Parametric Utilization Bounds for Implicit-Deadline Periodic Tasks in Automotive Systems

Georg von der Brüggen[1], Niklas Ueter[1], Jian-Jia Chen[1], and Matthias Freier[2]
[1]TU Dortmund University, Germany
[2]Robert Bosch GmbH, Germany

Abstract

Fixed-priority scheduling has been widely used in safety-critical applications. This paper explores the parametric utilization bounds for implicit-deadline periodic tasks in automotive uniprocessor systems, where the period of a task is either 1, 2, 5, 10, 20, 50, 100, 200, or 1000 milliseconds. We prove a parametric utilization bound of $90\%+z$ for such automotive task systems under rate-monotonic preemptive scheduling (RM-P), where z is a parameter defined by the input task set with $0 \leq z \leq 10\%$. Moreover, we explain how to perform an exact schedulability test for an automotive task set under RM-P by validating only three conditions. Furthermore, we extend our analyses to rate-monotonic non-preemptive scheduling (RM-NP). We show that very reasonable utilization values can still be achieved under RM-NP if the execution time of all tasks is below 1 millisecond. The analyses presented here are compatible with angle-synchronous tasks by applying the related arrival curves. It is shown in the evaluations that scheduling those angle-synchronous tasks according to their minimum inter-arrival time instead of assigning them to the highest priority can drastically increase the acceptance ratio in some settings.

1 Introduction

Embedded real-time computing systems for safety-critical applications have to satisfy the timing requirements to ensure the timeliness of a result in addition to the functional correctness. The sporadic task model [24] is the most basic task model in real-time systems, where each task τ_i releases an infinity number of *task instances* (*jobs*) under its *minimum inter-arrival time* (*period*) T_i and is further characterized by its *relative deadline* D_i and its *worst-case execution time* C_i. The sporadic task model is a generalization of the periodic task model used in the seminal work by Liu and Layland [23], in which a task releases its jobs exactly periodically. We consider *implicit-deadline* task systems, i.e, $T_i = D_i$ for all tasks.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

RTNS '17, October 4–6, 2017, Grenoble, France,

© 2017 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5286-4/17/10…$15.00
DOI: https://doi.org/10.1145/3139258.3139273

Researchers have devoted significant efforts to efficiently analyze whether a set of sporadic tasks can meet their deadlines under different scheduling strategies, platform models, and assumptions. Liu and Layland [23] presented the seminal utilization bound $\ln 2 \approx 69.3\%$ for rate-monotonic preemptive scheduling (RM-P) of periodic tasks. Lehoczky et al. [22] later provided a stochastic analysis, showing that in the average case RM-P can schedule task sets with a much higher total utilization, i.e., the average breakdown utilization is 88%. Bini [4] showed that the optimality degree is higher (over 90%) when the task utilization is uniformly distributed.

The above results for RM-P consider arbitrary configurations of task periods. If those configurations are not arbitrary, the analysis and utilization bounds should consider those parameters, as discussed by Chen et al. [9]. There are more precise utilization bounds that consider the ratios of task periods. Kuo and Mok [19] showed the utilization bound of 100% for harmonic task sets under RM-P, i.e., T_i is an integer multiple of T_j if $T_i \geq T_j$ for any two tasks in the task set, and explained how to improve the utilization bound of a non-harmonic task set by finding harmonic subsets. The harmonic relation of task periods was further exploited by Han and Tyan [15], Kuo et al. [18], and Nasri et al. [26]. The utilization bound in [15] analytically dominates those by Liu and Layland [23] and Burchard et al. [6]. Lauzac et al. [20] proposed a utilization bound of $\ln r + 2/r - 1$ based on the ratio r of the maximum to the minimum task period if $1 \leq r \leq 2$. It is $\ln 2$ if $r = 2$, i.e., the same as the Liu and Layland bound. Bini and Buttazzo [2] presented the hyperbolic bound $\prod_{\tau_i \in \tau}(1 + U_i) \leq 2$. Chen et al. [8] developed a utilization-based analysis framework that can provide hyperbolic bounds almost automatically. Lee et al. [21] presented linear programming formulations for calculating total utilization bounds when a task can choose its own period.

An alternative to RM-P is fixed-priority non-preemptive scheduling (FP-NP). As tasks are never preempted during their execution when non-preemptive scheduling is used, this results in a smaller number of context switches and therefore in total in a lower context switch overhead. Non-preemptive scheduling may also be enforced due to the hardware used in the system, e.g., control area network (CAN) buses [1]. The utilization bound for non-preemptive scheduling drops to 0 [25], since a low-priority task with low-utilization can have a very long period while its execution time is already longer than the shortest period among the tasks in the system. However, non-preemptive scheduling can still be applicable if the execution times of the tasks are short enough. For quantitive comparisons between preemptive and non-preemptive scheduling strategies based on resource augmentation factors, the recent results can be found in [13, 30]. Moreover, von der Brüggen et al. [31] and Andersson and Tovar [1] presented utilization-based analyses by incorporating the ratio of the blocking time, due to non-preemptive scheduling, to the execution time of a task.

The results mentioned above (except [21]) focused on the worst-case utilization bound among *infinitely* possible configurations of the periods of the tasks. However, in typical automotive systems, where periodic task systems are applied, only a few possible periods are *on the shelf*, usually $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ ms, see for example [14, 17, 27, 29]. Therefore, the most important settings in automotive system design are task systems that have only very limited possible periods instead of investigating all possible configurations of the periods. Although the utilization bound of $\ln 2$ for RM-P is sound, the utilization bound for task sets with periods in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ ms can be much higher. A simple combination of the results in [3, 21] already leads to a utilization bound of 90%, presented in Lemma 3.2 for completeness. Note that the focus of this paper is to further improve this bound by considering the parameters of the task set. Moreover, if the execution time of any task is always significantly shorter than 1 ms, applying fixed-priority non-preemptive scheduling is still a meaningful strategy if the achievable utilization is still high enough.

Contributions: In this paper, we explore the parametric utilization bounds of such automotive task systems in uniprocessor systems from both theoretical and practical perspectives. The contributions of this paper are as follows:

- We show that the parametric utilization bound of automotive task systems under RM-P is $90\% + z$ in Section 4.1, where z (with $0 \leq z \leq 10\%$) is a parameter defined by the input task set, detailed in Section 4.1. An efficient exact schedulability test for RM-P that tests only three conditions is provided in Section 4.2.
- Our analysis is extended to consider non-preemptive scheduling in Section 5 and angle-synchronous tasks in Section 6.
- Our evaluations in Section 7 are based on synthetic automotive task benchmarks [17], published by Bosch in 2015, which provide tasks with typical automotive characteristics [14, 27, 29]. Without angle-synchronous tasks, we show that the achievable utilization deemed schedulable for RM-P is nearly 100%, and that very reasonable utilization values can be achieved under RM-NP when the non-preemptive blocking time is small, compared to the minimum period of 1 ms. If the task set contains angle-synchronous tasks, we show that the acceptance ratio increases if the priority of the angle-synchronous tasks is assigned according to their minimum inter-arrival time instead of assigning them to the highest priority.

2 System Model

We are given a set $\mathbf{T}$ of n periodic (or sporadic) implicit-deadline real-time tasks. Each of these tasks τ_i releases an infinity number of task instances, called jobs, and is described by its inter arrival time or period T_i, its relative deadline D_i and its worst case execution time (WCET) C_i. This means, that a job of τ_i released at time t has to finish C_i computation units before $t + D_i$. The next job of τ_i is released at $t + T_i$ or not before $t + T_i$ for the period and sporadic task model, receptively. If $D_i = T_i$ for all tasks the task set has implicit deadlines; if $D_i \leq T_i$ for all tasks the task set has constrained deadlines; the task set has arbitrary deadlines if $D_i > T_i$ for some tasks. The period T_i of each task τ_i in $\mathbf{T}$ is one of the possible periods in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ milliseconds, which is common for task sets in automotive applications [14, 17, 27, 29]. For the simplicity of presentation, the unit of time is assumed to be *milliseconds* (ms) if not stated differently. Let set $\mathbf{T}_x$ be the subset of the tasks in $\mathbf{T}$ where all tasks have period x, i.e., $\mathbf{T}_x = \{\tau_i \mid \tau_i \in \mathbf{T} \text{ and } T_i = x\}$. We assume that $C_i > 0$ for each task τ_i. The utilization U_i of task τ_i is C_i/T_i. We denote such a task set $\mathbf{T}$ as an *automotive (implicit-deadline) periodic task set* in this paper. *In automotive systems, the entity for scheduling is a Runnable [17], which is equivalent to a task in this paper.*

We consider both fixed-priority preemptive scheduling (FP-P) and fixed-priority non-preemptive scheduling (FP-NP) for uniprocessor systems in this paper, as fixed-priority scheduling is widely used in practice. A fixed-priority scheduling policy assigns the same priority to every job of a task and at each point in time the scheduler executes the job with the highest priority currently in the system. Under FP-P, rate-monotonic (RM-P) priority assignment is optimal [23] for implicit-deadline task sets. Under FP-NP, we explore RM-NP, which has been proved to have a resource augmentation bound of 1.76322 against the optimal workload-conserving[1] non-preemptive scheduling in [30, 31]. By using RM-P and RM-NP, we know that all tasks in $\mathbf{T}_x$ have higher priorities than all tasks in $\mathbf{T}_y$ if $x < y$. We define $hp(\tau_k)$ as the set of the periodic tasks with priorities higher than task τ_k.

A schedule is *feasible* if all the temporal characteristics and timing constraints are respected and satisfied. A task set is *schedulable* by a scheduling algorithm if the resulting schedule is always feasible. A *schedulability test* for to a scheduling algorithm validates whether a given task set is schedulable by the scheduling algorithm. A schedulability test is referred to as *sufficient* if all the task sets it deems schedulable are in fact schedulable. A schedulability test is referred to as *necessary* if all the task sets it deems unschedulable are in fact unschedulable. Schedulability tests that are both sufficient and necessary are referred to as *exact*. A task set (under a specific task model) is always schedulable by a scheduling algorithm (or can always pass the sufficient schedulability

[1] A workload conserving scheduling algorithm is an algorithm that never lets the processor run idle if any task instance is ready to be executed.

test) if the utilization of the task set, i.e., the sum of the individual task utilizations, is no more than the *utilization bound* of the scheduling algorithm. Therefore, the utilization bound provides a quick schedulability test and also a significant evidence of the resource usages to the system designers. From the economical perspectives, nearly 100% utilization should be reached so that there is no resource loss. From the schedulability perspectives, boosting the utilization to be close to 100% may result in deadline misses. Therefore, although the utilization bound is not an exact schedulability test, it is still an important/simple metric for the system designers.

3 Preliminary Results

3.1 Preemptive Uniprocessor Schedulers

Liu and Layland showed that the schedulability of an implicit-deadline task τ_k under FP-P on a uniprocessor can be verified by considering the worst-case release pattern, termed *the critical instant*, which is to release the first jobs of the tasks in $hp(\tau_k)$ together with task τ_k and release the subsequent jobs of the tasks in $hp(\tau_k)$ by strictly following their periods [23], i.e., as early as possible. The critical instant theorem results in the time-demand analysis (TDA) [22], i.e., a task τ_k is schedulable under FP-P scheduling if and only if

$$\exists t | 0 < t \leq D_k = T_k, \quad C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t}{T_i} \right\rceil C_i \leq t \tag{1}$$

The following lemma shows that a task with a period 1, 2, 10, 20, 100, 200, or 1000 can miss its deadline if and only if the task set has more than 100% utilization.

Lemma 3.1 (Harmonic Subset). *In an automotive implicit-deadline task set, a task τ_k with $T_k \in \{1, 2, 10, 20, 100, 200, 1000\}$ is schedulable under RM-P scheduling if and only if*

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i \leq 1 \tag{2}$$

Proof. The only-if part is obvious. The if-part has been recently proved by Nasri et al. [26] and is sketched for completeness. By the exact test in Eq. (1), we only consider to test at time $t = T_k$. By definition, T_k is either one of $\{1, 2, 10, 20, 100, 200, 1000\}$ and for a higher-priority task τ_i the period T_i is either one of $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ with $T_i \leq T_k$. Therefore, T_k is an integer multiple of T_i for any task τ_i in $hp(\tau_k)$ when x is one of $\{1, 2, 10, 20, 100, 200, 1000\}$. Hence,

$$C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{T_k}{T_i} \right\rceil C_i = C_k + \sum_{\tau_i \in hp(\tau_k)} \frac{T_k}{T_i} C_i$$
$$= T_k (U_k + \sum_{\tau_i \in hp(\tau_k)} U_i) \leq T_k$$

where the inequality is due to the if-condition. □

Therefore, for an automotive implicit-deadline periodic task set, if a task misses its deadline under RM-P when the utilization of the task set is ≤ 100%, the period of the task must be either 5 ms or 50 ms. The following lemma is based on a simple combination of the results in [3, 21].

Lemma 3.2 (Utilization-Bound-Non-Harmonic Subset). *In an automotive implicit-deadline task set, a task τ_k with $T_k \in \{5, 50\}$ is schedulable under RM-P scheduling if*

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i \leq 0.9 \tag{3}$$

Proof. This follows directly from [21]. We only sketch the proof. Suppose that $Y_\ell = \sum_{\tau_i \in \mathrm{T}_\ell} U_i$ for $\ell = 1, 2, 5, 10, 20, 50$, i.e., Y_ℓ is the total utilization of the tasks with periods equal to ℓ. We prove this lemma only for $T_k = 5$. The objective is equivalent to finding the infimum $Y_1 + Y_2 + Y_5$ such that task τ_k misses its deadline. If we only test the schedulability condition in Eq. (1) when $t = 4, 5$, we can equivalently formulate this problem as a linear programming:

$$\begin{aligned} \text{minimize} \quad & Y_1 + Y_2 + Y_5 \\ \text{such that} \quad & 4Y_1 + 4Y_2 + 5Y_5 \geq 4 \\ & 5Y_1 + 6Y_2 + 5Y_5 \geq 5 \\ & Y_1, Y_2, Y_5 \geq 0 \end{aligned}$$

The optimal solution of the above linear programming is to set $Y_1 = 0, Y_2 = 0.5, Y_5 = 0.4$. Therefore, the utilization bound is 0.9. The proof for $T_k = 50$ is almost identical by testing only at time $t = 40$ and $t = 50$. □

Combining Lemmas 3.1 and 3.2 shows that the utilization bound of automotive implicit-deadline task sets under RM-P is 90%. The focus of this paper is to further push this bound upwards. In Section 4, we will explain how to derive the parametric utilization bound, which will be superior to 90%.

3.2 Non-Preemptive Schedulers

Under non-preemptive uniprocessor scheduling, a job of τ_k can be blocked by a job of a lower-priority task as executing jobs cannot be preempted, postponing the execution of the job of τ_k. Fortunately, due to the workload-conserving scheduling characteristics under FP-NP, i.e., the processor is never idle if there is a job ready to be executed, a job of task τ_k can only be blocked by at most one lower-priority job. Therefore, the blocking time B_k^* of task τ_k is $B_k^* = \max_{\tau_i \in lp(\tau_k)} (C_i - \varepsilon)$, where $lp(\tau_k)$ is defined as the set of the tasks with priorities lower than τ_k, and ε is an arbitrarily small positive number since a lower-priority job has to first start its execution before blocking task τ_k.

One intuitive way to test the schedulability of task τ_k under FP-NP is to inflate the execution time of task τ_k by B_k^* and apply the critical instant theorem. Therefore, as shown in [7, 12], a sufficient schedulability test of task τ_k under FP-NP is to verify whether

$$\exists t | 0 < t \leq D_k = T_k, \quad B_k^* + C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t}{T_i} \right\rceil C_i \leq t \tag{4}$$

However, the analysis in Eq. (4) is pessimistic since it implicitly implies that task τ_k can still be preempted by a higher-priority task. One possibility to remove the pessimism is to check whether the job of task τ_k can start its execution no later than $r + T_k - C_k$ after it arrives at time r. As shown by Tindell and Burns [28], this is equivalent to the validation of

$$\exists t | 0 < t \le T_k - C_k, \quad B_k^* + \sum_{\tau_i \in hp(\tau_k)} \left(\left\lfloor \frac{t}{T_i} \right\rfloor + 1 \right) C_i \le t \quad (5)$$

However, this is not safe enough. Bril et al. [5] presented the well-known self-pushing phenomenon for FP-NP, showing that deadline misses are possible even if the condition in Eq. (5) is satisfied, as a deadline miss not necessarily happens for the first job of a task under FP-NP. Davis et al. [10] presented an exact schedulability test for FP-NP by exploiting the *busy interval* concept. This test requires to check all the jobs of task τ_k released in the busy interval of task τ_k, i.e., the longest interval starting with a job blocking τ_k where only jobs of tasks in $hp(\tau_k)$ or jobs of τ_k itself are executed.

In some cases, another possibility to reduce the pessimism of the test in Eq. (4) is to adopt the following sufficient schedulability test from Yao, Buttazzo, and Bertogna [32].

Lemma 3.3 (Yao, Buttazzo, and Bertogna, 2010)**.** *The worst-case response time of a non-preemptive task occurs in the first job if the task is activated at its critical instant and the following two conditions are both satisfied:*

- *the task set is feasible under preemptive scheduling;*
- *the relative deadlines are less than or equal to periods.*

Therefore, a sufficient schedulability test for task τ_k under FP-NP is to validate whether Eq. (1) and Eq. (5) both hold.

4 Analysis for RM-P

In this section, we will first present a parametric utilization bound and tight schedulability analyses. Moreover, we will also present an exact schedulability test that only needs to validate 3 inequalities.

4.1 Parametric Utilization Bound

The following two theorems present tighter analysis and a concrete example for the utilization lower bounds for automotive task systems.

Theorem 4.1 (Parametric-Bound-Non-harmonic). *In an automotive implicit-deadline periodic task set, task τ_k is schedulable under RM-P scheduling if T_k is* 5 *and*

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i \le 0.9 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{10} \quad (6)$$

When T_k is 50*, task τ_k is schedulable under RM-P scheduling if*

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i \le 0.9 + \sum_{\tau_i \in \widehat{\mathbf{T}}} \frac{U_i}{10} \quad (7)$$

where $\widehat{\mathbf{T}}$ is $\mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10}$ for notational brevity. The above utilization bounds are lower bounded by 0.9*.*

Proof. We first classify the tasks in $hp(\tau_k)$ into two subsets $hp^<(\tau_k)$ and $hp^=(\tau_k)$, in which a higher-priority task τ_i is in $hp^<(\tau_k)$ if $T_i < T_k$ and is in $hp^=(\tau_k)$ if $T_i = T_k$. For notational brevity, let C'_k be $C_k + \sum_{\tau_i \in hp^=(\tau_k)} C_i$. For a specific t with $0 < t \le T_k$, the left-hand side in the schedulability test in Eq. (1) is equivalent to

$$C'_k + \sum_{\tau_i \in hp^<(\tau_k)} \left\lceil \frac{t}{T_i} \right\rceil C_i$$

We first consider the case when T_k is 5. By the definition of RM-P scheduling, we know that $hp^<(\tau_k)$ is $\mathbf{T}_1$ and $\mathbf{T}_2$. Suppose that task τ_k cannot pass the test in Eq. (1) when we test only $t = 4$ and $t = 5$. For $t = 4$, we have

$$\begin{aligned} & C'_k + \sum_{\tau_i \in \mathbf{T}_1} U_i \times 4 + \sum_{\tau_i \in \mathbf{T}_2} U_i \times 4 > 4 \\ \Rightarrow\ & \frac{C'_k}{5} + \sum_{\tau_i \in \mathbf{T}_1} U_i \times \frac{4}{5} + \sum_{\tau_i \in \mathbf{T}_2} U_i \times \frac{4}{5} > 0.8 \\ \Rightarrow\ & U_k + \sum_{\tau_i \in hp(\tau_k)} U_i > 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} \end{aligned} \quad (8)$$

For $t = 5$, we have

$$\begin{aligned} & C'_k + \sum_{\tau_i \in \mathbf{T}_1} U_i \times 5 + \sum_{\tau_i \in \mathbf{T}_2} U_i \times 6 > 5 \\ \Rightarrow\ & \frac{C'_k}{5} + \sum_{\tau_i \in \mathbf{T}_1} U_i + \sum_{\tau_i \in \mathbf{T}_2} 1.2 U_i > 1 \\ \Rightarrow\ & U_k + \sum_{\tau_i \in hp(\tau_k)} U_i > 1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} \end{aligned} \quad (9)$$

By the inequalities in Eq. (8) and Eq. (9), for a task τ_k with $T_k = 5$ the test in Eq. (1) can only fail at $t = 4$ and $t = 5$ if

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i > \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}, 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} \right\} \quad (10)$$

$$\ge 0.9 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{10} \quad (11)$$

where the $\ge$ comes from the intersection of the two upper bounds above. With similar arguments as for $T_k = 5$, if T_k is 50 and we only test $t = 40$ and $t = 50$ we reach the following conclusion: If the test in Eq. (1) fails at $t = 40$ and $t = 50$ for a task τ_k with $T_k = 50$ then

$$U_k + \sum_{\tau_i \in hp(\tau_k)} U_i > \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5}, 0.8 + \sum_{\tau_i \in \widehat{\mathbf{T}}} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5} \right\} \quad (12)$$

$$\ge 0.9 + \sum_{\tau_i \in \widehat{\mathbf{T}}} \frac{U_i}{10} \quad (13)$$

where the $\ge$ comes from the intersection of the two upper bounds above. Therefore, we reach the conclusion by using contrapositive based on Eqs. (11) and (13). □

Note that Eqs. (2), (11), and (13) determine parametric utilization bounds of $90\% + z_5$ and $90\% + z_{50}$, where

$$z_5 = \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{10} \text{ and } z_{50} = \sum_{\tau_i \in \mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10}} \frac{U_i}{10}$$

The conditions in Eqs. (2), (11), and (13) can be rewritten as follows: $\sum_{\tau_i \in \mathbf{T}} U_i \le 100\%$, $\sum_{\tau_i \in \mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5} U_i \le 90\% + z_5$, and $\sum_{\tau_i \in \mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10} \cup \mathbf{T}_{20} \cup \mathbf{T}_{50}} U_i \le 90\% + z_{50}$, respectively.

We now show that the bounds in Theorem 4.1 are tight.

Theorem 4.2 (Tight-Bound-Non-harmonic). *There exists an automotive implicit-deadline periodic task set with $U_k + \sum_{\tau_i \in hp(\tau_k)} U_i > 0.9$ in which task τ_k is not schedulable by RM-P for a task τ_k in $\mathbf{T}_x$ with $x \in \{5, 50\}$.*

Proof. We prove this theorem by providing two concrete examples. Suppose that $T_k = 5$ and let $\mathbf{T}$ consist of:

- $T_1 = 2, C_1 = 1$ and
- $T_2 = 5, C_2 = 2 + \varepsilon$ with $\varepsilon > 0$ but arbitrarily small.

The utilization of the task set is $0.9 + \varepsilon/5$ and task τ_2 misses its deadline obviously by using the exact test in Eq. (1). For $T_k = 50$, we have to multiple T_1, C_1, T_2, and C_2 with 10, leading to a task set with utilization $0.9 + \varepsilon/5$ again that is not schedulable according to the exact test in Eq. (1). □

This leads to the following corollaries:

Corollary 4.3. *The utilization bound of an automotive implicit-deadline task set is* 90% *which is analytically tight.*

Proof. This corollary follows by combining Lemma 3.1 and Theorems 4.1 and 4.2. □

Corollary 4.4. *For an automotive implicit-deadline task set, if $\sum_{\tau_i \in \mathbf{T}} U_i \leq 100\%$ and*

$$\sum_{\tau_i \in \mathbf{T}} U_i \leq 0.9 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{10} + \Bigg(\sum_{\tau_i \in \mathbf{T}_{100} \cup \mathbf{T}_{200} \cup \mathbf{T}_{1000}} U_i \Bigg) \tag{14}$$

then this task set is schedulable by RM-P.

Proof. Suppose that $\sum_{\tau_i \in \mathbf{T}} U_i \leq 100\%$. By Lemma 3.1, task τ_k can meet its deadline if $T_k = 1, 2, 10, 20, 100, 200, 1000$. By Theorem 4.1, task τ_k with $T_k = 5$ can always meet its deadline since the satisfaction of the condition in Eq. (14) also implies the satisfaction of the condition in Eq. (6). Moreover, task τ_k with $T_k = 50$ can always meet its deadline since the satisfaction of the condition in Eq. (14) also implies the satisfaction of the condition in Eq. (7). Therefore, we reach the conclusion. □

4.2 Efficient Exact Schedulability Test

In the proof of Theorem 4.1, we showed that testing $t = 4$ and $t = 5$ in Eq. (1) when $T_k = 5$ ($t = 40$ and $t = 50$ when $T_k = 50$, respectively) is sufficient to achieve the utilization bound of 90%. The following lemma shows that an exact test only needs to test also these two specific t values in Eq. (1).

Lemma 4.5. *For a task τ_k in $\mathbf{T}_5$, task τ_k is schedulable under RM-P scheduling **if and only if** the schedulability condition in Eq.* (1) *holds for $t = 4$ or $t = 5$. For a task τ_k in $\mathbf{T}_{50}$, task τ_k is schedulable under RM-P scheduling **if and only if** the schedulability condition in Eq.* (1) *holds for $t = 40$ or $t = 50$.*

Proof. We only prove the case $T_k = 50$ since the proof procedure is similar for $T_k = 5$. Let t^* be the minimum value with $0 < t^* \leq 50$ such that $C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t^*}{T_i} \right\rceil C_i = t^*$. We show that the existence of t^* implies either
$C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{40}{T_i} \right\rceil C_i \leq 40$ or $C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{50}{T_i} \right\rceil C_i \leq 50$.

Recall the definition of C'_k, $hp^<(\tau_k)$, and $hp^=(\tau_k)$ in the proof of Theorem 4.1.

- Case 1 when $0 < t^* \leq 40$: This means that
$$t^* = C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t^*}{T_i} \right\rceil C_i \geq C'_k + \sum_{\tau_i \in hp^<(\tau_k)} t^* U_i$$
Clearly, $\sum_{\tau_i \in hp^<(\tau_k)} U_i \leq 1$. Since 40 is an integer multiple of 1, 2, 5, 10, and 20,
$$\begin{aligned} C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{40}{T_i} \right\rceil C_i &= C'_k + \sum_{\tau_i \in hp^<(\tau_k)} 40 U_i \\ &\leq t^* \Big(1 - \sum_{\tau_i \in hp^<(\tau_k)} U_i\Big) + \sum_{\tau_i \in hp^<(\tau_k)} 40 U_i \leq 40 \end{aligned}$$
we reach the conclusion $C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{40}{T_i} \right\rceil C_i \leq 40$.
- Case 2 when $40 < t^* \leq 50$: This means that
$$t^* = C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t^*}{T_i} \right\rceil C_i \geq C'_k + \sum_{\tau_i \in \widehat{\mathbf{T}}} t^* U_i + \sum_{\tau_i \in \mathbf{T}_{20}} 3C_i$$
where $\widehat{\mathbf{T}}$ is $\mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10}$. Clearly, $\sum_{\tau_i \in \widehat{\mathbf{T}}} U_i \leq 1$. Since 50 is an integer multiple of 1, 2, 5, and 10,
$$\begin{aligned} C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{50}{T_i} \right\rceil C_i &= C'_k + \sum_{\tau_i \in \widehat{\mathbf{T}}} 50 U_i + \sum_{\tau_i \in \mathbf{T}_{20}} 3C_i \\ \leq t^* \Big(1 - \sum_{\tau_i \in \widehat{\mathbf{T}}} U_i\Big) + \sum_{\tau_i \in \widehat{\mathbf{T}}} 50 U_i &\leq 50 \Big(1 - \sum_{\tau_i \in \widehat{\mathbf{T}}} U_i\Big) + \sum_{\tau_i \in \widehat{\mathbf{T}}} 50 U_i = 50 \end{aligned}$$
we reach the conclusion $C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{50}{T_i} \right\rceil C_i \leq 50$.

□

Theorem 4.6. *The given automotive implicit-deadline periodic task set is schedulable by RM-P If and only if all the following conditions are satisfied:*

$$\sum_{\tau_i \in \mathbf{T}} U_i \leq 1 \tag{15}$$

$$\sum_{\tau_i \in \mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5} U_i \leq \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}, 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} \right\} \tag{16}$$

$$\sum_{\tau_i \in \widehat{\mathbf{T}} \cup \mathbf{T}_{20} \cup \mathbf{T}_{50}} U_i \leq \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5}, 0.8 + \sum_{\tau_i \in \widehat{\mathbf{T}}} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5} \right\} \tag{17}$$

where $\widehat{\mathbf{T}}$ is $\mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10}$. Therefore, testing Eq. (15)*, Eq.* (16)*, and Eq.* (17) *is an exact schedulability test.*

Proof. This is due to Lemma 3.1 and Lemma 4.5. The last two conditions represent the tests at time $t = 4$ and $t = 5$ for a task τ_k with $T_k = 5$ (from Eq. (10)) and at time $t = 40$ and $t = 50$ for a task τ_k with $T_k = 50$ (from Eq. (12)), respectively. □

5 Non-Preemptive Scheduling

We now provide a sufficient schedulability test for automotive task sets under RM-NP based on the utilization of the task τ_k itself, the utilization of the higher-priority tasks, and the blocking time for task τ_k divided by T_k. Note that the assumption that $C_k < 1$ in the following Theorem is not too restrictive. If $C_k \geq 1$ the task set is unschedulable by default if one task τ_b in the system has a period of 1 due to the blocking time for τ_b.

Theorem 5.1. *Suppose that* $\mathbf{T}$ *is an automotive implicit-deadline periodic task set and* $C_i < 1$ *for every task* τ_i *in* $\mathbf{T}$.

When T_k *is in* $\{1, 2, 10, 20, 100, 200, 1000\}$*, task* τ_k *is schedulable under RM-NP scheduling if*

$$\frac{\max_{\tau_i \in lp(\tau_k)} C_i}{T_k} + U_k + \sum_{\tau_i \in hp(\tau_k)} U_i \leq 1 \tag{18}$$

When T_k *is* 5*, task* τ_k *is schedulable under RM-NP scheduling if the condition in Eq.* (6) *holds and*

$$\frac{\max_{\tau_i \in lp(\tau_k)} C_i}{T_k} + \sum_{\tau_i \in hp(\tau_k)} U_i \leq \max\left\{1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} - U_k, 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}\right\} \tag{19}$$

When T_k *is* 50*, task* τ_k *is schedulable under RM-NP scheduling if the condition in Eq.* (7) *holds and*

$$\frac{\max_{\tau_i \in lp(\tau_k)} C_i}{T_k} + \sum_{\tau_i \in hp(\tau_k)} U_i \leq \max\left\{1 - \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5} - U_k, 0.8 + \sum_{\tau_i \in \widehat{\mathbf{T}}} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5}\right\} \tag{20}$$

where $\widehat{\mathbf{T}}$ *is* $\mathbf{T}_1 \cup \mathbf{T}_2 \cup \mathbf{T}_5 \cup \mathbf{T}_{10}$ *for notational brevity.*

Proof. The condition from Eq. (18) comes from the same analysis as for the if-part in Lemma 3.1 when applying Eq. (4) instead of Eq. (1). We focus ourselves on the other cases when $T_k = 5$ and $T_k = 50$ by using Lemma 3.3, where a sufficient schedulability test is to validate whether both conditions in Eq. (1) and Eq. (5) hold. The condition in Eq. (1) can be tested by using Theorem 4.6. We focus on a utilization-based test by simplifying Eq. (5).

The test in Eq. (5) uses $\left\lfloor \frac{t}{T_i} \right\rfloor + 1$ when summing up the interference from the higher-priority tasks. This test can be translated into using $\left\lceil \frac{t}{T_i} \right\rceil$ instead if we use $B_k = \max_{\tau_i \in lp(\tau_k)} C_i$ as the blocking time instead of $B_k^* = \max_{\tau_i \in lp(\tau_k)} C_i - \varepsilon$. Increasing the blocking time by ε makes the test a bit more pessimistic. However, if ε can be considered to be small compared to the WCETS of the tasks in the task set this pessimism is negligible. For the simplicity of presentation, instead of using Eq. (5), we hereby use the following test

$$\exists t | 0 < t \leq T_k - C_k, \qquad B_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t}{T_i} \right\rceil C_i \leq t \tag{21}$$

Recall the definition of $hp^{<}(\tau_k)$ and $hp^{=}(\tau_k)$ in the proof of Theorem 4.1. In this proof, we will use $C_k^{\dagger} = \sum_{\tau_i \in hp^{=}(\tau_k)} C_i$. We first consider the case when T_k is 5, i.e., $hp^{<}(\tau_k)$ is $\mathbf{T}_1 \cup \mathbf{T}_2$. Suppose that task τ_k cannot pass the test in Lemma 3.3 due to the non-preemptive case using condition Eq. (21). Since $C_k < 1$, it is sufficient to test the condition in Eq. (21) at time $t = 4$ and $t = 5 - C_k$ with $C_k < 1$.

For $t = 4$, we have

$$\begin{aligned}
& B_k + C_k^{\dagger} + \sum_{\tau_i \in \mathbf{T}_1} U_i \times 4 + \sum_{\tau_i \in \mathbf{T}_2} U_i \times 4 > 4 \\
\Rightarrow\ & \frac{B_k}{5} + \frac{C_k^{\dagger}}{5} + \sum_{\tau_i \in \mathbf{T}_1} U_i \times \frac{4}{5} + \sum_{\tau_i \in \mathbf{T}_2} U_i \times \frac{4}{5} > 0.8 \\
\Rightarrow\ & \frac{B_k}{T_k} + \sum_{\tau_i \in hp(\tau_k)} U_i > 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}
\end{aligned} \tag{22}$$

For $t = 5 - C_k$, we have

$$\begin{aligned}
& B_k + C_k^{\dagger} + \sum_{\tau_i \in \mathbf{T}_1} U_i \times 5 + \sum_{\tau_i \in \mathbf{T}_2} U_i \times 6 > 5 - C_k \\
\Rightarrow\ & \frac{B_k}{5} + \frac{C_k^{\dagger}}{5} + \sum_{\tau_i \in \mathbf{T}_1} U_i + \sum_{\tau_i \in \mathbf{T}_2} 1.2 U_i > 1 - U_k \\
\Rightarrow\ & \frac{B_k}{T_k} + \sum_{\tau_i \in hp(\tau_k)} U_i > 1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} - U_k
\end{aligned} \tag{23}$$

By the inequalities in Eq. (22) and Eq. (23), the failure of the test in Eq. (21) at $t = 4$ and $t = 5 - C_k$ happens if

$$\frac{B_k}{T_k} + \sum_{\tau_i \in hp(\tau_k)} U_i > \max\left\{1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} - U_k, 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}\right\} \tag{24}$$

Using contrapositive, we reach the conclusion in Eq (19).

When T_k is 50, we test $t = 40$ and $t = 50 - C_k$, which leads to the conclusion in Eq (20). □

Note that while Theorem 4.6 is an exact test, the test in Theorem 5.1 is only sufficient as the blocking time is greedily included. In general, to verify the schedulability under FP-NP, the schedulability of each task has to be verified individually. The reason is that the blocking time is a decreasing function with respect to the priority, i.e., a task τ_j with a lower priority than task τ_i may be schedulable while τ_i is not schedulable because the blocking time for τ_i is larger.

6 Angle-Synchronous Tasks

In addition to the periodic tasks, an automotive task system may involve *event-triggered* aperiodic/sporadic tasks. One specific type is the angle-synchronous tasks where the jobs are triggered by the rotation of the crankshaft. According to [17], the inter-arrival time between two jobs of an angle-synchronous task for engine control can be modeled as

$$\frac{120}{rpm \times \#cyl} \times 1000 \text{ milliseconds} \tag{25}$$

where rpm is the revolutions per minute of the engine and #cyl is the number of cylinders. Even though the inter-arrival time of the jobs in such angle-synchronous tasks changes over time, they are scheduled based on the fixed-priority scheduling. We consider two general approaches for the priority assignment of those angle-synchronous tasks:

1. assigning them to the highest priority
2. the priorities are assigned according to the shortest possible inter-arrival time between to jobs, i.e., the inter-arrival time at the maximum rotation speed. For example, for #cyl = 4 and 6000 rpm, this leads to a minimum inter-arrival time of $\frac{120 \times 1000}{6000 \times 4} = 5$ ms.

An angle-synchronous task τ_i with m execution modes can be modeled by using a tuple $< C_i^1, T_i^1, C_i^2, T_i^2, \ldots, C_i^m, T_i^m >$ where C_i^j is the WCET for the j-th mode, and T_i^j is the minimum inter-arrival time of the next job when a job in the j-th mode of task τ_i is released. In the literature, such tasks are also called variable-rate-behaviour tasks [11] or multi-mode tasks [16]. Schedulability tests of such angle-synchronous tasks under FP scheduling have been proposed in [11, 16].

Essentially, for analyzing the schedulability of a periodic task τ_k, we need to calculate the interference from all the higher-priority tasks. There are two existing methods to calculate the interference due to an angle-synchronous task in an interval length Δ. One is to find the worst-case workload by investigating the worst-case release patterns using integer linear programming (ILP) [11] or dynamic programming [16]. Another, as shown in the following lemma, is to safely approximate the interference due to an angle-synchronous task τ_i by using $U_i^{max} = \max_{j \in \{1,\ldots,m\}} \left\{ \frac{C_i^j}{T_i^j} \right\}$, $C_i^{max} = \max_{j \in \{1,\ldots,m\}} \left\{ C_i^j \right\}$, and $T_i^{min} = \max_{j \in \{1,\ldots,m\}} \left\{ T_i^j \right\}$

Lemma 6.1. *The maximum interference $I_i(\Delta)$ incurred by an angle-synchronous task τ_i in an interval of length Δ is at most*

$$I_i(\Delta) = \begin{cases} U_i^{max} \times \Delta + C_i^{max} & \text{if } \Delta > T_i^{min} \\ C_i^{max} & \text{if } \Delta \leq T_i^{min} \end{cases} \tag{26}$$

Proof. This is based on Theorem 1 from Davis et al. [11] and Lemma 2 from Huang and Chen [16]. Without loss of generality, let the interval start from 0. Theorem 1 in [11] shows that the maximum interference by an angle-synchronous task τ_i to a lower-priority job arriving at time 0 happens in the following worst-case pattern: a) release the first job at time 0, b) follow the minimum period needed in the particular execution mode, and c) release the last job with execution time C_i^{max} before Δ. We consider the two cases individually. For the first case, i.e., $\Delta > T_i^{min}$, let $t^* < \Delta$ be the arrival time of the last job in the above pattern. Lemma 2 in [16] proves that the maximum interference from 0 to t^* (without the last job) is at most $U_i^{max} \times t^*$. Therefore, by including the (last) job released at or after t^*, the maximum interference incurred by τ_i is at most $U_i^{max} \times \Delta + C_i^{max}$. In the second case, i.e., $\Delta \leq T_i^{min}$, only one job of the angle-synchronous task is released. Therefore, the interference is at most C_i^{max}. □

We can now revise the schedulability test in Eq. (1) to further consider angle-synchronous tasks: An implicit-deadline periodic task τ_k is schedulable under FP-P scheduling if

$$\exists t | 0 < t \leq T_k, \;\; C_k + \sum_{\tau_i \in hp(\tau_k)} \left\lceil \frac{t}{T_i} \right\rceil C_i + \sum_{\tau_i \in \mathbf{T}_{as}} I_i(t) \leq t \tag{27}$$

where $hp(\tau_k)$ is the set of the *periodic* tasks with priorities higher than task τ_k and $\mathbf{T}_{as}$ is the set of the *angle-synchronous* tasks with higher priority. The schedulability test in Eq. (27) does not significantly increase the difficulty for testing the schedulability of a periodic task τ_k under FP-P. All the utilization-based schedulability tests in Section 4 and the test in Theorem 5.1 can easily be revised by considering the interference from the angle-synchronous tasks based on Eq. (27). For example, by revising Theorem 4.6 we get:

Theorem 6.2. *Suppose that tasks in $\mathbf{T}_{as}$ are assigned to higher priorities than any periodic task, and the priorities of the periodic tasks are assigned by the rate-monotonic priority assignment. For each value y in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ let $\widehat{\mathbf{T}}_y$ be defined as the set of periodic tasks with period less than or equal to y for notational brevity. The given automotive implicit-deadline periodic task set is schedulable under the fixed-priority preemptive scheduling if the angle-synchronous tasks are schedulable at the highest priority and all the following conditions are satisfied:*

$$\sum_{\tau_i \in \widehat{\mathbf{T}}_{\mathbf{x}}} U_i + \sum_{\tau_i \in \mathbf{T}_{as}} \frac{I_i(x)}{x} \leq 1 \qquad \forall x \in \{1, 2, 10, 20, 100, 200, 1000\} \tag{28}$$

$$\sum_{\tau_i \in \widehat{\mathbf{T}}_5} U_i \leq \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5} - \sum_{\tau_i \in \mathbf{T}_{as}} \frac{I_i(5)}{5}, 0.8 + \sum_{\tau_i \in \mathbf{T}_1 \cup \mathbf{T}_2} \frac{U_i}{5} - \sum_{\tau_i \in \mathbf{T}_{as}} \frac{I_i(4)}{5} \right\} \tag{29}$$

$$\sum_{\tau_i \in \widehat{\mathbf{T}}_{50}} U_i \leq \max \left\{ 1 - \sum_{\tau_i \in \mathbf{T}_{20}} \frac{U_i}{5} - \sum_{\tau_i \in \mathbf{T}_{as}} \frac{I_i(50)}{50}, 0.8 + \sum_{\tau_i \in \widehat{\mathbf{T}} \cup \mathbf{T}_{20}} \frac{U_i}{5} - \sum_{\tau_i \in \mathbf{T}_{as}} \frac{I_i(40)}{50} \right\} \tag{30}$$

Proof. The proof follows directly by considering the interference due to the angle-synchronous tasks in Eq. (27) and repeating the same procedures in the proofs of Section 4. □

Note that the schedulability has to be tested individually for each period as the interference due to the utilization U_i^{max} of the angle synchronous tasks is constant for each period while the interference due to C_i^{max} decreases when the period is increased. In addition, testing Eq. (28), Eq. (29), and Eq. (30) is only a sufficient test while Theorem 4.6 was an exact schedulability test. This is due to the fact that the terms we introduce to calculate the interference due to angle-synchronous tasks are safe approximations but not tight.

Putting the angle-synchronous tasks to the highest priority from a scheduling point of view introduces unnecessary pessimism to the system, as tasks with high priority are postponed while angle-synchronous tasks that arrived later and have a larger relative deadline are executed. Instead, regarding schedulability, those tasks should be scheduled according to their minimal inter-arrival time if this value can determined safely. However, if the angle-synchronous tasks are only considered according to their minimal inter-arrival time this would most likely not lead to harmonic periods. We propose to determine the priority of the angle-synchronous tasks based on the minimal inter-arrival time T_{as} and to analyse them assuming they have a period that is the maximum $p \in \{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ with $p \leq T_{as}$. This means that T_{as} is removed from the analysis when only tasks in $\widehat{\mathbf{T}}_p$ are considered and the schedulability test for the angle-synchronous tasks can be done by using Eq. (28), Eq. (29), or Eq. (30), depending on p. For example, assuming a maximum of 6000 rpm and 6 cylinders, according to Eq. (25), the angle-synchronous tasks would have a minimal inter-arrival time of ≈ 3.33. Therefore the angle-synchronous tasks would have a priority between the tasks in $\mathbf{T}_2$ and $\mathbf{T}_5$. They would be removed from the analysis for $x = 1$ and $x = 2$ and the schedulability of the angle-synchronous tasks would be determined using Eq. (28) for $x = 2$. We analyze the impact of this approach in the evaluations.

7 Evaluations

As we analyze the schedulability of implicit-deadline periodic tasks in automotive systems, it would be the best to analyze task sets of real-world applications. However, real-world automotive task sets are not publicly available. Therefore, we use synthetic task sets that are similar to real-world task sets, based on "Real world automotive benchmarks for free" by Kramer, Ziegenbein, and Hamann [17] in WATERS 2015. Note that similar period distributions are used in other works related to automotive applications, e.g., in [14, 27, 29].

In automotive systems, the entity for scheduling is a Runnable, which is equivalent to a task in this paper, i.e, the information mentioned in [17] about Runnables is used to create tasks. We analyzed the schedulability of those task sets using the schedulability tests presented in this paper, both for scaled and for unscaled task sets, both for RM-P and RM-NP, without considering angle-synchronous tasks in Section 7.2. Task sets with angle-synchronous tasks is analyzed in Section 7.3.

7.1 Evaluation Setup

The information used when creating the task sets comes from Tables III, IV, V in [17] and is presented in Table 1 in a compact form. For each period in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ ms and for angle-synchronous tasks the distribution of the tasks among those periods is given in Table III in [17]. The minimum, average, and maximum value of the average-case execution time of tasks is given in Table IV in [17] for each of those periods. According to [17], the distribution of those values can be approximated using a Weibull distribution that has the probability density function:

$$f(x) = \frac{k}{\lambda} \cdot \left(\frac{x}{\lambda}\right)^{k-1} \cdot e^{-\left(\frac{x}{\lambda}\right)^k} \quad (31)$$

for $x \geq 0$ under given values for the shape parameter k and the scale parameter λ. As only C_{min}, $C_{average}$, and C_{max} of the distribution are give for each period we numerically approximated k and λ for each period, using the maximum likelihood estimators for k and λ as the starting values, then drew a sample containing 10000 numbers using those values for k and λ, compared the results with the given values for C_{min}, $C_{average}$, and C_{max}, and adjusted the values for k and λ based on the resulting distribution, iterating until the resulting distribution matched the values for C_{min}, $C_{average}$, and C_{max} given in [17].[2] Based on those average execution time values the WCETs can be calculated by scaling up the average execution time with a randomly distributed factor in the interval $[f_{min}, f_{max}]$ related to the period of the task.

[2] As the Weibull distribution is only approximated up to a certain accuracy, drawn C_i values that were not in the related $[C_{min}, C_{max}]$ were discarded.

Period	Share	Average ET in μs			WCET factor	
		Min	Avg.	Max	f_{min}	f_{max}
1 ms	3%	0.34	5.00	30.11	1.30	29.11
2 ms	2%	0.32	4.20	40.69	1.54	19.04
5 ms	2%	0.36	11.04	83.38	1.13	18.44
10 ms	25%	0.21	10.09	309.87	1.06	30.03
20 ms	25%	0.25	8.74	291.42	1.06	15.61
50 ms	3%	0.29	17.56	92.98	1.13	7.76
100 ms	20%	0.21	10.53	420.43	1.02	8.88
200 ms	1%	0.22	2.56	21.95	1.03	4.90
1000 ms	4%	0.37	0.43	0.46	1.84	4.75
angle-syn.	15%	0.45	6.52	88.58	1.20	28.17

Table 1. The information to generate the automotive task sets, combined from Table III, Table IV, and Table V in [17].

Those scaling factors can be found in Table V in [17]. As no further information regarding the distribution was given in [17], we used a uniform distribution over $[f_{min}, f_{max}]$.

We conducted evaluations in two general setups: 1) using the reported execution times, i.e., the values generated based on the Weibull distributions were used unscaled, refereed to as *unscaled tasks*, and 2) the values were scaled using the WCET scaling factors. As we wanted to test how the acceptance ratio for the automotive task sets depends on the utilization of the task sets we needed a way to create task sets with a given target utilization. We empirically found that a task set containing unscaled tasks with a total utilization of $\approx 100\%$ contained around 1500 individually drawn random tasks based on the values in Table 1. As individual random draws were very time consuming, we efficiently generated tasks withs with a given utilization by:

- Drawing the periods of 3000 tasks according to the period distribution, i.e., the distribution in Table 1.
- Counting the number of tasks for each period and drawing the execution time of those tasks randomly according to the related Weibull distribution.
- Drawing the scaling factors and calculate WCETs if selected.
- Combining the tasks to $\mathbf{T}_{base}$ and then shuffle $\mathbf{T}_{base}$.
- Taking tasks from $\mathbf{T}_{base}$ until the total utilization is larger than the target utilization U_t.
- If the total utilization U_{sum} of the set is in $[U_t, U_t + \gamma]$ for a threshold value γ, then take the task set.
- If not, discard the last task, take the next task from $\mathbf{T}_{base}$, and check if U_{sum} is in $[U_t, U_t + \gamma]$ etc..

The threshold γ depended on the utilization steps in our experiments, i.e., it was always smaller, normally 0.1. Independent from the settings we always created 1000 task sets under each setting for each utilization value we analyzed.

7.2 Evaluation - General Schedulability

We analyzed the schedulability under RM-P and RM-NP for task sets without angle-synchronous tasks, using the distribution presented in Table 1 and the schedulability tests in Theorem 4.6 and Theorem 5.1 for the preemptive and the

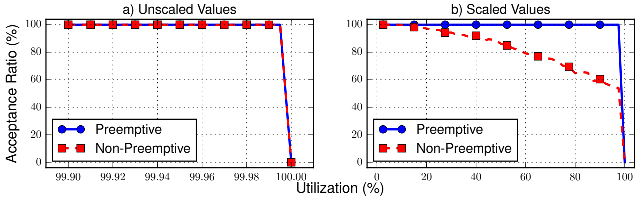

Figure 1. The acceptance ratio of unscaled and scaled task sets for both preemptive and non-preemptive scheduling.

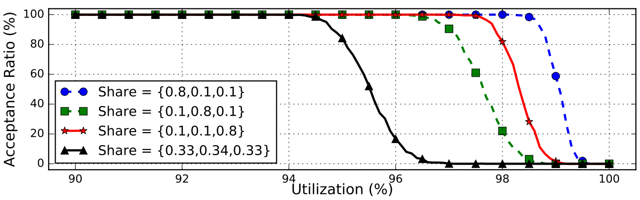

Figure 2. The effect of non-harmonic subsets in Theorem 4.6. The share values indicate the average ratio of tasks with period 1, 2, and 5 ms, respectively.

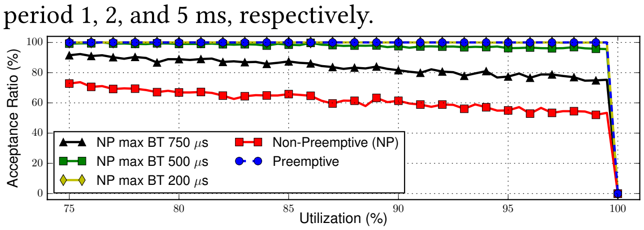

Figure 3. Impact of the maximum blocking time for RM-NP.

non-preemptive case, respectively. Figure 1a and Figure 1b show the results with and without scaling, respectively. The task sets with unscaled values are (nearly) always schedulable when RM-P is used. As the setting in Table 1 does not lead to the case where Corollary 4.4 can be applied directly (i.e., the combined utilization of the tasks with periods 100, 200, and 1000 was always $< 10\%$) we analyzed the utilization values for the individual periods, using task sets with 99.99% utilzation as an example here. As $99.99\% \leq 100\%$, we only have to look at the non-harmonic periods 5 and 50. In the setting provided in Table 1 the combined utilization for periods $1, 2$, and 5 was always $\leq 47.68\%$. Therefore, the tasks up to period 5 are always schedulable as the utilization is below 90%. The combined utilization of periods $1, 2, 5, 10, 20$, and 50 was at most 97.52% while the combined utilization of periods $1, 2, 5, 10$, and 20 was at least 92.78%. Therefore, by putting these values to Eq (17) we get a guaranteed schedulability as $80\% + 92.78\%/5 = 98.556\%$ which is larger than 97.52%. Task sets are never schedulable with 100% utilization for RM-P due to their construction, i.e., we created sets with a utilization in $[U_t, U_t + \gamma]$. Note, that due to the random distribution of task periods it is possible that task sets that have a utilization U^* with $90\% < U^* < 100\%$ are created that are not schedulable in the preemptive case. However, this is very unlikely as it never happened in our evaluations.

Therefore, we additionally analyzed the impact of the distribution of tasks among non-harmonic periods, i.e., for periods of $\{1, 2, 5\}$, for RM-P under different period settings. The results are shown in Figure 2. The individual tasks were created according to the C_i distribution given in Table 1. The probability that a task has period x is according to the share value given for that period in the related label, i.e., it shows the probability to be in $\{T_1, T_2, T_5\}$. If the distribution of probabilities is $\{0.8, 0.1, 0.1\}$ (blue curve) the task sets were always schedulable up to 98.1%. The reason is that the utilization of $\mathbf{T}_1$ is very large in this setting and that task sets are schedulable up to $0.9 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{10}$ (Eq. (11)). The other cases can be explained by looking at Eq. (16) in Theorem 4.6, i.e., $\sum_{\tau_i \in \widehat{\mathbf{T}}_5} \leq \max\left\{1 - \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}, 0.8 + \sum_{\tau_i \in \mathbf{T}_1} \frac{U_i}{5} + \sum_{\tau_i \in \mathbf{T}_2} \frac{U_i}{5}\right\}$. For the green and the red curve, the acceptance ratio drops only a bit earlier than for the blue curve. The reason is that in Eq. (16) either a large or a small utilization for tasks with period 2 leads to a large value on the right hand side. If the tasks are distributed equally over the periods 1, 2, and 5 (black curve) none of the terms in the right hand side of Eq. (16) is as large as in the previous cases, leading to the earlier drop of the acceptance ratio in this case.

When the tasks are scaled (Figure 1b), all task sets with utilization less than 100% are still deemed schedulable for RM-P with similar reasons. For RM-NP the schedulability drops from 10% utilization onwards, because after scaling the execution time it can be larger than 1 ms for some tasks. However, if it is possible to bound the blocking time by a smaller value, i.e., by adopting a limited preemptive approach where tasks are separated into non-preemptive subtasks with a given maximum length, the acceptance ratio can still be very reasonable as shown in Figure 3. Even for a comparatively large maximum blocking time of 750 μs the improvement compared to the non-preemptive case is significant while for a maximum blocking time of 500 μs the acceptance ratio is always above 95.6%. For a maximum blocking time of 200 μs and below the acceptance ratio is the same as in the preemptive case, i.e., the task sets are always schedulable.

7.3 Evaluation - Angle-Synchronous Tasks

In Section 6 we not only provided the schedulability tests for automotive task sets with angle-synchronous tasks, but also stated that the priorities of the angle-synchronous tasks should be set according to their minimum inter-arrival times instead of the highest-priority level if possible. In Figure 4 we compare those two approaches. We consider angle synchronous tasks that have a minimum inter-arrival time of 5 ms, i.e., 6000 rpm and 4 cylinders in Eq. (25). We drew C_i randomly according to Table 1 and set $C_i^{max} = C_i$ and $U_i^{max} = C_i/6000$. We considered two settings.

- In **Setting 1** the periods are distributed according to Table 1. Here the acceptance ratio of the two approaches is nearly identical. However, assigning the priority of the angle-synchronous tasks according to their minimum inter-arrival time (black curve) is never worse than assigning them to the highest priority (green curve).
- In **Setting 2** we considered a different distribution over the periods: the periods distribution is based on Table 1 but we exchanged the probabilities for tasks

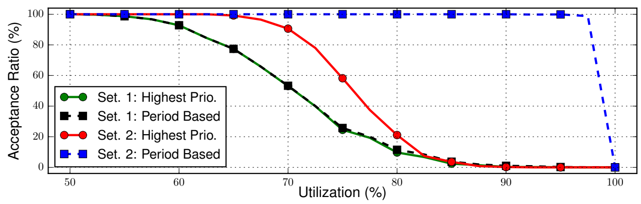

Figure 4. Different strategies for angle-synchronous tasks under different period distributions.

to have period 1 ms and period 10 ms, i.e., period 1 ms has a probability of 25% and period 10 ms one of 3%. In this setting the tasks are nearly always schedulable if the priority of the angle-synchronous tasks is assigned according to the minimal inter-arrival time (blue curve) while the acceptance ratio starts dropping at 65% and is under 20% for a utilization of 80% if the angle-synchronous tasks have the highest priority (red curve). We already observed that a higher utilization for $\mathbf{T}_1$ increases the schedulability, but if the angle-synchronous tasks are scheduled with highest priority they unnecessarily keep the tasks in $\mathbf{T}_1$ from executing, potentially leading to deadline misses.

8 Conclusion

For automotive task sets with nearly harmonic task periods, i.e., they are either $1, 2, 5, 10, 20, 50, 100, 200$, or 1000 milliseconds, we provided parametric utilization bounds under rate-monotonic preemptive scheduling and present an exact schedulability analysis that only tests three simple utilization bounds. The analysis was extended to non-preemptive scheduling by providing an easy sufficient schedulability test. In addition, we show how our schedulability tests can be revised to consider angle-synchronous tasks. The evaluations show that using our parametric utilization bounds and the schedulability tests leads to nearly 100% utilization for preemptive scheduling and to possibly very high utilizations for non-preemptive scheduling, depending on the maximum length of non-preemptive intervals. Our approach can directly include addition harmonic periods, i.e, 0.1, 0.5, and 2000, and can directly be extended to consider additional nearly harmonic periods, i.e, 0.2 and 0.5 together, or 500. An analysis with similar steps to the one for periods 5 and 50 can be adopted for other configurations with nearly harmonic periods, e.g., if the possible periods are 1, 3, 8, and 24 ms, the period of 8 has to be analyzed individually.

References

[1] B. Andersson and E. Tovar. The utilization bound of non-preemptive rate-monotonic scheduling in controller area networks is 25%. In *IEEE Fourth International Symposium on Industrial Embedded Systems - SIES*, pages 11–18, 2009.
[2] E. Bini, G. Buttazzo, and G. Buttazzo. A hyperbolic bound for the rate monotonic algorithm. In *Real-Time Systems, 13th Euromicro Conference on, 2001.*, pages 59–66, 2001.
[3] E. Bini and G. C. Buttazzo. Schedulability analysis of periodic fixed priority systems. *IEEE Trans. Computers*, 53(11):1462–1473, 2004.
[4] E. Bini and G. C. Buttazzo. Measuring the performance of schedulability tests. *Real-Time Systems*, 30(1-2):129–154, 2005.
[5] R. J. Bril, J. J. Lukkien, and W. F. Verhaegh. Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption. *Real-Time Systems*, 42(1-3):63–119, 2009.
[6] A. Burchard, J. Liebeherr, Y. Oh, and S. H. Son. New strategies for assigning real-time tasks to multiprocessor systems. *IEEE Trans. Computers*, 44(12):1429–1442, 1995.
[7] A. Burns. Preemptive priority-based scheduling: An appropriate engineering approach. In *Advances in Real-Time Systems, chapter 10*, pages 225–248. Prentice Hall, 1994.
[8] J.-J. Chen, W.-H. Huang, and C. Liu. k2U: A general framework from k-point effective schedulability analysis to utilization-based tests. In *Real-Time Systems Symposium (RTSS)*, pages 107–118, 2015.
[9] J.-J. Chen, G. von der Brüggen, W.-H. Huang, and R. I. Davis. On the pitfalls of resource augmentation factors and utilization bounds in real-time scheduling. In *Euromicro Conference on Real-Time Systems, ECRTS*, pages 9:1–9:25, 2017.
[10] 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*, 35(3):239–272, 2007.
[11] R. I. Davis, T. Feld, V. Pollex, and F. Slomka. Schedulability tests for tasks with variable rate-dependent behaviour under fixed priority scheduling. In *20th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS*, pages 51–62, 2014.
[12] R. I. Davis, L. George, and P. Courbin. Quantifying the sub-optimality of uniprocessor fixed priority non-pre-emptive scheduling. In *International Conference on Real-Time and Network Systems (RTNS'10)*, 2010.
[13] R. I. Davis, A. Thekkilakattil, O. Gettings, R. Dobrin, and S. Punnekkat. Quantifying the exact sub-optimality of non-preemptive scheduling. In *Real-Time Systems Symposium, 2015 IEEE*, pages 96–106, Dec 2015.
[14] A. Hamann, D. Dasari, S. Kramer, M. Pressler, and F. Wurst. Communication centric design in complex automotive embedded systems. In *29th Euromicro Conference on Real-Time Systems, ECRTS 2017, June 27-30, 2017, Dubrovnik, Croatia*, pages 10:1–10:20, 2017.
[15] C.-C. Han and H.-Y. Tyan. A better polynomial-time schedulability test for real-time fixed-priority scheduling algorithms. In *Real-Time Systems Symposium (RTSS)*, pages 36–45, 1997.
[16] W.-H. Huang and J.-J. Chen. Techniques for schedulability analysis in mode change systems under fixed-priority scheduling. In *Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2015.
[17] S. Kramer, D. Ziegenbein, and A. Hamann. Real world automotive benchmarks for free. In *6th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2015.
[18] T.-W. Kuo, L.-P. Chang, Y.-H. Liu, and K.-J. Lin. Efficient online schedulability tests for real-time systems. *IEEE Trans. Software Eng.*, 29(8):734–751, 2003.
[19] T.-W. Kuo and A. K. Mok. Load adjustment in adaptive real-time systems. In *IEEE Real-Time Systems Symposium*, pages 160–171, 1991.
[20] S. Lauzac, R. G. Melhem, and D. Mossé. An efficient RMS admission control and its application to multiprocessor scheduling. In *IPPS/SPDP*, pages 511–518, 1998.
[21] C.-G. Lee, L. Sha, and A. Peddi. Enhanced utilization bounds for QoS management. *IEEE Trans. Computers*, 53(2):187–200, 2004.
[22] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *IEEE Real-Time Systems Symposium'89*, pages 166–171, 1989.
[23] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.
[24] A. K. Mok. Fundamental design problems of distributed systems for the hard-real-time environment. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1983.
[25] M. Nasri, S. K. Baruah, G. Fohler, and M. Kargahi. On the optimality of RM and EDF for non-preemptive real-time harmonic tasks. In *RTNS*, page 331, 2014.
[26] M. Nasri, M. Mohaqeqi, and G. Fohler. Quantifying the effect of period ratios on schedulability of rate monotonic. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems, RTNS 2016, Brest, France, October 19-21, 2016*, pages 161–170, 2016.
[27] A. Sailer, S. Schmidhuber, M. Deubzer, M. Alfranseder, M. Mucha, and J. Mottok. Optimizing the task allocation step for multi-core processors within autosar. In *2013 International Conference on Applied Electronics*, pages 1–6, Sept 2013.
[28] K. Tindell and A. Burns. Guaranteeing message latencies on control area network (can). In *Proceedings of the 1st International CAN Conference*. Citeseer, 1994.
[29] S. Tobuschat, R. Ernst, A. Hamann, and D. Ziegenbein. System-level timing feasibility test for cyber-physical automotive systems. In *2016 11th IEEE Symposium on Industrial Embedded Systems (SIES)*, pages 1–10, May 2016.
[30] G. von der Brüggen, J.-J. Chen, R. I. Davis, and W.-H. Huang. Exact speedup factors for linear-time schedulability tests for fixed-priority preemptive and non-preemptive scheduling. *Information Processing Letters (IPL)*, 2016.
[31] G. von der Bruggen, J.-J. Chen, and W.-H. Huang. Schedulability and optimization analysis for non-preemptive static priority scheduling based on task utilization and blocking factors. In *Euromicro Conference on Real-Time Systems, ECRTS*, pages 90–101, 2015.
[32] G. Yao, G. Buttazzo, and M. Bertogna. Feasibility analysis under fixed priority scheduling with fixed preemption points. In *Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 71–80, 2010.