Impact of clock drifts on CAN frame response time distributions

Aurélien Monot[1,2], Nicolas Navet[2,3], Bernard Bavoux[1]

1: PSA Peugeot Citroën, Route de Gisy, 78 943 Vélizy-Villacoublay, France
2: INRIA - Nancy Université, BP 239, 54506 Vandoeuvre, France
3: RealTime-at-Work, 615 Rue du Jardin Botanique, 54506 Vandoeuvre-lès-Nancy, France
Contact author: Nicolas Navet (Nicolas.Navet@realtimeatwork.com)

Abstract: The response time distributions of the frames sent on a Controller Area Network (CAN) bus are of prime interest to dimension and validate automotive electronic architectures. However, the existing work on the timing behaviour of the CAN network does not take into account that all the data exchanges between the Electronic Control Units (ECUs) are driven by different and independent clocks which are subject to clock drifts. This paper proposes a model for clock drifts and describes their impact on the CAN frame response time distributions. By implementing the clock drifts in a CAN simulation tool, we show experimentally that the response time distributions converge, for drift values chosen randomly within the same range on all ECUs, whatever the initial phasings between the sending nodes. Furthermore, we show that, as a result of the clock drifts, the situations leading to the worst case response times are transient.
keywords: Automotive embedded systems, design, simulation, clock drifts, response times, CAN.

1 Introduction

1.1 Context of the study.

Electronic architectures in the automotive domain are defined years in advance and their real-time properties need to be evaluated and validated before going further in the design process. For this reason, the response times of the messages transmitted over CAN have been of prime interest of research and experimentations. Therefore CAN frame response times have been studied extensively for years through the means of analysis and simulation. As worst case response times (WCRT) have been initially the main focus of these research studies in order to ensure the safety requirements, response time distributions were eventually looked into as they provide richer information on the real-time behaviour of the system. Our experiments suggest that WCRT are very rare events and quantiles such as 99,999% might be more relevant when dimensioning the electronic architecture as they can be significantly lower and might still provide enough guarantees to meet the safety requirements. being able to obtain accurate response time distribution can lead to significant cost optimization on hardware. However, in the previous works, as the response time distributions stemmed from different variability factors such as all the possible phasing configurations between the ECUs connected to the bus, another variability factor is usually overlooked : the clock drifts.

1.2 Clock drifts

Clock drifts result from the fact that the clocks of the ECUs which drive the instantiations of the CAN frames do not exactly operate at the same frequency. Due to production tolerances, the oscillators are not exactly identical and their frequency may also change over time because of environmental factors such as the temperature. In practice, the clock drifts are unavoidable and cause the phasings between the communicating ECUs to vary continuously over time. As a direct consequence, the response times of periodic frames vary continuously too. This paper describes how clock drifts are modeled in a CAN bus simulation tool [1] and describes their impact on the response time distributions of periodic CAN frames.

1.3 Previous work

The timing analysis of CAN has been rather extensively investigated in the past. Bounds on the worst case response times have been provided [2] and [3], researchers then started to integrate the limitations of hardware [4] and the communication stack, as the first analyses usually overlooked them, and also considered the effect of aperiodic traffic on CAN frame response times [5] and the consequences of transient perturbations [6]. Of course, methods to minimize the response times, or make them more predictable, were investigated too, e.g. in [7] and [8]. Some studies were addressing probabilistic analysis, and specifically response time distributions as a consequence of 1) all the possible phasing configurations between ECUs [9, 10] and 2) CAN bit-stuffing mechanism [11]. None of these works assumed the possibility of ECU clock drifts though they are a subject of interest in the field of wireless sensor networks [12]. GM researchers looked into the effects of clock drifts on the end-to-end latencies using model checking [13]. However their work is not about CAN and deriving response time distributions.

1.4 Outline of the paper

In section 2, we introduce a model of clock drifts and explain the set of working assumptions. Then, in section 3, we present the use-cases addressed and the simulation tool used for that purpose. Section 4 then describes the experimental results obtained. Finally, section 5 presents the conclusion and future work.

2 System model

2.1 Modeling clock drifts

In car electronic architectures, ECUs are typically driven by quartz oscillators. If ceramic resonators are considered for the less critical applications, they usually do not meet the requirements on maximum clock drift to drive CAN communication controller. For quartz oscillators, clock drifts result from various factors, the main of them being fabrication tolerance, aging, and temperature. Humidity and vibrations may also influence the clock drift of a quartz oscillator but are considered negligible compared to the ambient temperature and the quality of the quartz. Clock drifts are measured in “parts per million” or *ppm*, which express how slower or faster a clock is compared to a “perfect” clock. For instance, 1 ppm corresponds to a deviation of $1\mu s$ every secondTypical values provided by quartz manufacturers for an automotive usage are tolerances of +/- 50 ppm for the fabrication, +/- 5 ppm per year for aging and +/- 150 ppm for temperature in their operating temperature range (*i.e.*, -40°C/125°C). In practice, the observed values tend to be smaller, *e.g.* +/- 20 ppm over the whole temperature range.
Clock drifts can be modeled in different ways but we will choose here a rather simple and widely applicable one, based on fixed deviations of clock speeds (positive or negative) with respect to a nominal speed. For a given clock c driving a CAN node, its local time t_c with respect to a global time t is simply as follows :

$$t_c(t) = \phi_c + \delta_c \cdot t$$

where ϕ_c is the initial start time (the phasing) of the CAN node with regart to a global time referential, and δ_c is the constant drift value. For instance, a drift rate of +50ppm means that $\delta_c = 1.000050$. For this work, every CAN node j is assigned a clock defined by the tuple (ϕ_j, δ_j).
Here, the variation due to aging is assumed to be negligible for the simulation lengths that we are considering. Furthermore, we assumed that the CAN nodes are operating at a constant temperature. This later assumption is

arguable, especially for the controllers in the engine compartment. For this reason, we have additionally implemented variable clock drifts in the simulation tool, but this will not be discussed further in this paper.

2.2 Hypotheses on the CAN nodes

At each communicating node, the messages are queued in priority order and there is a sufficient number of communication buffers such that the highest priority message is always considered for arbitration. The copy time from the message queue to the communication buffers are negligible and no specific hardware limitations, other that the clock drifts considered here. A CAN message m_i is described by (T_i, O_i, C_i, P_i) where T_i is its period, O_i its initial offset, C_i its transmission time and P_i its priority. In the context of this study, the worst case of bit-stuffing is always considered for C_i. Since, the instantiations of the CAN messages are driven by their sending node clock, T_i and O_i are the values as seen by the local clock of its sending node. As a result, messages with equal periods but sent by different CAN nodes may have different frequencies with respect to the global time reference. However, C_i is the transmission time on the CAN bus which only depends of the message payload (message length) and the bit stuffing, and is expressed in the global time referential (assuming that the bus speed is perfect). Simulating this system eventually consists in emulating the CAN bus behaviour (mainly the arbitration process), in this global time referential by converting the dates of the events happening in each sending node subjective time referential into the observation absolute time referential.

3 Use-cases and toolset

Understanding the impact of clock drifts on frame response times raises numerous questions. However, some of them seemed more relevant and are now going to be described through two use-cases. In order to study these use cases, the system model described in the previous section has been implemented in a CAN simulation tool which will be also presented : RTaW-Sim [1].

3.1 Deriving frame response time distributions

The purpose of having a long observation window is to be able to observe the overall response time statistics over long scenarios (a single trip or the car estimated total functioning time, *e.g.* 7000 hours). This will allow engineers to draw conclusions about the dimensioning of the CAN bus: its speed and the message set characteristics. The statistics obtained by simulation will show what kind of performance can be expected from the studied configuration. For this use case, we also want to determine how big is the impact of the initial conditions on the response time statistics.

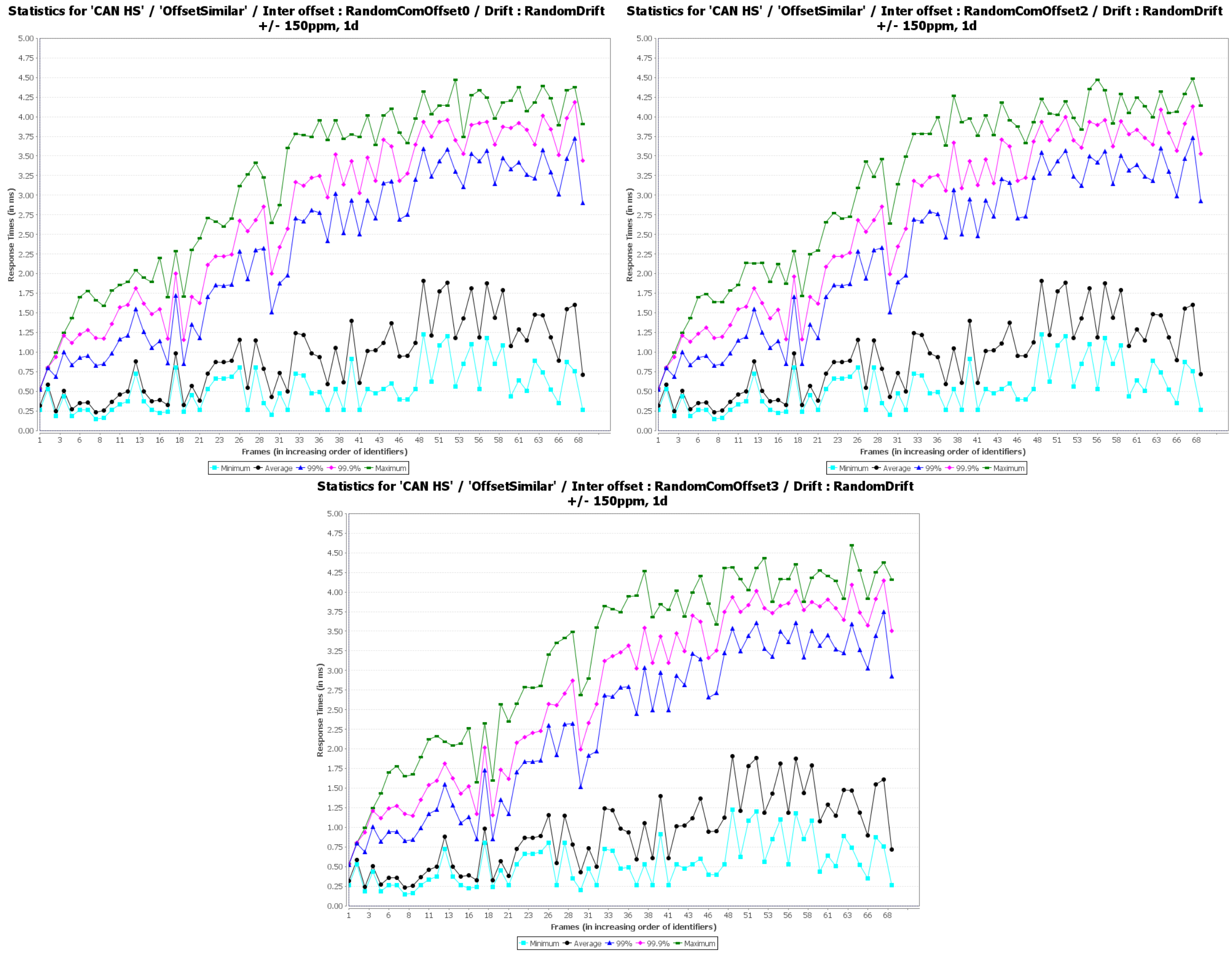


Figure 1: Distribution statistics of a CAN message set for different clock drifts configurations and initial phasing configurations. The two left graphics share initial phasing parameters and the two right graphics share the same clock drifts configurations. The X-axis indicates the frames sorted by decreasing priorities (*i.e.*, increasing IDs) while the Y-axis indicates the response times. The different curves display different response time statistics. From lowest to highest : minimum, average, 99% quantile, 99.9% quantile and maximum.

3.2 Reproducing a worst case scenario

Alternatively, the short simulation purpose is to be able to observe some specific scenarios, namely with specific phasings between node. For instance, it is possible to find by analysis the phasing conditions that provoke the worst case response time for some frame. Then, using the simulation tool, it is now possible to observe how long this situation lasts and where the clock drifts lead from there. Such simulations also contribute to validate the results obtained from the analysis tool, which is needed because these tools are black box and have to to make simplifications about the hardware and the communication stack [14].

3.3 Software toolset

The software used in this study, RTaW-Sim [1] is a fine-grained discrete-event CAN bus simulator providing the frame response time distributions. It is free for all uses and available for download. The granularity of the simulation is $1\mu s$ allowing thus to simulate accurately a CAN bus at the bit level even at the highest possible speed of 1 Mbit/s. RTaW-Sim has various other features such as a precise modeling of the communication buffers, and fault injection capabilities that will not be used here. Also, we are going to use its import feature from NETCAR-Analyzer[1] that allows to replay the phasing scenarios leading to the worst case response times[2].

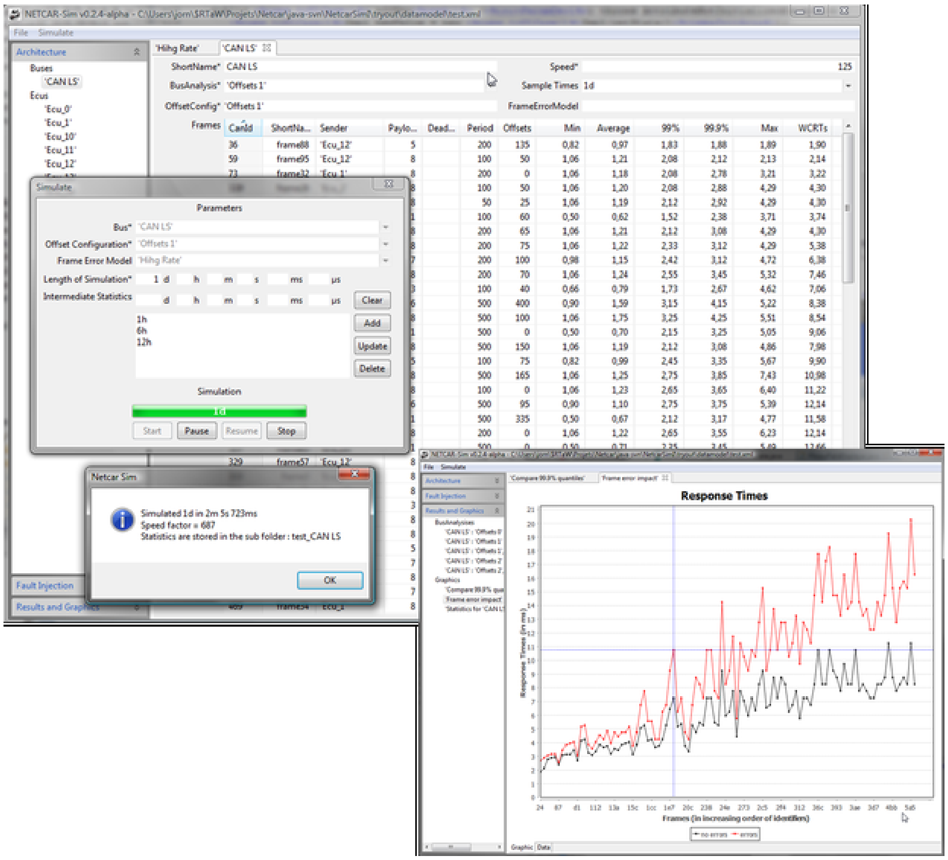


Figure 2: Screenshot of RTAW-Sim. The largest window in the underground contains the frames properties. The two smallest windows show the simulation options and its progress. The last bottom right window displays simulation statistics after a run.

4 Experimental results

In this section, we present the experimental results obtained by simulation on realistic sets of message. The simulator is able to compute about 700 kilo-events per seconds on a 2 GhZ laptop computer, which means for instance that it can simulate 24 hours of communication on a 60% loaded CAN bus in about 20 minutes. When simulating a CAN bus, the usual approach is to compute a few hyperperiods for as much phasing configurations possible, resulting often in tens of hours of simulation even using a rather coarse-grained time granularity. With clock drifts, as the phasing configuration is slowly changing over time, longer simulation times are required. Thus, the length of the simulation is an important parameter. In the following experiments, we consider two opposite lengths of observation window corresponding to the two aforementioned use cases.

[1] NETCAR-Analyzer is a worst-case response time analysis tool [15] ©INRIA/INPL/RTaW

[2] For each frame, the initial offset configuration leading to the worst case is usually different.

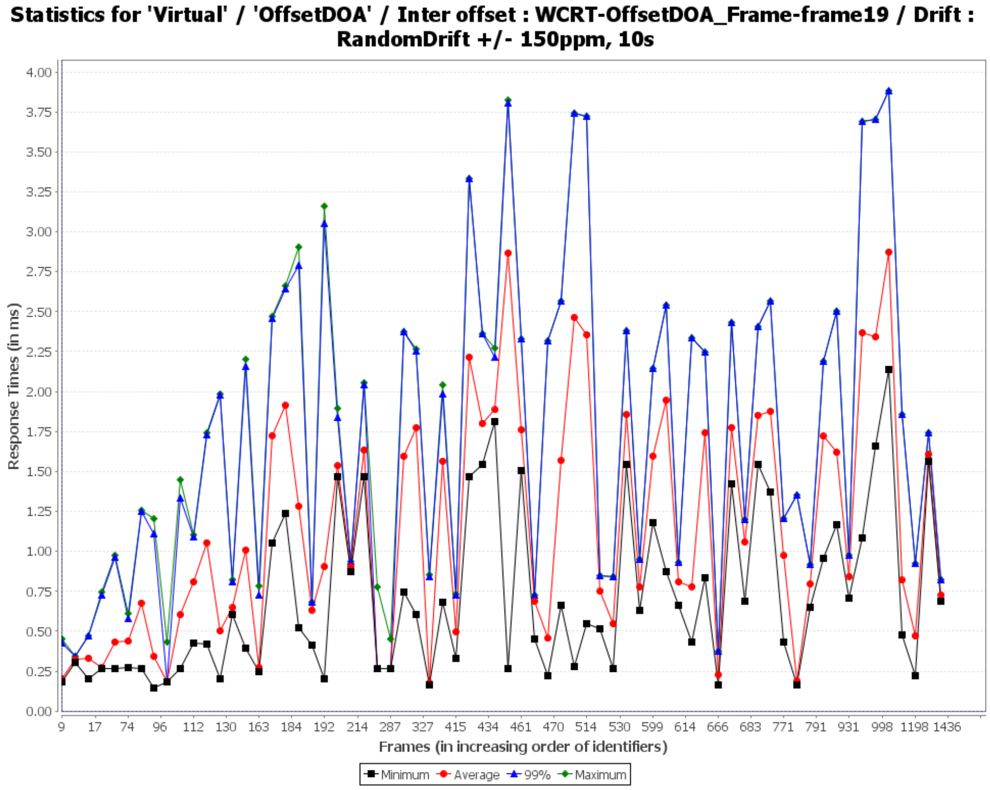
4.1 Deriving frame response time distributions

For these first experiments, we study the overall behaviour of the CAN frames response time distributions over long simulations. We provide here results for a simulated time of 24h. The message set is the one introduced in [9] which is made up of 69 CAN frames sent by 6 different nodes over a 512 Kbit/s CAN bus. The total load of this message set is 60.25%. The offsets for the frame were computed using the SOA algorithm from NETCAR-Analyzer [15].

The figure 1 shows response times statistics for 3 configurations corresponding to different clock drifts parameters and initial phasing configurations. The two left simulation results share the same initial phasing configurations for the sending nodes while the two right ones have the same clock drift parameters. The initial phasing and clock drift values were chosen randomly (clock drifts are limited to the +/- 150 ppm range). As expected, the green curves, corresponding to maximum simulated response times, are slightly different, especially for the lowest priority frames (right-side of the graphic) since the parameters differ for the different runs. The minimum response times curves are equal because the phasings between sending nodes changes in such a way that every frame will be sent without delay at least once during the 24h. What is more surprising is that the average response times, the 99% and the 99.9% quantiles are almost identical. Intermediate statistics corresponding to intermediate simulation times show that, after a while, the different curves converge. The lower the quantile, the sooner it stabilizes. This is a meaningful result because it suggests that, for a long simulation time, the clock-drift parameters and initial simulation phasings are not relevant for the final response time distributions statistics (except of course for the maximum observed response times). As a consequence, a single and long run might be enough to get robust response time distribution statistics for a given message set. Additionally, other experiments have shown that higher clock drifts help to stabilize the statistics faster which means that it can reduce the simulation times to obtain long-term CAN frame response time distributions.

4.2 Reproducing a worst case scenario

In this second experiments, we replay the configuration leading to the wort-case response time for a specific frame and examine what happens shortly after a worst case response time. In order to do that, we start a simulation with the phasing parameters leading to a specific frame (here, the frame with identifier 192) that obtain through analysis. Here, the system is composed of 84 frames sent by 12 can nodes corresponding to a load of 35% on a 512 Kbit/s CAN bus which was derived from an existing CAN set from PSA Peugeot Citroën. Simulation statistics are shown in figure 3. The top graphic shows that after 10s of simulation, high response times were reached for numerous frames as a consequence of the initial parameters and that the average response times are halfway in between the minimum and maximum response times. However, the bottom graphic shows that after 1 minute of simulation, the minimum and average response times are significantly lower. Clock drifts, which is usually seen as being detrimental because they reduce the predictability of the system, might also be beneficial in some cases: they help going out of phasing configurations leading to large response times relatively quickly, which in other words means that these worst case situations become transient.



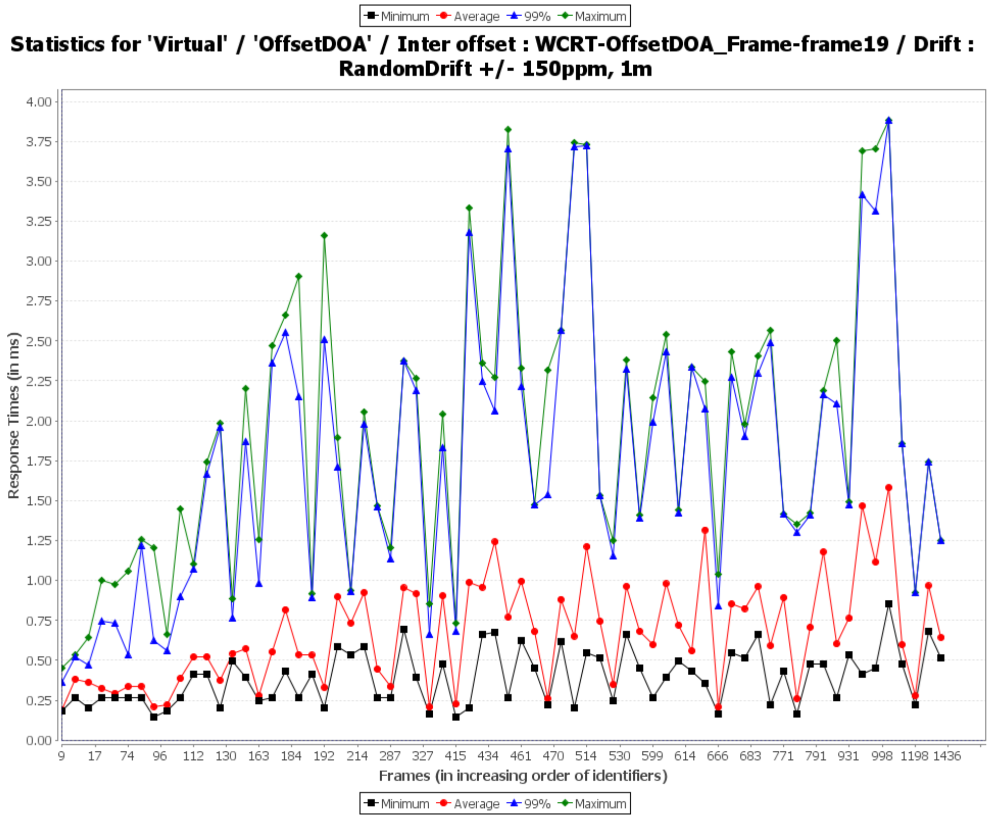


Figure 3: Simulation statistics with initial phasings leading to the worst case response time for the frame with identifier 192. The top and bottom graphics respectively show response time statistics after 10 seconds and 1 minute of simulation. Lowest to highest curves are : minimum, average, 99.9% quantile and maximum.

5 Conclusion

The aim of this paper is to understand the impact of clock drifts on CAN frame response times distributions. Here, we presented how it is possible to analyze their impact from a given initial configuration or over the overall lifetime of a vehicle through a simulation tool. We showed experimentally that the response time distributions converge whatever the initial phasings between sending nodes. This result is intriguing and deserves further research in order to understand it formally, and be able to identify the conditions that are necessary for it to hold (e.g., conditions on the clock drifts). Furthermore, we showed that, as a result of the clock drifts, the worst case response times are transient. Our ongoing work is to extend the existing CAN stochastic analyses [9] to take into account clock drifts.

References

[1] RealTime-at-Work, "RTAW-Sim: a CAN simulation tool," available at http://www.realtimeatwork.com/software/rtaw-sim/, 2009.

[2] K. Tindell, A. Burns, and A. J. Wellings, "Calculating Controller Area Network (CAN) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163 – 1169, 1995.

[3] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[4] D. Khan, R. Bril, and N. Navet, "Integrating hardware limitations in CAN schedulability analysis," in *8th IEEE International Workshop on Factory Communication Systems (WFCS2010)*, 2010, pp. 207 –210.

[5] D. Khan, N. Navet, B. Bavoux, and J. Migge, "Aperiodic traffic in response time analyses with adjustable safety level," in *IEEE Conference on Emerging Technologies Factory Automation (ETFA 2009)*, 2009.

[6] N. Navet, Y. q. Song, and F. Simonot, "Worst-case deadline failure probability in real-time applications distributed over CAN (Controller Area Network)," *J. Syst. Archit.*, vol. 46, pp. 607–617, 2000.

[7] M. Grenier, L. Havet, and N. Navet, "Pushing the limits of CAN - scheduling frames with offsets provides a major performance boost," in *European Congress of Embedded Real-Time Software (ERTS 2008)*, 2008.

[8] M. Nahas, M. J. Pont, and M. Short, "Reducing message-length variations in resource-constrained embedded systems implemented using the Controller Area Network (CAN) protocol," *J. Syst. Archit.*, vol. 55, pp. 344–354, May 2009.

[9] H. Zeng, M. D. Natale, P. Giusto, and A. L. Sangiovanni-Vincentelli, "Stochastic analysis of distributed real-time automotive systems," *IEEE Trans. Industrial Informatics*, vol. 5, no. 4, pp. 388–401, 2009.

[10] ——, "Using statistical methods to compute the probability distribution of message response time in Controller Area Network," *IEEE Trans. Industrial Informatics*, vol. 5, no. 4, pp. 678–691, 2010.

[11] T. Nolte, H. Hansson, and C. Norström, "Probabilistic worst-case response-time analysis for the Controller Area Network," in *the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '03)*, 2003.

[12] M. De Biasi, C. Snickars, K. Landernas, and A. Isaksson, "Simulation of process control with WirelessHART networks subject to clock drift," in *32nd Annual IEEE International Computer Software and Applications (COMPSAC '08)*, 2008, pp. 1355 –1360.

[13] P. K. Pandya and P. Vijay Suman, "End-to-end message latency under clock drift in Uppaal," GM, Tech. Rep., 2008.

[14] N. Navet, A. Monot, and J. Migge, “Frame latency evaluation: when simulation and analysis alone are not enough,” 8th IEEE International Workshop on Factory Communication Systems (WFCS2010), Industry Day, 2010.

[15] RealTime-at-Work, “NETCAR-Analyzer: a CAN analysis tool,” available at http://www.realtimeatwork.com/software/netcar-analyzer/, 2009.