probability density function (pdf) of $N(t)$. Note that $F_{mk} = 0$ for $m > k$.
Equation (10) essentially says that S_m is a sum of m i.i.d. random variables y_i. Therefore, its PDF is an mth convolution power of the PDF of y_i. Denote these distributions by F_m and A, respectively; then, F_m can be obtained through $F_m = A^{m\circledast}$ or, equivalently,

$$F_m = F_{m-1} \circledast A \quad \text{for } m = 1, 2, \ldots.$$

This equation, written explicitly, takes the form

$$F_{mk} = \sum_{i=1}^{k-m+1} a_i F_{m-1,k-i} \quad \text{for } m, k = 1, 2, \ldots. \tag{11}$$

Here, $a_i = \mathbb{P}[y_j = i]$ is the pdf of the random variables y_j.
With $F_{0k} = 1$ for $k = 0, 1, 2, \ldots$, Equation (11) allows to determine the matrix F_{mk}, provided the coefficients a_i are known.
So, in order to evaluate Equation (11), some hypothesis has to be accepted on the kind of distributions of $N(t)$ and y_i.
Navet & Song propose the following: error events form a Poisson stream, i.e. the pdf of $N(t)$ is

$$P_m = \frac{(\lambda t)^m}{m!} e^{-\lambda t}.$$

Then, the resulting distribution of $X(t)$ is a so-called generalized Poisson distribution (or mixed compound distribution).
Furthermore, they assume that the burst size u is distributed with parameter p ($p + q = 1$) according

$$\mathbb{P}[u = k] = kp^2 q^{k-1} \text{ for } k = 2, 3, \ldots.$$

Then, the distribution $a_k = \mathbb{P}[y_i = k]$ turns out to be (for details see [24])

$$a_k = \begin{cases} 0 & \text{for } k = 0 \\ 1 - \alpha + \alpha p^2 & \text{for } k = 1 \\ \alpha k p^2 q^{k-1} & \text{for } k \geq 2. \end{cases} \tag{12}$$

Key Equations (11) with (12) and (9) can be solved numerically. Some applications using cost functions will be presented in section 5.

There remains the task of undersetting these distributions empirically or, at least, to determine the distribution parameters by measurements. For further details, see [24].
As demonstrated in the present paper another problem is that a worst case analysis might be too pessimistic in some situations. A system can violate conditions (6) and (8) although the messages of all classes are scheduable as simulations show. The mini network in [21, p. 419] is one example. This will be discussed in more detail in section 5.1.

4.2 Traditional stochastic queueing systems

Another way of modelling CAN systems is to look at analytic queueing models. Queueing models such as the the classical systems investigated in [13, 5], see also [9], rely on the assumption of a Poisson stream of message arrivals which is not well adapted to show the behaviour of a mixture of sporadic and periodic messages. So, at best, this will give some general indications on the real time behaviour of CAN systems.
On the other hand, in the two or three past decades an immense knowledge on all kinds of priority queueing systems has been accumulated. In particular, systems with so-called server vacations might be interesting for forming error models of CAN systems. An excellent review on priority queueing systems may be found in [8].
Despite of the restrictions discussed here, a first step could be to look at the detailed response time distribution corresonding to the classical model [13, 5], thus generalizing results of Dudeck et al. [6].
Adopting some of the notational conventions introduced by Gnedenko & König [9], the key equations describing an M/GI/1/∞-PRIO[NP] system can be summarized as follows.
The busy-period distribution $S_{(m)}$ by demands of priority m and higher is given by

$$S^*_{(m)}(s) = B^*_{(m)}(s + \lambda_{(m)} - \lambda_{(m)} S^*_{(m)}(s)) \quad m = 1, \ldots, M \tag{13}$$

where $\lambda_{(m)} = \sum_{i=1}^{m} \lambda_i$, $B^*_{(m)}(s) = \sum_{i=1}^{m} \frac{\lambda_i}{\lambda_{(m)}} B^*_i(s)$ and $\lambda := \lambda_{(M)}$. Here, following a standard notationial convention, we denote by $X^*(s)$ the Laplace transform of the pdf of the random variable X, explicitely, $X^*(s) = \mathbb{E}\left[e^{-sX}\right]$ for $s \in \mathbb{C}$.
Although a closed-form solution of Equation (13) is available [9, p. 276] a numerical solution could more easily be obtained on basis of an iterative scheme.
Then, the waiting time is given by

$$W^*_m(s) = \frac{\overline{\rho}_M \gamma_{m-1}(s) + \sum_{j=m+1}^{M} \lambda_j \left(1 - B^*_j(\gamma_{m-1}(s))\right)}{s - \lambda_m + \lambda_m B^*_m(\gamma_{m-1}(s))} \tag{14}$$

where

$$\rho_{mn} = \sum_{i=1}^{m} \lambda_i \, \mathbb{E}\left[B_i{}^n\right], \text{ in particular } \frac{1}{\mu_i} := \mathbb{E}\left[B_i\right]$$

$$\overline{\rho}_m = 1 - \rho_{m1}$$

$$\gamma_m(s) = s + \lambda_{(m)} - \lambda_{(m)} S^*_{(m)}(s).$$

From this equation, the classical Cobham formula (1) may be reproduced through $\mathbb{E}\left[W_m\right] = -{W^*_m}'(0)$. Equation (14) generalizes the results by Dudeck et al. [6] mentioned above, provided the Laplace transform involved will be inverted.
So, the technical problem to deal with is a numerical inversion of the Laplace transform in Equation (14). An extensive literature exists on techniques of numerical inversion of Laplace transforms. An in-depth review has been given by Abate & Whitt [1].

So far, in our project, a few tests have been carried out but further investigation will be needed to obtain numerically stable solution under realistic conditions of a CAN bus, in particular to obtain numerically stable estimates of distribution tails which are required in Equation (3). Results will be published elsewhere.

4.3 Simulative Models

Simulation is the prominent method for studying complex systems. The main advantage is that realistic conditions can be implemented which could hardly ever be incorporated into an analytical model. Therefore, in the project presented here, we use the simulator NetSim [12], a commercial product of i+ME. A variant of this simulator with restricted functionality is distributed with the CAN monography [21]. Some preliminary results may be found in section 5. Simulative results of that section were obtained by an analysis of the files `*.tim`, see [21, p. 422] for details. Up to now, however, no serious statistical analysis has been performed. Instead, we simply take relative frequencies as probabilities.
On the other hand, there are at least two major drawbacks of simulation. Firstly, simulation times can be extremely large so that rare events such as missing deadlines will not be seen. Often however, these are of particular interest. We plan to include recent results on rare event simulation of queueing systems. A review with special emphasis on queueing systems has been given in [17]. Secondly, presently available simulators are not truely*extensible* in a contemporary understanding. Here, newer software technologies such as *Extensible Programming* as conceived by Wirth, Gutknecht, and others [33, 26, 22] should be applied. Among other things, this should open the possibility to use arbitray, user-defined distributions including empirical distributions in simulation without recompiling any part of the existing simulator. So, e.g., NetSim does not allow for the possibility to consider a distribution as in Equation (12).

4.4 Comparison of methods

Unfortunately, so far, there seems to exist no method which will answer all questions related to response times of CAN systems in all situations. All of the methods described in this section have their specific benefits and shortcomings. Table 2 summarizes and compares some of them.

5 Case studies

In order to give an idea of the performance measure introduced in the previous sections we start our discussion with a typical example. The curves plotted in Figure 1 represent cost functions obtained from a worst case analysis with the stochastic error model described in subsection 4.1. The analysis is based on data of a configuration ("PSA benchmark"), which has been introduced and discussed by Navet & Song [24]. More details will be given in subsection 5.2.
The parameters α, p, and λ of subsection 4.1 have been set to $\alpha = 0.1$, $p = 0.04$ (taken from [24]) and $\lambda/s^{-1} = 0, 20, 40, 60, 80$. Sections (a) and (b) of Figure 1 show cost functions for standard CAN and extended CAN, respectively. For simplicity, the cost coefficients c_m introduced in Equation (3) have been set to unity.
The shape of the curves in sections (a) and (b) is similar, but the curves of section (b) are shifted to the left resulting in some increase of cost.
If there are no errors on the bus ($\lambda = 0$) the cost function is a step function. This can easily be explained: decreasing the transfer rate, i.e. proceeding from lower to higher values of τ_{bit} in the diagram, tightens the timing constraints (7) because the response times are increased. As a consequence, messages will not keep up with these constraints. Therefore, at each of these steps, the expected cost is increased by 1 until the maximum is reached. The maximum of $\mathbb{E}[C]$ could be normalized to unity but the only effect would be a rescaling of the ordinate.

Table 2: Comparison of methods

Method	Benefits	Shortcomings
Worst-Case-Analysis	▹ exact formula exists ▹ can be solved by iteration ▹ algorithm is easy to implement	▹ special assumptions on interarrival times required ▹ estimate may be too pessimistic
Analysis of Stoch. QM	▹ exact statements on probabilities, expected cost, and asymtotic behaviour available ▹ dependence on parameters can be studied fairly easily ▹ method is more general than worst case analysis ▹ conceptually: hard and soft real time behaviour described in one model	▹ in traditional models: interarrival times are distributed exponentially ▹ therefore, in these models, cyclic messages are not included adequately ▹ formal complexity of analytical models can be high
Simulation	▹ realistic assumptions can be used in implementions (e.g. on interarrival times)	▹ simlulation is one single experiment for one configuration ▹ simulations be may extremely time consuming ▹ conventional simulation does not include rare events, which are of particular interest so: *rare event simulation* should be applied

Of course, from a practical point of view, bit times beyond the critical value τ_{bit}^c are not of interest. The critical bit time τ_{bit}^c is the upper limit of bit times such that for $\tau_{bit} < \tau_{bit}^c$ all messages are schedulable.
Figure 1 also shows the cost function for some other values of λ. The interesting point here is that the expected cost at values below τ_{bit}^c may be above 1. A system operated within this range behaves as if the messages of one of the priority classes would constantly miss their deadline or, to put it differently, operating a system in a noisy environment just below but close to the critical point τ_{bit}^c might be as "expensive" as operating the same system in the case $\lambda = 0$ above τ_{bit}^c where timing constraints will constantly be violated.
It appears to be one of the benefits of cost functions to offer some quantitative measure in this situation. So, from figures similar to Figure 1, one can derive limits e.g. on τ_{bit} and λ to operate a CAN system with the desired reliability.

5.1 A toy example

Although the worst case analysis combined with the error model conceived by Navet & Song is an excellent method to calculate the probabilities required in Equation (3) this estimate may occasionally be too pessimistic. Consider the toy example in [21, p. 418] ("Mini Network") which serves as an example to demonstrate the functionality of the i+ME simulator NetSim.
In this example four variants of a CAN network are considered which differ with respect to some of the configuration parameters such as transfer rate, assignment of priorities, and the offset (relative distance of transmissions). One of the variants (Demo 2) is shown in Table 3 in a syntactical form accepted by NetSim. Table 4 displays results obtained from a worst case analysis of all variants. Here, as a consequence of the extremely low value of deadline D_B, none of the configurations appears to be schedulable.
Simulation results show however that this is not true for variants 2, 3, and 4. In Figure 2, left section (a), probability distributions of response times are shown for each of the priority classes which demonstrate that message class B meets its deadline $D_B = 500\mu s$. Even in a noisy

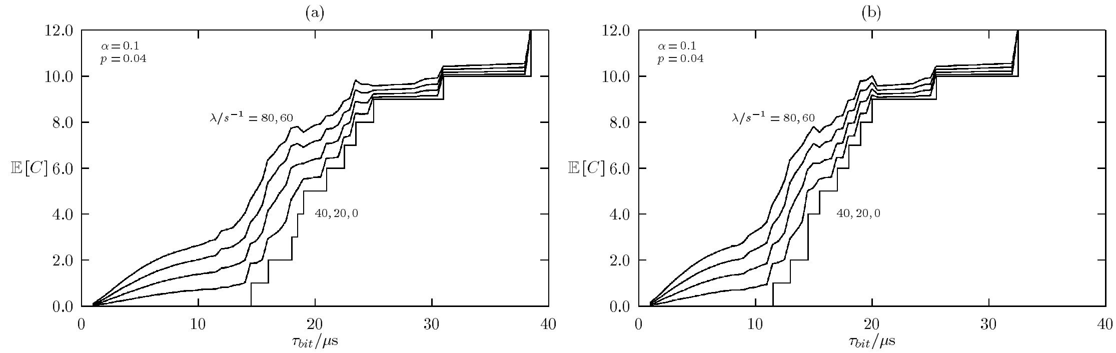

Figure 1: Overall structure of cost function for PSA benchmark and standard (a) and extended (b) CAN

NOISE GENERATORS DEFINITIONS							
NOISE	NOISE-REPETITION			NS. OFFSET	NOISE-LENGTH		
number	mode	t_1/ms	t_2/ms		mode	t_1/ms	t_2/ms

NOISE GENERATORS DEFINITIONS							
NOISE	NOISE-REPETITION			NS. OFFSET	NOISE-LENGTH		
number	mode	t_1/ms	t_2/ms		mode	t_1/ms	t_2/ms
1	exp	50.0		0.0	const	0.04	

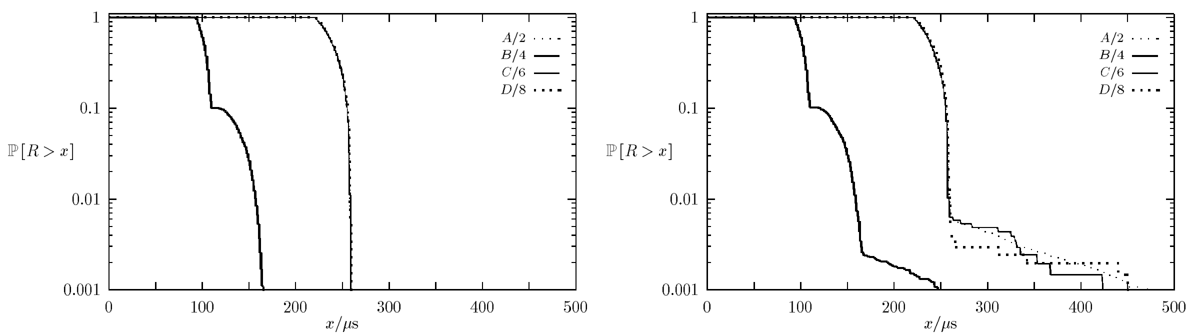

Figure 2: Complementary response time distribution of mini network without and with noise (left and right sections). Simulative results

environment with a resulting error rate of $\lambda = 20\text{s}^{-1}$ the probability of missing the deadline D_B is far below 10^{-3} (right section (b) of Figure 2).

Finally, Figure 3 shows detailed shapes of the cost function around τ^c_{bit}. The outcome of the worst case analysis suggests that τ_{bit} should be lowered to approximately 1.5μs in order to make the system schedulable.

In spite of these considerations the worst case analysis together with the error model of Navet & Song is a good method to determine probabilities required in Equation (3). The arguments presented so far should simply clarify the point that other methods are not obsolete.

5.2 PSA benchmark

Navet & Song have illustrated their error model [24] by an example proposed by Peugeot-Citroën Automobiles Company ("PSA benchmark"). Some of the properties of this model are summarized in Table 5. There are some minor differences in this table compared to [24], which are due to the fact that we adopt a divisor of 4 in Equation (5) and use in the first term of Equation (6) the value $\max_{j \in lp(m)} C_j$ instead of a transmisson time corresponding to 8 data bytes. In general, this places slightly tighter bounds on the response times.

Figure 4 shows a set of cost functions below the critical value τ^c_{bit} for $\alpha = 0.1$, $p = 0.04$ (adopted from [24]) and for $\lambda/s^{-1} = 0, 20, 40, 60, 80$. Obviously, the values are fairly large even near

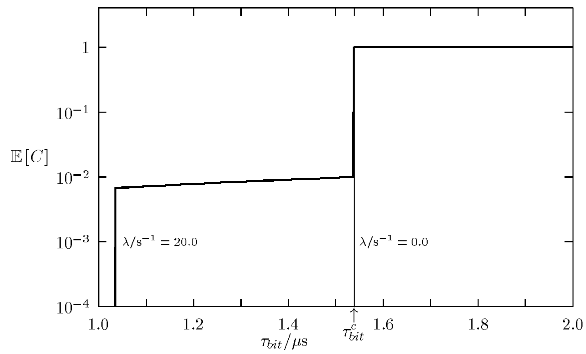

Figure 3: Cost function of mini network configuration around critical value τ^c_{bit}. Method of subsection 4.1

Table 3: Mini network configuration for simulation

NETLIST										
ND	MESSAGE NAME	ID	tx/rx	RTR	TXREPETITION mode	t_1/ms	t_2/ms	TXOFF /ms	RXDLY /ms	LGTH /byte
2	D: OIL_TEMPERATURE	8	tx		const	1000.0		0.000		8
2	A: OIL_PRESSURE	2	tx		const	200.0		0.400		8
3	C: GEAR_INFORMATION	6	tx		const	1000.0		0.800		8
4	B: BREAK_INFORMATION	4	tx		const	100.0		0.200		0
1	A: OIL_PRESSURE	2	rx	N					100.0	8
1	D: OIL_TEMPERATURE	8	rx	N					500.0	8
1	C: GEAR_INFORMATION	6	rx	N					500.0	8
1	B: BREAK_INFORMATION	4	rx	N					0.5	0

$\tau_{bit} = 1\mu$s, which would probably be unacceptable for practical purposes. So, special action has to be taken to deal with this situation.
The assignment of priorities as given in Table 5 has been adopted from [24]. It is interesting to compare the resulting configuration with the alternative obtained by reassigning the priorities according to the algorithm *Earliest Deadline First* (EDF) as proposed by Tindell et al. [29], [21].
Figure 5 shows cost functions for these configurations. Clearly, the dotted curves, which correspond to the original configuration, are above the curves obtained when EDF is applied. This confirms the well-known result that EDF is an optimal algorithm.

5.3 SAE benchmark

Finally, an example will be considered which is based on the control system of a prototype electric car [27], henceforth referred to as "SAE benchmark". A detailed discussion and analysis of this system can be found in [15, 30], see also [21, p. 261]. For convenience, we briefly reproduce in Table 6 a description of the network which connects a total of seven subsystems (Vehicle Controller (V/C), Batteries, Driver Inputs, Brakes, Transmission Control (*Trans*), Inverter/Motor Controller (I/M C), Instrument Panel Display (*Ins*)).
This table describes 17 priority classes containing signals 1 to 53 as defined in the SAE benchmark. The signals of a given class are either periodic ("P") or sporadic ("S"). Table 6 also corrects some minor errors in [21].
A timing analysis along the lines of [24] is shown in Table 7. For comparison, columns 1–5 reproduce results published in [21, p. 264].
For simplicity, we have set $\alpha = 0$ in the following (no error bursts). Figure 6 shows the results in case of standard (a) and extended (b) CAN for a wide range of error rates $\lambda/s^{-1} =$

Table 4: Mini network

Variant	Symbolic name	m	s_m /bytes	T_m /ms	D_m /ms	I_m /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms	on time?
Demo 1 ($\tau_{bit} = 4\mu$s)	A	2	8	200.0	100.0	0.540	1.080	156	99.672	✓
	B	4	0	100.0	0.5	1.080	1.300	–	–	–
	C	6	8	1000.0	500.0	1.300	1.840	785	499.920	✓
	D	8	8	1000.0	500.0	1.300	1.840	785	499.920	✓
Demo 2 ($\tau_{bit} = 2\mu$s)	A	2	8	200.0	100.0	0.270	0.540	314	99.764	✓
	B	4	0	100.0	0.5	0.540	0.650	–	–	–
	C	6	8	1000.0	500.0	0.650	0.920	1576	499.916	✓
	D	8	8	1000.0	500.0	0.650	0.920	1576	499.916	✓
Demo 3 ($\tau_{bit} = 4\mu$s)	B	1	0	100.0	0.5	0.540	0.760	–	–	–
	A	2	8	200.0	100.0	0.760	1.300	156	99.892	✓
	C	6	8	1000.0	500.0	1.300	1.840	785	499.920	✓
	D	8	8	1000.0	500.0	1.300	1.840	785	499.920	✓
Demo 4 ($\tau_{bit} = 4\mu$s)	A	2	8	200.0	100.0	0.540	1.080	156	99.672	✓
	B	4	0	100.0	0.5	1.080	1.300	–	–	–
	C	6	8	1000.0	500.0	1.300	1.840	785	499.920	✓
	D	8	8	1000.0	500.0	1.300	1.840	785	499.920	✓

Table 5: PSA benchmark

				$\tau_{bit} = 8\mu$s			$\tau_{bit} = 4\mu$s			$\tau_{bit} = 1\mu$s		
m	s_m /bytes	T_m /ms	D_m /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms
1	8	10.0	10.0	2.080	6	9.664	1.040	14	9.888	0.260	61	9.898
2	3	14.0	14.0	2.760	8	13.952	1.380	19	13.928	0.345	85	13.910
3	3	20.0	20.0	3.440	11	19.104	1.720	27	19.664	0.430	122	19.926
4	2	15.0	15.0	4.040	7	13.968	2.020	19	14.908	0.505	90	14.945
5	5	20.0	20.0	4.880	10	19.880	2.440	25	19.420	0.610	120	19.865
6	5	40.0	40.0	5.720	21	39.584	2.860	52	39.384	0.715	242	39.866
7	4	15.0	15.0	6.480	5	13.880	3.240	17	14.864	0.810	88	14.934
8	5	50.0	50.0	7.320	22	49.448	3.660	61	49.372	0.915	299	49.947
9	4	20.0	20.0	8.080	6	18.784	4.040	22	19.504	1.010	117	19.886
10	7	100.0	100.0	8.920	44	98.136	4.460	124	99.968	1.115	598	99.884
11	5	50.0	50.0	9.440	19	49.296	4.720	59	49.928	1.180	296	49.928
12	1	100.0	100.0	9.440	43	98.232	4.720	122	99.384	1.180	597	99.896

$0, 2, 10, 50, 250$.

Two points seem worth mentioning. Firstly, an interesting point about these plots is the following: at $\tau_{bit} \approx 2$ to 3μs the cost function takes values which are extremely low. This shows one of the strengths of the worst case analysis. In our model the value 1 (in some suitable unit) of the cost function means that the system cannot be operated safely; therefore, values around 10^{-16} guarantee that, under all practical circumstances, the system appears to be "absolutely" safe even in a noisy environment. One might expect that the cost resulting from failures of other electronic components will be considerably higher. So, in this special case, it would not be of interest to apply any other method of timing analysis which might yield even lower results.

Secondly, another interesting point about Table 7 appears to be that, although messages 6, 9, and 10 are schedulable at $\tau_{bit} = 8\mu$s (transmission rate of 125 kbps), the occurence of one single error would have the effect that all of these messages miss their deadlines. Therefore, costs resulting from this event should be relatively large. The detailed shape of the cost function near τ_{bit}^c, shown in Figure 7, gives a quantitative description of this fact.

Figure 7, right section (b), also shows three curves obtained by simulation. Compared with the results obtained from a worst-case analysis, these are substantially smaller. However, some of the underlying modelling assumptions are not exactly the same. Further investigations will be required to get some more insight.

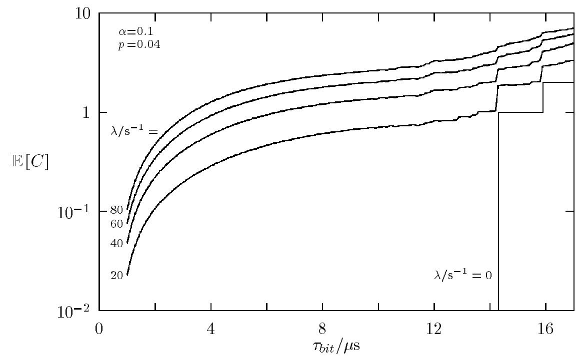

Figure 4: Cost function for PSA benchmark with standard CAN and small values of τ_{bit}

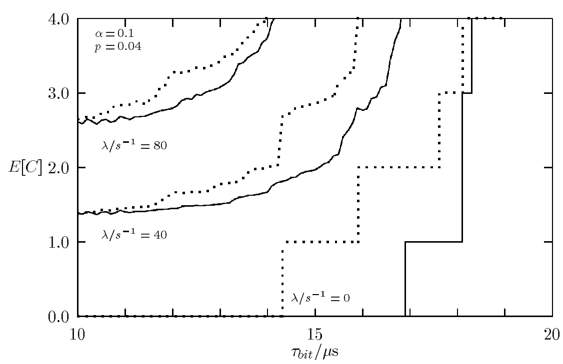

Figure 5: PSA benchmark with reassignment of priorities: original (dotted lines) and EDF

Table 6: SAE benchmark

from	prio to	V/C	Battery	Brakes	Trans	I/M C	Ins
V/C	**6**P		32			42	
	7S		34,35	37,38	31	40,44,46,48,53	39
	17P		29,30,33	36			
Battery	**1**S	14					
	8S	23,24,25,28					
	12P	1,2,4,6					
	15P	3,5,13					
Driver	**3**P	7					
	9S	15,16,17,19,20,22,26,27					
Brakes	**2**P	8,9					
	11S	18					
	13P	12					
Trans	**5**P	11					
	14P	10					
	16P	21					
I/M C	**4**P	43, 49					
	10S	41,45,47,50,51,52					

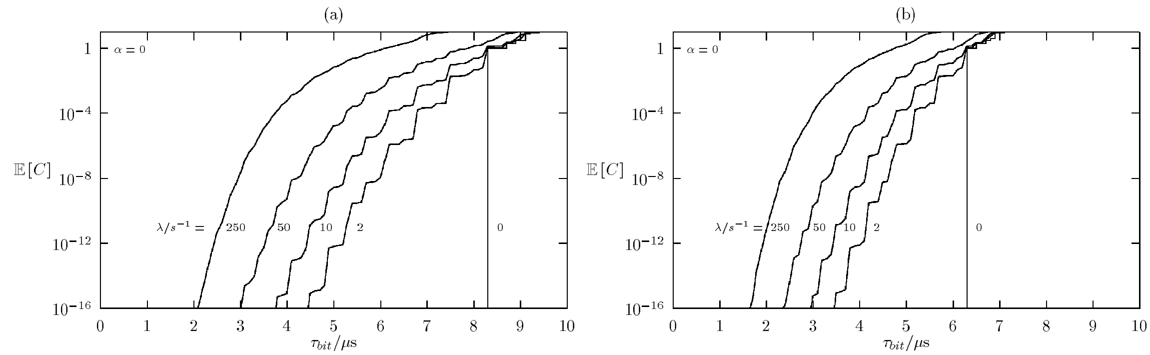

Figure 6: Cost function for SAE benchmark with standard (a) and extended (b) CAN for low values of τ_{bit}

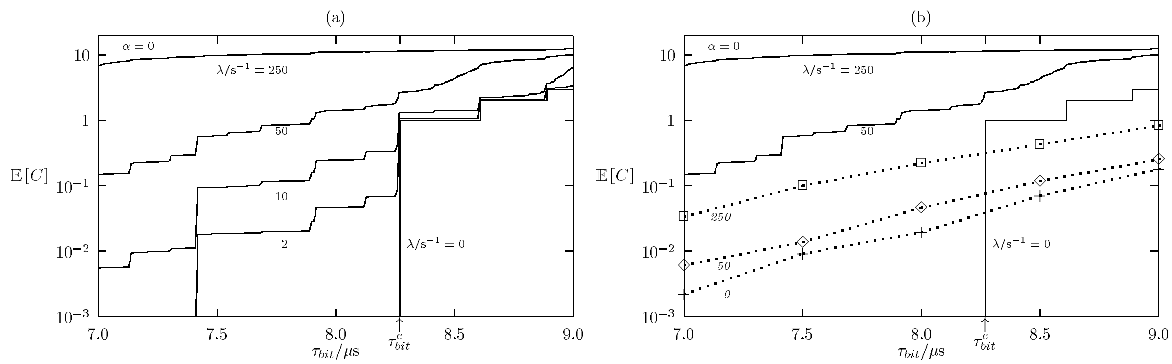

Figure 7: Cost function for SAE benchmark with standard CAN around critical value of τ_{bit}. Method of subsection 4.1 (a) compared with simulative results (b) (dotted lines, values of λ in italic type)

Table 7: SAE benchmark

				$\tau_{bit} = 8\mu$s			$\tau_{bit} = 4\mu$s			$\tau_{bit} = 1\mu$s		
m	s_m /bytes	T_m /ms	D_m /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms	R_m /ms	$k_m^{(max)}$	$R_m^{(max)}$ /ms
1	1	50.0	5.0	1.440	5	4.960	0.720	12	4.944	0.180	54	4.932
2	2	5.0	5.0	2.040	3	4.392	1.020	10	4.940	0.255	48	4.959
3	1	5.0	5.0	2.560	3	4.912	1.280	9	4.808	0.320	47	4.926
4	2	5.0	5.0	3.160	2	4.728	1.580	8	4.716	0.395	46	4.903
5	1	5.0	5.0	3.680	1	4.464	1.840	8	4.976	0.460	46	4.968
6	2	5.0	5.0	4.280	0	4.280	2.140	7	4.884	0.535	45	4.945
7	6	10.0	10.0	5.040	1	8.984	2.520	10	9.460	0.630	65	9.955
8	1	10.0	10.0	8.400	1	9.504	2.780	10	9.720	0.695	64	9.882
9	2	10.0	10.0	9.000	0	9.000	3.080	9	9.468	0.770	64	9.957
10	3	10.0	10.0	9.680	0	9.680	3.420	9	9.808	0.855	63	9.904
11	1	50.0	20.0	10.200	1	19.704	3.680	19	19.788	0.920	128	19.989
12	4	100.0	100.0	19.280	11	99.664	4.020	101	99.512	1.005	645	99.950
13	1	100.0	100.0	19.800	10	99.080	4.280	101	99.772	1.070	644	99.877
14	1	100.0	100.0	20.320	10	99.600	4.540	100	99.480	1.135	644	99.942
15	3	1000.0	1000.0	29.240	108	999.232	4.800	1014	999.728	1.200	6449	999.962
16	1	1000.0	1000.0	29.760	108	999.752	5.060	1014	999.988	1.265	6448	999.889
17	1	1000.0	1000.0	29.760	108	999.752	5.060	1014	999.988	1.265	6448	999.889

6 Conclusions

The present paper introduces a qualtity measure of *timeliness* of real time systems and discusses some applications to CAN based systems. The technical problem to deal with consists in determining probability distributions of response times. We report on several methods and apply these to various configurations, which have been described in the literature.
Many questions have not yet been answered. It would be important to look more closely at some points such as an application of rare event simulation, more sophisticated analytical queueing models, and a refined implementation of numerical Lapace inversion techniques.
To a large extent, the interest in this work relies on the fact that all of the methods considered here could as well be applied to systems other than CAN.

Acknowledgments

The author would like to thank R. Isernhagen, G. Kircher, and W. Lawrenz for valuable comments and clarifying discussions.

References

[1] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10:5–88, 1992.

[2] Robert Bosch GmbH. *CAN Specification. Version 2.0*, 1991.

[3] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 2:70–76, 1954.

[4] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 9, 1961.

[5] R.W. Conway, W.L. Maxwell, and L.W. Miller. *Theory of Scheduling*. Addison-Wesley, 1967.

[6] Ingo Dudeck, Ulrich Hipp, and Thomas Raith. Investigations into the electromagnetc compatibility and performance evaluation of databus systems in motor vehicles. *SAE 885083*, pages 1.629–1.635, 1988.

[7] K. Etschberger. *CAN Controller-Area-Network, Grundlagen, Protokolle, Bausteine, Anwendungen*. Hanser Verlag München Wien, 1994.

[8] Markus Ettl. Wartezeitverteilungen in Warteschlangensystemen mit Prioritäten. Master's thesis, Universität Erlangen-Nürnberg, 1990.

[9] B.W. Gnedenko and D. König. *Handbuch der Bedientheorie*, volume 2. Akademie-Verlag Berlin, 1984.

[10] Ulrich Herzog. Optimal scheduling strategies for real-time computers. *IBM Journal of research and development*, 19(5):494–504, 1975.

[11] Ulrich Herzog. Priority models for communication processors including system overhead. *Angewandte Informatik*, 3:120–127, 1978.

[12] Rolf Isernhagen. Automotive network simulator. In *ISATA Conference Report*, volume 88130, 1988.

[13] H. Kesten and J. Th. Runnenberg. Priority in waiting-line problems i and ii. *Nederlandse Akademie van Wetenschappen. Proceedings*, Series A, 60:312–324, 325–336, 1975.

[14] Leonard Kleinrock. *Queueing Systems. Computer Applications*, volume 2. John Wiley, 1976.

[15] H. Kopetz. A solution to an automotive control system benchmark. Research report, Institut für Technische Informatik. TU Wien, April 1994.

[16] Hermann Kopetz. *Real-Time Systems. Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997.

[17] I.N. Kovalenko. Rare events in queueing systems. *Queueing Systems*, 16:1–49, 1994.

[18] W. Lawrenz. „Autobusse“ für die Sensor/Aktor-Vernetzung am Beispiel CAN. *Automatisierungstechnische Praxis*, 35:450, 1993.

[19] W. Lawrenz, editor. *CAN Controller Area Network. Grundlagen und Praxis*. Hüthig Verlag, 1994.

[20] W. Lawrenz. Worldwide status of CAN – present and future. In *2nd International CAN Conference (iCC'95)*, 1995.

[21] W. Lawrenz, editor. *CAN System Engineering. From Theory to Practical Applications.* Springer Verlag New York, 1997.

[22] H. Mössenböck. *Object-Oriented Programming in Oberon-2.* Springer, 1993.

[23] H. Müller, editor. *Lexikon der Stochastik.* Akademie Verlag, 1991.

[24] N. Navet and Y.-Q. Song. Performance and fault tolerance of real-time applications distributed over CAN (Controller Area Network). *CiA Research Award 1997*, 1997.

[25] Nimal Nissanke. *Realtime Systems.* Computer Science. Prentice Hall, 1997.

[26] M Reiser and N. Wirth. *Programming in Oberon. Steps beyond Pascal and Modula-2.* Addison-Wesley, 1992.

[27] Class C application requirement considerations. SAE Technical Report J2056/1, June 1993.

[28] K. Tindell, A. Burns, and A. Wellings. Calculating Controller Area Network (CAN) Message Response Times. In *IFAC workshop on Distributed Computer Control Systems, Toledo, Spain*, September 1994.

[29] K. Tindell, A. Burns, and A.J. Wellings. Calculating Controller Area Network (CAN) Message Response Times. *Control Engineering Practice*, 3(8):1163–1169, 1995.

[30] Ken Tindell and Alan Burns. Guaranteeing Message Latencies on Control Area Network (CAN). In *1^{st} International CAN Conference (iCC'94), Mainz*, 1994.

[31] K.W. Tindell, H. Hansson, and A.J. Welling. Analysing Real-Time Communications: Controller Area Network (CAN). In *Real-Time Systems Symposium, Puerto Rico*, Dezember 1994.

[32] Z. Wang, H. Lu, G. Hedrik, and M. Stone. Message delay analysis for CAN based network. In *Proc. of 1992 ACM SIGAPP symposium on applied computing, Kansas City, MO (USA)*, March 1992.

[33] N. Wirth and J. Gutknecht. *Project Oberon. The Design of an Operating System and Compiler.* Addison-Wesley, 1992.

Company:	Institute for Distributed Systems University of Applied Sciences (FH) Braunschweig/Wolfenbüttel
Address:	Salzdahlumer Str. 46/48, D-38302 Wolfenbüttel, Germany
Phone:	+49 5331 939 649
Fax:	+49 5331 939 602
Email:	`R.Ruediger@FH-Wolfenbuettel.DE`
Homepage:	`http://www.fh-wolfenbuettel.de/fb/i/organisation/personal/ruediger/index.html`