Sufficient Real-Time Analysis for an Engine Control Unit with Constant Angular Velocities

Victor Pollex*, Timo Feld*, Frank Slomka*, Ulrich Margull†, Ralph Mader‡, Gerhard Wirrer‡
*Institute of Embedded Systems/Real-Time Systeme
Ulm University, 89069 Ulm, Germany
firstname.lastname@uni-ulm.de
†Faculty of Electrical Engineering and Computer Science
Ingolstadt University of Applied Sciences
Esplanade 10, 85049 Ingolstadt, Germany
firstname.lastname@haw-ingolstadt.de
‡Continental AG
Siemensstr. 12, 93055 Regensburg, Germany
firstname.lastname@continental-corporation.com

***Abstract*—Engine control units in the automotive industry are particular challenging real-time systems regarding their real-time analysis. Some of the tasks of such an engine control unit are triggered by the engine, i.e. the faster the angular velocity of the engine, the more frequent the tasks are executed. Furthermore, the execution time of a task may vary with the angular velocity of the engine. As a result the worst case does not necessarily occur when all tasks are activated simultaneously. Hence this behavior cannot be addressed appropriately with the currently available real-time analysis methods. In this paper we make a first step towards a real-time analysis for an engine control unit. We present a sufficient real-time analysis assuming that the angular velocity of the engine is arbitrary but fixed.**

I. Introduction

Embedded systems are increasingly used in electronic devices. Especially in the automotive area these are deployed on a large scale. The addition of applications, like driver assistance systems, increases the demand of resources, but cost pressure dictates to refrain from unnecessarily increasing the number of embedded systems. Therefore the available resources have to be utilized efficiently. Furthermore many of these systems are hard real-time systems. This real-time capability can be verified by means of a real-time analysis.

An engine control unit (ECU) is a hard real-time system on which many parameters are calculated, including the quantity that is injected and the timing of the ignition. These calculations increase the performance of the engine, however if a calculation does not finish on time, then the performance of the engine degrades. Therefore the real-time capability of the ECU must be verified.

The duration of the work cycle of a piston correlates with the speed of the engine. Therefore the frequency with which the ECU calculates the timing of the ignition correlates equally with the speed of the engine. Furthermore the corresponding deadline as well as the execution time can also depend on the speed of the engine.

In this paper we present a sufficient real-time analysis that takes the aforementioned properties of an ECU into account. This analysis assumes that the speed of the engine is arbitrary but fixed, which on the one hand reduces the complexity of the analysis. On the other hand however, the analysis can only be regarded as a first step.

This paper is organized as follows: In section II a short overview of the related work is given, followed by the computational model used in this paper which is described in section III. In section IV we present the real-time analysis for the ECU, which is then applied on an existing ECU provided by Continental. The ECU and the results are shown in section V. A summary and future work are given in section VI.

II. Related Work

In the past fifteen years various task models have been researched. Among them are the multiframe task model [1], its generalized form [2], the recurring real-time task model [3], and the currently most general model, the digraph task model [4]. The digraph task model describes a task with a directed graph. Each node represents a type of job the task can release and has a execution time and deadline assigned to it. An edge is labeled with the minimum time between the activation of the jobs that the edge connects.

A task whose frequency of activation depends on the engine speed, like calculating the ignition timing, cannot be properly modeled by the digraph task model. For a single engine speed however it would be possible to model such a task. The range of the engine speed is limited, but the number of different engine speeds is infinite. Even though we assume an arbitrary but fixed engine speed for our analysis, we would require an infinite amount of graphs to correctly model and engine-speed dependent task. The reason is that the ECU has to be real-time capable at all possible engine speeds.

Stoimenov et al. present in [5] a real-time analysis for systems that can change their mode of operation. Each mode is represented by a set of tasks, where each task is associated with an activation pattern, a best-case and worst-case execution

978-3-9815370-0-0/DATE13/©2013 EDAA

time, and a deadline. Even though a task whose execution time depends on the engine speed can only have a finite amount of different execution times, once the deadline or its activation depends on the engine speed, again an infinite amount of modes would be required to appropriately model the behavior.

III. Computational Model

In this section we present the model used for the analysis of the ECU. The ECU executes different types of tasks. These tasks are event-triggered. Whenever a certain event occurs, an instance of the corresponding task is created and made ready to be executed. Hereafter this will be referred to as activation of a task. The tasks are coarsely categorized based on the source of the event. On the one hand we have time-triggered tasks. The events of these tasks occur after a certain amount of time has elapsed. On the other hand we have engine-triggered tasks. Here the events occur when the engine has reached a certain angular position. As the engine revolves, this angular position is reached with every revolution, hence these events recur. Furthermore not only the occurrences of the event of the engine-triggered tasks depend on the angular velocity of the engine, but also some parameters of the tasks themselves.

A. Engine

The engine is a four-cycle engine. Each cycle requires half a revolution, hence the complete work cycle of four cycles requires two revolutions. We denote the number of cylinders of the engine with ζ. The engine can only operate within a fixed range of angular velocities. The lower and upper bound of this range are denoted by ω^- and ω^+ respectively.

B. Tasks

A task τ is described by a 5-tuple

$$\tau := (\pi, \rho, \eta, \gamma, \delta) \tag{1}$$

the priority π, the scheduling type ρ, the event function η, the execution time function γ, and the deadline function δ. A set of tasks is denoted by Γ.

Every task τ has a priority π_τ assigned which is used in the priority-based scheduling policy of the ECU. If tasks have identical priorities then the instances of these tasks are processed according to the first come first serve policy. The scheduling type ρ of a task can either be fully preemptive or deferred preemptive. A fully preemptive task can preempt at any given time any other task with a priority lower than its own and the task can be preempted at any given time by a task with a higher priority than its own. Whereas the execution of a deferred preemptive task is divided into a series of non-preemptive segments $s_{\tau,i,\omega}$ and hence can only be preempted between two non-preemptive segments. $s_{\tau,i,\omega}$ denotes the length of the i-th non-preemptive segment of task τ for the angular velocity ω.

As previously mentioned, the tasks executed on the ECU are event-triggered and the events that activate the tasks occur repeatedly. With the event function $\eta_{\tau,\omega}$ we describe an upper bound on the number of events activating task τ that can occur in any interval of time of non-negative length when the angular velocity is ω.

$$\eta_{\tau,\omega} : \mathbb{R}_0^+ \to \mathbb{N}_0 \tag{2}$$

We also mentioned that some parameters of the tasks depend on the angular velocity of the engine. In particular these are the execution time and the relative deadline of the task. Hence both are described as a function of the angular velocity:

$$\gamma_\tau : [\omega^-, \omega^+] \to \mathbb{R}^+ \tag{3}$$

$$\delta_\tau : [\omega^-, \omega^+] \to \mathbb{R}^+ \tag{4}$$

The execution time function $\gamma_\tau(\omega)$ of a task τ describes the required amount of processing time for its execution if the angular velocity was ω at the time of the activation. Whereas the deadline function $\delta_\tau(\omega)$ describes the amount of time that a task may require at most from the time the corresponding event activated it until it finishes its execution.

The reason that the execution time varies over the angular velocity of the engine is that for different ranges of angular velocities different algorithms are used. On the one hand this is done to consider the reduced amount of time available at higher angular velocities. On the other hand parts of calculations done at lower angular velocities are omitted at higher angular velocities. For example at lower angular velocities the fuel is injected with several pulses. At higher angular velocities the number of pulses is reduced due to technical limitations. Hence part of the calculations are omitted as they are not required. However in general it cannot be assumed that with increasing angular velocity the execution time decreases.

Time-triggered tasks are repeatedly activated after a specified amount of time has elapsed. Hereafter referred to as period of a task τ and denoted by p_τ. Furthermore the relative deadline of time-triggered tasks are considered to be arbitrary but constant, i.e. they do not depend on the angular velocity of the engine.

C. Interrupts

Besides the execution of tasks on the ECU, also interrupts have to be processed. Interrupts can preempt any currently running task on the ECU. To take the interrupts in the analysis into account we require a function

$$\iota : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \tag{5}$$

which describes the maximum amount of processing time required to serve the interrupts that occur within any interval of time of a given length. How to obtain such a function is described more in detail in section V.

IV. Real-time analysis

The analysis we present is based on the response-time analysis of Lehoczky [6]. As the tasks on the ECU are either fully preemptive or deferred preemptive, we combine the analysis of Lehoczky with the approach presented by Haid and Thiele [7] and part of the analysis presented by Bril et al. [8].

The condition for schedulability is modified to:

$$\forall \tau \in \Gamma \; \forall \omega \in [\omega^-, \omega^+] : r^+_{\tau,\omega} \leq \delta_\tau(\omega) \tag{6}$$

If the worst-case response time $r^+_{\tau,\omega}$ of each task τ in the set of tasks Γ is no larger than the corresponding deadline $\delta_\tau(\omega)$ for each possible angular velocity ω between the minimum ω^- and the maximum ω^+, then the system is schedulable.

$$r^+_{\tau,\omega} = \max_{k\in[1,m_\omega]} \{\beta_{\tau,\omega}(k) - \alpha_{\tau,\omega}(k)\} \tag{7}$$

$$m_\omega = \min\{k \in \mathbb{N} : \beta_{\tau,\omega}(k) \leq \alpha_{\tau,\omega}(k+1)\} \tag{8}$$

The worst-case response time $r^+_{\tau,\omega}$ is the maximum of any response time of an instance in a level-i busy period. The level-i busy period ends with the m-th instance which is the first instance that finishes its execution no later than the activation of its succeeding instance. The response time of the k-th instance is the difference of the time when its execution finishes $\beta_{\tau,\omega}(k)$ and the time it was activated $\alpha_{\tau,\omega}(k)$.

$$\beta_{\tau,\omega}(k) = \min_{\Delta\in\mathbb{R}^+} \{\Delta : \Delta = f_{\tau,k,\omega}(\Delta)\} \tag{9}$$

$$\alpha_{\tau,\omega}(k) = \inf_{\Delta\in\mathbb{R}^+_0} \{\Delta : k \leq \eta_{\tau,\omega}(\Delta)\} \tag{10}$$

$\beta_{\tau,\omega}(k)$ is the smallest positive fix-point of the function $f_{\tau,k,\omega}$.

$$f_{\tau,k,\omega}(\Delta) = b_{\tau,\omega} + k\cdot\gamma_\tau(\omega) + \iota(\Delta) + \sum_{\tau'\in hep(\tau)} (\eta_{\tau',\omega}(\Delta)\cdot\gamma_{\tau'}(\omega)) \tag{11}$$

$f_{\tau,k,\omega}$ describes the maximum amount of execution time that has been requested within an interval of time of length Δ. This includes the blocking time $b_{\tau,\omega}$ due to the deferred preemptable tasks, the execution time of all k instances of task τ, the processing of the interrupts ι, and the interference of all higher or equal priority tasks $hep(\tau)$. This is done to take the possibility into account that different tasks can have the same priority.

The blocking time of a fully preemptive task is zero as it can preempt any task with a lower priority than itself. For a deferred preemptive task it is the maximum length of a non-preemptive segment of any task τ which has a lower priority than τ.

$$b_{\tau,\omega} = \begin{cases} 0 & \text{if } \tau \text{ is fully preemptive} \\ \max\limits_{\tau'\in lp(\tau)} \{s_{\tau',i,\omega}\} & \text{if } \tau \text{ is deferred preemptive} \end{cases} \tag{12}$$

The time when the k-th instance is activated $\alpha_{\tau,\omega}(k)$ is the pseudo-inverse of the event function. It is the smallest length of an interval in which k events can occur.

The event function is defined for every type of task executed on the ECU. As a time-triggered task is not affected by the angular velocity of the engine, the event function is:

$$\eta_{\tau,\omega}(\Delta) = \left\lceil \frac{\Delta}{p_\tau} \right\rceil \tag{13}$$

However, for engine-triggered tasks the angular velocity of the engine does affect the number of activations. The crankshaft task is activated once per revolution. Given the angular velocity ω in revolutions per time unit and a length of an interval Δ in time units it follows that the event function is:

$$\eta_{\tau,\omega}(\Delta) = \lceil \omega\cdot\Delta \rceil \tag{14}$$

In the segment task the majority of the calculations for the fuel injection and ignition timing is performed, therefore it can be activated several times per revolution. The actual amount depends on the number of cylinders used in the engine. The segment task is activated once per work cycle of a piston. As the engine is a four-cycle engine, each piston in a cylinder has a work cycle of two revolutions. The start of a work cycle of a piston is distributed equiangular over the work cycle of the engine. For example for a four cylinder engine the segment task is activated every 180 degrees, hence for an engine with ζ cylinders the event function is:

$$\eta_{\tau,\omega}(\Delta) = \left\lceil \frac{1}{2}\cdot\zeta\cdot\omega\cdot\Delta \right\rceil \tag{15}$$

The camshaft task is similar to the segment task as it also has a work cycle of two revolutions. But unlike the segment task the activations of the camshaft task are not distributed equiangular over the work cycle. Instead the camshaft task is activated at arbitrary but fixed angular positions within the work cycle. It is used for adjustable camshafts to regulate the adjustment of the opening and closing time of the inlet and outlet valve. This however is strongly dependent on the engine geometry, thus the asymmetric distribution of the events over the work cycle. Given the sequence of these angular positions, a function $\lambda : \mathbb{R}^+_0 \to \mathbb{R}^+_0$ can be obtained by means of a deconvolution in the min-plus algebra. Details on the min-plus deconvolution are found in [9]. λ describes the maximum amount of events than can occur within a given number of revolutions. With that the event function is:

$$\eta_{\tau,\omega}(\Delta) = \lambda(\omega\cdot\Delta) \tag{16}$$

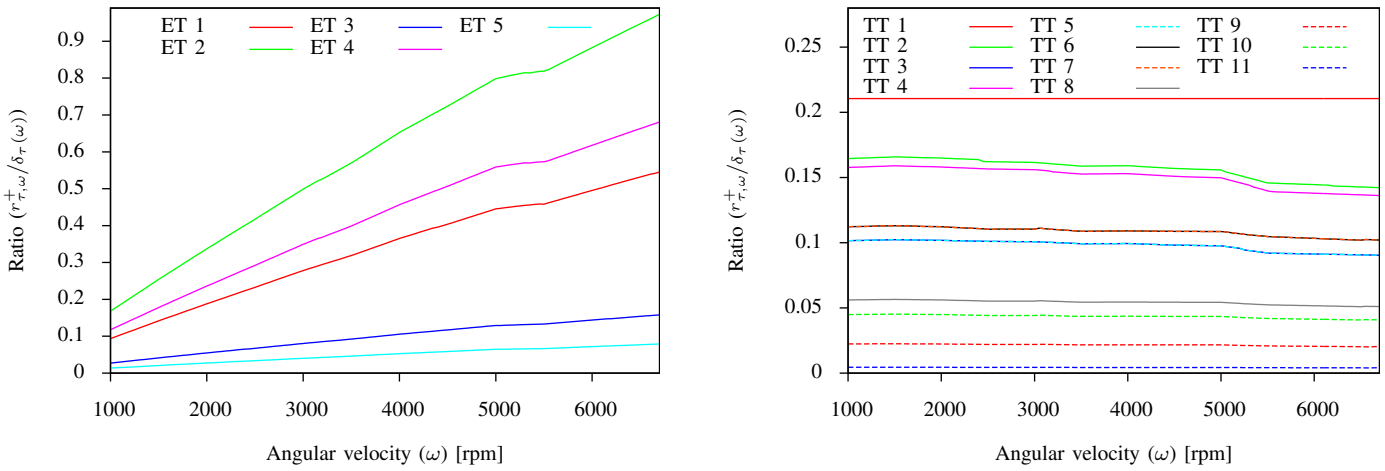
Table I: Tasks executed on the ECU

Task	$\delta(\omega^+)$	Task	Type	$\delta(\omega^+)$
TT 1	0.2 ms	ET 1	seg	1.3 ms
TT 2	5.0 ms	ET 2	seg	0.7 ms
TT 3	10.0 ms	ET 3	seg	4.5 ms
TT 4	5.0 ms	ET 4	cam	1.0 ms
TT 5	10.0 ms	ET 5	crk	9.0 ms
TT 6	20.0 ms			
TT 7	20.0 ms			
TT 8	40.0 ms			
TT 9	100.0 ms			
TT 10	50.0 ms			
TT 11	500.0 ms			

V. Experiments

We applied the analysis presented in section IV on a ECU in order to determine its schedulability. The ECU is that of a gasoline engine car and its tasks and their parameters are shown in table I. On the ECU 11 time-triggered tasks and 5 engine-triggered tasks are executed. 3 of the 5 engine-triggered tasks are segment tasks. The other two are each a camshaft task and a crankshaft task. All tasks are deferred preemptable except for the first time-triggered task (TT 1). This one is fully preemptive. The column of the table denoted with $\delta(\omega^+)$ contains the value of each deadline function at the maximum angular velocity. For time-triggered tasks the deadline function is constant, thus it does not change with the angular velocity of the engine. However, for engine-triggered tasks it does change and the deadline function is defined as follows:

$$d_\tau(\omega) = \frac{\omega^+}{\omega}\cdot d_\tau(\omega^+) \tag{17}$$



(a) Engine-triggered tasks

(b) Time-triggered tasks

Figure 1: Ratio worst-case response times to deadlines of the tasks executed on the ECU

The execution times of every task executed on the ECU was measured for every 500 rpm starting at 1000 rpm up to 6000 rpm and at 6700 rpm.

To obtain the function ι to consider the processing of the interrupts several steps were performed. First the amount of time that the processing of the interrupts required was measured over the course of 30 ms. These values are then transformed into an accumulative function. Given a point in time relative to the start of the measurement, this accumulative function describes the sum of the processing time that was required by all the interrupts that occurred up to the given point in time. By means of a deconvolution in the min-plus algebra with itself ι is obtained. Details on the min-plus deconvolution are found in [9].

The results of the analysis for this ECU are shown in Fig. 1. On the x-axis we have the angular velocity ranging from 1000 rpm up to 6700 rpm. On the y-axis we have the ratio worst-case response time to deadline for the corresponding angular velocity. The schedulability condition (6) can be restated as follows:

$$r^+_{\tau,\omega} \leq \delta_\tau(\omega) \quad \Leftrightarrow \quad \frac{r^+_{\tau,\omega}}{\delta_\tau(\omega)} \leq 1 \tag{18}$$

Therefore a ratio no greater than 1 means that the ECU is real-time capable for that specific angular velocity. However if the ratio for an angular velocity is greater than 1 then no statement can be made regarding the real-time capability of the ECU, because the condition used in the analysis is sufficient. The results show that the ratio for the engine-triggered task is clearly affected by the angular velocity. Whereas the ratio for the time-triggered tasks remains roughly constant.

VI. Summary and Future Work

In this paper we presented a sufficient real-time analysis for an engine control unit. On the engine control unit tasks are executed which are either triggered by a timer or by the engine itself. As a first step we assumed arbitrary but fixed angular velocities of the engine. The analysis was applied on an existing ECU provided by Continental.

For future work we intend to extend the presented analysis to take offsets between tasks into account. As well as dropping the assumption of arbitrary but fixed angular velocities.

References

[1] A. K. Mok and D. Chen, "A Multiframe Model for Real-Time Tasks," *IEEE Transactions on Software Engineering*, vol. 23, no. 10, pp. 635–645, October 1997.

[2] S. Baruah, D. Chen, S. Gorinsky, and A. Mok, "Generalized Multiframe Tasks," *Real-Time Systems*, vol. 17, pp. 5–22, 1999.

[3] S. K. Baruah, "Dynamic- and static-priority scheduling of recurring real-time tasks," *Real-Time Systems*, vol. 24, pp. 93–128, 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1021711220939

[4] M. Stigge, P. Ekberg, N. Guan, and W. Yi, "The Digraph Real-Time Task Model," in *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 11-14 2011, pp. 71–80.

[5] N. Stoimenov, S. Perathoner, and L. Thiele, "Reliable Mode Changes in Real-Time Systems with Fixed Priority or EDF Scheduling," in *Proceedings of Design, Automation and Test in Europe, 2009 (DATE 09)*. Nice, France: IEEE, 2009, pp. 99–104.

[6] J. P. Lehoczky, "Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines," in *Proceedings of the 11th Real-Time Systems Symposium (RTSS)*, Dec 1990, pp. 201–209.

[7] W. Haid and L. Thiele, "Complex Task Activation Schemes in System Level Performance Analysis," in *CODES+ISSS*, S. Ha, K. Choi, N. Dutt, and J. Teich, Eds. New York, NY, USA: ACM, September 30 - October 5 2007, pp. 173–178.

[8] R. Bril, J. Lukkien, and W. Verhaegh, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption," *Real-Time Systems*, vol. 42, pp. 63–119, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11241-009-9071-z

[9] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer-Verlag, 2001, vol. 2050.