Sufficient Response Time Analysis considering Dependencies between Rate-Dependent Tasks

Timo Feld, Frank Slomka
Institute of Embedded Systems/Real-Time Systems
Ulm University, 89069 Ulm, Germany
firstname.lastname@uni-ulm.de

***Abstract*—In automotive embedded real-time systems, such as the engine control unit (ECU), some tasks are activated whenever the engine arrives at a specific angular position. In consequence, the frequency at which this task is activated changes with the speed of the engine i. e. angular velocity. Additionally, these tasks have worst case execution times and deadlines that also depends on the angular velocity. Such tasks exhibit rate-dependent behaviour.**

In recently published works analytical methods for tasks with this rate-dependent behaviour were introduced. Though those methods do not consider dependencies between tasks. For instance one event might be displaced a certain angular position after an event of another task. In this paper, a sufficient analysis will be introduced, which considers those dependencies to improve the accuracy of existing methods.

I. Introduction

Many existing methods for real-time analysis are focused on time-triggered tasks. However, for example on an ECU also rate-dependent tasks are executed. Such as the task that calculates the moment when to ignite the combustion. The higher the angular velocity of the engine, the more frequently this task is executed. Additionally, at higher angular velocities the available time for the execution of the rate-dependent tasks is reduced. Furthermore, different algorithms with different execution times are used at different ranges of angular velocities. In summary, the frequency of activation, the deadlines, and the execution times of such a rate-dependent task depends on the angular velocity of the engine.

Consider the simplified example of two camshaft tasks (an example of a real camshaft task can be found in [1]). The camshaft rotates and triggers task A every 180° (at the specific positions 0° and 180°) and task B every 360° (at the specific position 50°). Lets assume that task A has a higher priority than task B. If task A finishes its execution always within an angular difference below 50°, it won't interfere task B. Letting the relation to a rotating source aside, this general kind of dependency has been referred to as the offset-based dependency [2].

This example shows, that between rate-dependent tasks, dependencies measured in angular position can be observed. Taking these dependencies into account can lead to significant improvement upon the exactness of the schedulability analysis. Therefore we extend in this paper existing real-time analysis to consider dependencies between rate-dependent tasks.

A. Related Work

The problem of scheduling rate-dependent tasks with the dependency of several of its parameters on the speed of the engine has been addressed by Kim et al. [3] (referred to as rhythmic tasks). However, in this analysis exactly one rate-dependent task can be in a task set, and this task must have the highest priority in that task set.

In recent work from Pollex et al. [1] [4] an analysis for an engine control unit is described. In this analysis the maximum response time of rate-dependent tasks (referred to as engine-triggered tasks) is determined by maximizing each parameter separately within certain bounds. Davis et al. [5] present an analysis for tasks with Variable Rate Dependent Behaviour (VRB-tasks). The difference to rate-dependent tasks is the different set of assumptions regarding the relation between execution times and angular velocity. In this analysis the exactness is improved by considering combinations of parameters. Buttazzo et. al [6] extend the response time analysis and the processor demand test (EDF) for VRB-tasks. Biondi et. al [7] present an exact analysis for VRB-tasks under fixed priority scheduling. Those analyses do not consider dependencies between tasks. In this paper, we introduce an analysis with the consideration of dependencies between rate-dependent tasks.

Tindell et al. [8] use a transaction model to analyse dependencies. A transaction t has a minimum period and an offset, which denotes the release of an event after the beginning of the transaction. Palencia et al. [9] extend this analysis for jitter and offsets larger than the period.

Pellizzoni and Lipari [10] and Traore et al. [11] used offsets to describe dependencies and improve the schedulability analysis. Kollmann [2] [12] introduced limiting event streams, which describe task dependencies and improve the precision of schedulability analysis. Henia [13] and Redell [14] capture timing correlations between tasks in distributed systems with treeshaped task dependencies.

All those approaches are about dependencies between time-triggered tasks. For a rate-dependent task, the interconnection between tasks is given in terms of an angle difference, while the execution times are given in time. In this paper we introduce a new schedulability analysis, which considers varying parameters as in [1] and task dependencies which are given in the angle domain.

B. Contributions and Structure

This paper provides the following novel contribution:

- A Response Time Analysis for rate-dependent tasks, which considers dependencies to improve the exactness.

The Analysis is based on the following assumptions:

- the analysis is for fixed priority preemptive scheduling.
- constraint deadlines: the deadline is always lower or equal to the inter-arrival-distance.
- the execution time can vary over the speed. This relation can be expressed with a continuous function.
  - Note the difference to the assumption made in [5] or [7] for VRB-Tasks, where the execution time is expressed with a number of discrete modes.
- the current rate at the occurrence of the event causes the execution time (like for engine-triggered tasks [1][4]).
  - Note the difference to the assumption made in [5] or [7] for VRB-Tasks, where the previous inter-arrival-distance causes the execution time.

The remainder of the paper is structured as follows: The computational model used for the analysis is presented in section II, followed by the real-time analysis in section III. This section is subdivided into the recapitulation of an existing real-time analysis and the analysis capable of analyzing a task set with the consideration of task-dependencies. In section IV experiments are run, comparing results with and without consideration of task-dependencies. The paper is concluded with a summary.

II. Computational Model

In this section an appropriate model for the analysis of rate-dependent tasks is presented. The model consists of two parts, the model for the rotating source, which triggers tasks, such as an engine or a camshaft and a model for rate-dependent tasks.

The real-time analysis is conducted in the interval domain, therefore when referring to an *interval*, the *length* of the interval is meant.

A. Rotating Source

The rotating source is constantly in motion and its physical properties, like velocity or acceleration, change over time due to exterior influences. The angular acceleration α is limited.

$$\alpha : \mathbb{R} \to [-a, a] \tag{1}$$

The absolute value of the angular acceleration does not exceed the value a at any given time. Its antiderivative is the angular velocity (or the speed, both expressions are synonymic used in this paper):

$$\omega : \mathbb{R} \to [w^-, w^+] \tag{2}$$

$$\omega(v) - \omega(u) = \int_u^v \alpha(t)\, \mathrm{d}t \tag{3}$$

The rotating source operates between the minimum angular velocity w^- and the maximum angular velocity w^+, therefore the co-domain of ω is $[w^-, w^+]$.

The antiderivative of the angular velocity ω is the angle ϕ.

$$\phi(v) - \phi(u) = \int_u^v \omega(t)\, \mathrm{d}t \tag{4}$$

The work cycle is the longest difference in angular position, after the specific pattern of events repeats itself. The work cycle is denoted with ζ and given in number of rotations.

B. Tasks

A task is denoted by τ and is event-triggered. This triggering event causes an *activation* of the task by creating an instance of the task and putting it in the ready queue of the scheduler. The scheduling policy is priority-based, therefore a unique priority is assigned to each task, which is denoted by π_τ.

Rate-dependent tasks are triggered in terms of angular position. When the rotating source arrives at one or more specific angular positions the task is triggered. Thus when referring to an angle, the angular position, where an event arrives is meant. For each rate-dependent task τ the angular positions are defined in number of rotations. Thus, when referring to angle units, number of rotations are meant. The k-th angular position of τ is denoted by $\phi_{\tau,k}$. All n positions of τ are given with:

$$\Phi_\tau = \{\phi_{\tau,1}, \phi_{\tau,2}, \ldots, \phi_{\tau,n}\} \tag{5}$$

To obtain schedulability tests for rate-dependent tasks, we need to consider the maximum amount of interference due to a higher priority task τ that can be released in a window of length Δ. The interference of task τ in an interval Δ is denoted with $I_\tau(\Delta)$:

$$I_\tau(\Delta) : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \tag{6}$$

With increasing angular velocity, the available time for computation is reduced. In order to take this into account, different algorithms with varying execution times are used for different ranges of angular velocities. When comparing the number of pulses used to inject the fuel into the cylinder at different angular velocities, then at higher angular velocities fewer pulses are used than at lower angular velocities. This disparity in the used pulses is due to technical limitations. Hence the calculations for the unused pulses are not required and therefore omitted. The execution time of a task may as well depend on the current temperature (as an input variable) and the temperature depends on the angular velocity. In order to model this behavior a continuous function for the execution time is given for each task. In general no correlation can be assumed between the angular velocity and the execution time of the task, i.e. the execution time is not assumed to be monotonically increasing or decreasing with increasing angular velocity.

$$c_\tau : [w^-, w^+] \to \mathbb{R}^+ \tag{7}$$

$c_\tau(w)$ denotes an upper bound on the interval that is required to execute task τ if the angular velocity was w at the time of the activation. Note that the *current* angular velocity causes the length of the execution time.

$d_{\tau,\phi}$ denotes the interval that the execution of event ϕ of task τ is allowed to require since its activation. For rate-dependent tasks, the deadline is given in number of rotations i. e. angle units, to denote this difference of angular positions. Since each angular position can have a different deadline, all deadlines are given with D.

$$D_\tau = \{d_{\tau,\phi_1}, d_{\tau,\phi_2}, \ldots, d_{\tau,\phi_n}\} \tag{8}$$

In summary a task τ consists of four parameters:

$$\tau := (\pi, c, D, \Phi) \tag{9}$$

the priority π, the execution time function c, the set of deadlines D and the set of angular positions Φ. A set of tasks is denoted by Γ.

$\Gamma_{HP,i}$ contains all tasks with a higher priority than task τ_i:

$$\Gamma_{HP,i} = \{\tau \in \Gamma | \pi_\tau > \pi_{\tau_i}\} \tag{10}$$

Tasks have two kinds of dependencies, the dependency of the task to the rotating source (i) and the dependency of tasks between each other (ii):

(i) the task is rate-dependent: execution time, number of events and deadline depend on the speed of the rotating source.

(ii) the offset-based dependency: two (or more) events of different tasks occur at a certain angular difference apart from each other. Note that the length of this difference measured in time depends as well on the rotating source.

Both dependencies will be considered in this paper.

III. Real-time analysis

At first an existing real-time analysis for fixed priorities is restated. This analysis is then extended to a new analysis for varying task activations and execution times, which considers dependencies between tasks to improve accuracy of the schedulability test.

A. Recapitulation of existing real-time analysis

A system is composed of a set of tasks Γ where each task τ is time-triggered and has therefore a constant deadline d_τ. The system is schedulable if the following condition holds:

$$\forall\tau \in \Gamma : R_\tau^+ \leq d_\tau \quad (11)$$

For each task τ in the set of tasks Γ its worst-case response time R_τ^+ is no greater than its deadline d_τ.

Under fixed priority preemptive scheduling, the worst-case response time R_i of a constrained-deadline, sporadic task τ_i corresponds to the length of the longest priority level-i busy period. The busy period comprises two components, the execution time C_i^+ of the task itself, and so called interference, equal to the time for which task τ_i is prevented from executing by higher priority tasks. The length of the busy period r_i, can be computed using the following fixed point iteration [15], with the summation term giving the interference due to the set of higher priority tasks $\Gamma_{HP,i}$.

$$r_i^{q+1} = C_i^+ + \sum_{\forall j \in \Gamma_{HP,i}} \left\lceil \frac{r_i^q}{T_j} \right\rceil C_j^+ \quad (12)$$

Iteration starts with an initial value r_i^0, typically $r_i^0 = C_i^+$, and ends when either $r_i^{q+1} = r_i^q$ in which case the worst-case response time R_i^+, is given by r_i^{q+1}, or when $r_i^{q+1} > d_i$ in which case the task is unschedulable.

B. Sufficient response time analysis for rate-dependent tasks

In this subsection we extend the analysis presented in the previous subsection such that the dependency of the task activations and their execution times on the angular velocity are considered. Especially dependencies between rate-dependent tasks are considered to improve the precision of the analysis.

For simplicity, we suppose the task set only consisting of rate-dependent tasks in this paper. We suggest that an analysis for a combination of time-triggered and rate-dependent tasks, can be done by combining the results of this paper with existing approaches [2] [12]. We leave this analysis for future work.

The system can operate within the given speed range $[w^-, w^+]$. To verify that the system is real-time capable at any possible speed, we would need to calculate the response time for any $w \in [w^-, w^+]$, where w is the speed at the occurrence of the event and can possibly change over time with acceleration a. Since this speed range is a real axis, this would result in an infinite number of values w. Therefore we split instead the range into n equally sized smaller ranges (see figure 1). Note that the speed range could be split arbitrary, but for simplification we choose equally sized smaller ranges.

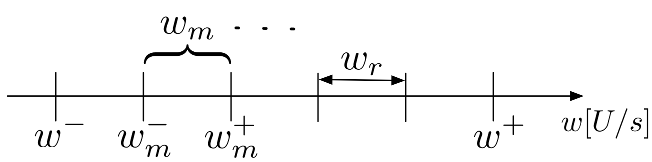

Figure 1: Quantizing Speed Range into Smaller Ranges w_m

Each smaller range has the size: $w_r = \frac{w^+ - w^-}{n}$ and is denoted by: $w_m = [w_m^-, w_m^+]$, with $w_m^- = w^- + m \cdot w_r$ and $w_m^+ = w^- + (m+1) \cdot w_r$ for $m = 0, 1, 2, \ldots, (n-2), (n-1)$.

To verify that the system is real-time capable at any possible speed, we calculate the response time for any w_m, while this smaller range represents the possible initial speeds, that the system has at the occurrence of the event, we are calculating the response time of. Note that starting inside the range w_m, the speed can change over time with acceleration a and can possibly reach any speed within $[w^-, w^+]$. The reachable speeds might be as well limited by the available time. For instance, it can't be reached any speed in a millisecond for a relatively slow acceleration.

A task can have a certain pattern of events occurring, where the distances between those events are arbitrary. Note that any pattern repeats itself after the work cycle. Since each event within the work cycle can have a different response time, we are going to ascertain the response time for every one of those events.

Therefore we extend condition (11):

$$\begin{aligned} &\forall\tau \in \Gamma \wedge \ \forall\phi \in \Phi_\tau \wedge \ \forall w_m \in [w^-, w^+] : \\ &\qquad\qquad R_{\tau,\phi}^+(w_m) \leq d_{\tau,\phi}(w_m) \end{aligned} \quad (13)$$

A task set is schedulable if for every angle ϕ of every task τ that is executed on the task set Γ and for every angular velocity w (covered by all w_m) at which the system can operate, the worst-case response time $R_{\tau,\phi}^+(w_m)$ is no larger than the corresponding deadline $d_{\tau,\phi}(w_m)$.

The activation pattern and the deadlines of rate-dependent tasks are given in angle units. The execution times are given in time units. To be able to compare response time and deadline, we are going to determine the response time in angle units.

In the following subsections, we are going to determine the response time of the event at the kth angle position of task τ_i, $R_{i,k}^+(w_m)$:

- Subsection III-B1: In this subsection a function, which converts time units into angle units and the determination of the maximum execution time of a task is described.
- Subsection III-B2: Here, the interference caused by all previous events is derived.
- Subsection III-B3: This subsection explains the interference caused by all subsequent events.
- Subsection III-B4: Finally, the calculation of the response time is summarized.

1) Maximum Execution Time in "Angle Units": A task is real-time capable, if the relative deadline (for instance 0.5 rotations) is larger or equal than the response time. To be able to compare the relative deadline (given in angle units) with the response time, we are going to determine the response time in angle units. For this purpose, we convert time units into angle units with lemma 1:

Lemma 1. In a given interval Δ the maximum angle difference that can be reached beginning at speed w with acceleration a is:

$$\rho(w, \Delta, a) = \frac{1}{2} \cdot a \cdot {\Delta_1}^2 + w \cdot \Delta_1 + w^+ \cdot \Delta_2 \quad (14)$$

Proof. The interval Δ is split into two. Δ_1 is the length of the interval in which the rotating source accelerates with its maximum possible value a until it reaches the upper limit of the angular velocity w^+. For the remaining interval Δ_2 the rotating source remains at the upper limit of its operational range, since the highest speed causes the most rotations.

$$\Delta_1 = \min\left\{\Delta, \frac{w^+ - w}{a}\right\} \qquad \Delta_2 = \Delta - \Delta_1 \quad (15)$$

With (16) the maximum number of rotations for Δ_1 is computed, if the system accelerates. (17) determines the number of rotations for Δ_2, if the system operates at w^+. The sum of the two results is the maximum angle difference in a time Δ for a given angular velocity w and a maximum acceleration a (18).

$$\rho_1(w, \Delta_1, a) = \frac{1}{2} \cdot a \cdot {\Delta_1}^2 + w \cdot \Delta_1 \tag{16}$$

$$\rho_2(w, \Delta_2, a) = w^+ \cdot \Delta_2 \tag{17}$$

$$\rho(w, \Delta, a) = \frac{1}{2} \cdot a \cdot {\Delta_1}^2 + w \cdot \Delta_1 + w^+ \cdot \Delta_2 \tag{18}$$

□

With lemma 1 we transfer the execution times given in time units ($c_{i,k}(w)$) into angle units ($C^+_{i,k,\phi}$):

Lemma 2. An event with the speed inside w_m at its occurrence, can take no longer for its execution time than:

$$C^+_{i,k,\phi}(w_m, a) = \rho(w^+_m, \max_{w^-_m \le w' \le w^+_m}(c_{i,k}(w')), a) \tag{19}$$

Proof. Formula (14) can be used to get the maximum angle difference in a given time. A higher initial speed, leads to a higher number of rotations. Therefore, w^+_m is used as the initial speed. Since the speed at the occurrence of the event can vary per definition inside the range $[w^-_m, w^+_m]$, $c_{i,k}(w')$ is maximized within this range. □

The aim is to ascertain the response time of an angle ϕ_l of task τ_i for a given speed range w_m. The response time comprises the execution of the task itself plus the interference of all higher priority tasks (see formula (12)). In the following, the maximum interference of a higher priority task τ_j is determined, such that dependencies between tasks given with each activation pattern are considered to improve the precision of the analysis.

Consider the example in figure 2, where two tasks with τ_j having the higher priority are displayed. The first event of τ_j is denoted with $\phi_{j,1}$ and the second with $\phi_{j,2}$. The aim is to compute how long τ_j is interfering the processing of the event of τ_i. While $\phi_{j,1}$ occurs before and $\phi_{j,2}$ after $\phi_{i,k}$, both events have to be considered to get the maximum interference. In the following, it is distinguished whether the event occurs before or after event $\phi_{i,k}$. The interference of all events occurring before are denoted with I_B and the interference of events occurring afterwards are denoted with I_A. In the next subsections III-B2 and III-B3 the computation for each case is explained.

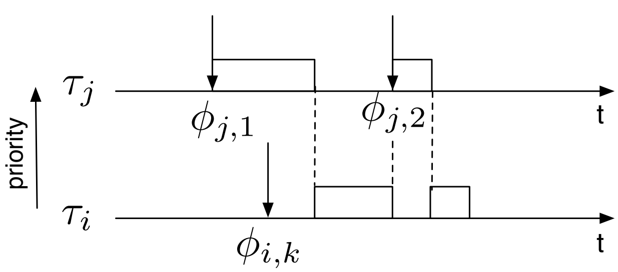

Figure 2: Interference

2) Interference of previous events: In figure 3 a simple example with two tasks is illustrated. Task τ_1 has the higher priority. The lth event of task τ_j: $\phi_{j,l}$ occurs before $\phi_{i,k}$. The difference between those two events is:

$$\varphi_{j,l} = \phi_{i,k} - \phi_{j,l} \tag{20}$$

The value of $\phi_{j,l}$ can be higher than $\phi_{i,k}$, but $\phi_{j,l}$ could be the angle of the previous work cycle. Therefore we use the modulo-operator to handle this and always get positive results for $\varphi_{j,l}$. Since the modulo operation only handles discrete numbers, we convert temporarily from angle units into degrees (1 number of rotation equates 360°):

$$\varphi_{j,l} = ((\lfloor 360 \cdot (\phi_{i,k} - \phi_{j,l}) \rfloor)\%(360 \cdot \zeta))/360 \tag{21}$$

As it can be viewed in figure 3, the maximum possible interference of the previous event of this higher priority task (event $\phi_{j,l}$) is the overlap:

$$I = r_{j,l} - \varphi_{j,l} \tag{22}$$

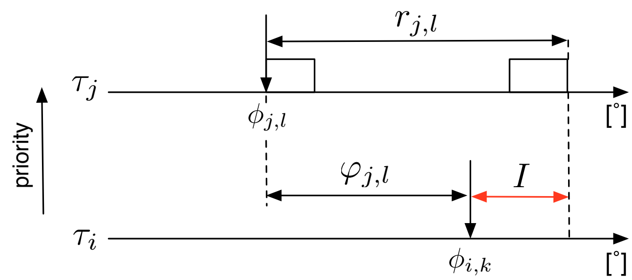

Figure 3: Interference - One Previous Event

Now consider figure 4, with more than one task of a higher priority. Each higher priority task can interfere with the difference $I = r_{j,l} - \varphi_{j,l}$. To find the maximum interference, we maximize over all higher priority tasks. Because there can be no negative interference, we maximize with zero:

$$I_B = \max(\max_{\forall \tau_j \in \Gamma_{HP,i}, \phi_l \in \Phi_{\tau_j}} (r_{j,l} - \varphi_{j,l}), 0) \tag{23}$$

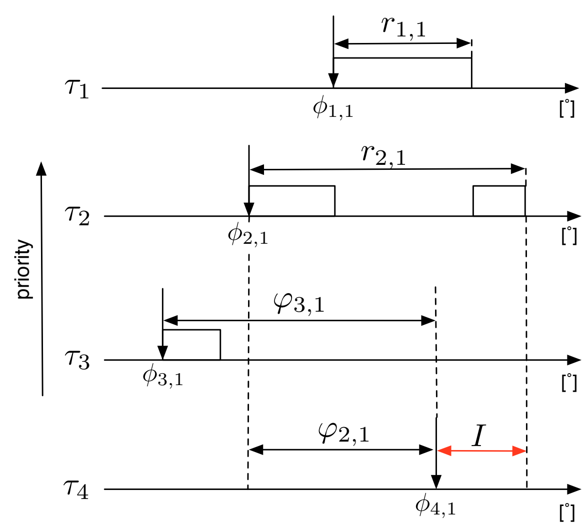

Figure 4: Interference - Previous Events

3) Interference of subsequent events: To determine the execution time of a task with a higher priority, we first derive the bounds of the speed w', which causes the execution time.

Lemma 3. The speed of an interfering event, given an initial speed anywhere in w_m, is always inside these bounds:

$$\begin{aligned} w_{lA} &= \max(\sqrt{w_m^{-2} - 2aU}, w^-) \\ w_{uA} &= \min(\sqrt{w_m^{+2} + 2aU}, w^+) \end{aligned} \tag{24}$$

Proof. The following values are known:

- the speed range $w_m = [w_m^-, w_m^+]$
- the maximum distance to the interfering event in number of rotations: $U = r_i^q$ (where the current result of the fix point iteration is denoted with r_i^q, see formula (12)).
- the acceleration a $[\frac{U}{s^2}]$ and deceleration -a $[\frac{U}{s^2}]$

Accelerating beginning at speed w_m^+ for a time t would result in U number of rotations:

$$U = \frac{1}{2}at^2 + w_m^+ t \tag{25}$$

Next, we rearrange (25) to make t the subject of the equation (the solution with a positive value for t is taken):

$$t = \frac{-w_m^+ + \sqrt{w_m^{+2} + 2aU}}{a} \tag{26}$$

The highest reachable speed in time t is:

$$w_{uA} = w_m^+ + at \tag{27}$$

We replace t of (27) with (26) and get the maximum possible speed, beginning in range w_m:

$$w_{uA} = w_m^+ + a(\frac{-w_m^+ + \sqrt{w_m^{+2} + 2aU}}{a}) = \sqrt{w_m^{+2} + 2aU} \tag{28}$$

Since the speed cannot exceed w^+, we take the minimum between w^+ and equation (28) to get the final upper bound:

$$w_{uA} = \min(\sqrt{w_m^{+2} + 2aU}, w^+) \tag{29}$$

In case of deceleration, we replace a with $-a$ and start at speed w_m^-:

$$U = \frac{1}{2}(-a)t^2 + w_m^- t \tag{30}$$

Next, we solve (30) after t, taking the positive solution:

$$t = \frac{w_m^- - \sqrt{w_m^{-2} - 2aU}}{a} \tag{31}$$

The lowest reachable speed is then:

$$w_{lA} = w_m^- - at \tag{32}$$

Replacing t again:

$$w_{lA} = w_m^- - a(\frac{w_m^- - \sqrt{w_m^{-2} - 2aU}}{a}) = \sqrt{w_m^{-2} - 2aU} \tag{33}$$

For the final lower bound, we take the maximum with the lowest speed possible:

$$w_{lA} = \max(\sqrt{w_m^{-2} - 2aU}, w^-) \tag{34}$$

□

The execution time of event l of the higher priority task τ_j $C_{j,l}^+$ is maximized within the bounds given with lemma 3 and can be converted into angle units:

Lemma 4. The maximum execution time in angle units of event l of the higher priority task j, given acceleration a and the speed range w_m can be determined with:

$$C_{j,l,\phi}^+(w_m, a) = \rho(w_m^+, \max_{w_{lA} \leq w' \leq w_{uA}} (c_{j,l}(w')), a) \tag{35}$$

Proof. The execution time can be maximized within the bounds given by formula (24). This follows from lemma 3. With formula (18) the maximum angle difference that can be achieved in a given time is determined. □

We now focus on the interference of one event $\phi_{j,l}$, with $\phi_{j,l} > \phi_{i,k}$ and with the corresponding task having a higher priority (see figure 5).

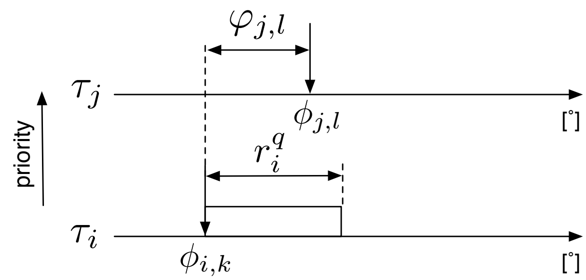

Figure 5: Interference - Subsequent Event

The angle difference is $\varphi_{j,l} = \phi_{j,l} - \phi_{i,k}$. The interval in the current result of the fix point iteration (see (12)) given in angle units is denoted with r_i^q. If the event at $\phi_{j,l}$ occurs after τ_i has processed its event, this event doesn't interfere. Otherwise if the event occurs before, it does interfere. Therefore we distinguish whether r_i^q is larger or lower than $\varphi_{j,l}$.

Lemma 5. The interference of all events $\phi_{j,l}$ occurring after $\phi_{i,k}$ is given by:

$$\begin{aligned} &I_A(r_i^q, w_m, a, \phi_{i,k}) = \\ &\sum_{\forall \phi_{j,l} \in \Phi_{\tau_j} | \tau_j \in \Gamma_{HP,i}} \begin{cases} 0 & , \varphi_{j,l} \geq r_i^q \\ C_{j,l,\phi}^+(w_m, a) & , \varphi_{j,l} < r_i^q \end{cases} \end{aligned} \tag{36}$$

Proof. Note that $\phi_{i,k}$ is the kth event of task τ_i, which is the event we want to compute the response time of and $\phi_{j,l}$ is the lth event of task $\tau_j \in \Gamma_{HP,i}$ (τ_j has a higher priority than τ_i). If the angle difference $\varphi_{j,l}$ is larger or equal than r_i^q this event won't interfere and zero is taken. In the other case ($\varphi_{j,l} < r_i^q$) the event might interfere and $C_{j,l,\phi}^+(w_m, a)$, the maximum execution time of event l of task τ_j, is added to the total interference. $C_{j,l,\phi}^+(w_m, a)$ is given with lemma 4 and takes into account the changing speed. □

4) Response Time: With the total interference of all previous and subsequent events, we can now summarize the response time in angle units:

$$\begin{aligned} &r_{i,k}^{q+1}(w_m, a) = \\ &\quad C_{i,k,\phi}^+(w_m, a) + I_B(r_{i,k}^q, w_m, a) + I_A(r_{i,k}^q, w_m, a) \end{aligned} \tag{37}$$

Iteration starts with an initial value $r_{i,k}^0(w_m, a) = C_{i,k,\phi}^+(w_m, a)$, and ends when either $r_{i,k}^{q+1} = r_{i,k}^q$ in which case the worst-case response time $R_{i,k}^+$, is given by $r_{i,k}^{q+1}$, or when $r_{i,k}^{q+1} > d_{i,k}$ in which case the task is unschedulable.

The deadline is already given in angle units. With the deadline given in angle units and all necessary equations provided for the maximum response time in angle units, we can determine for a task set whether condition (13) holds and thus verify if the task set is real-time capable.

IV. Experiments

To evaluate the performance of the new schedulability test, several tests are applied to compare:

- RTA-SP: A sufficient schedulability test obtained by reducing each rate-dependent task to the sporadic task model by assuming the maximum execution time, the minimum period and the minimum deadline.
- RD: The schedulability test from [1] for rate-dependent tasks.
- RD-DEP: The schedulability test introduced in this paper (condition (13)), considering additional dependencies between tasks.

The parameters for rate-dependent tasks used in our experiments were randomly generated as follows:

- The UUniFast algorithm [16] was used to generate a set of n utilisation values U_i, with a total Utilisation of U.
- The minimum and maximum angular velocity is by default $w^- = 10\ rps$ and $w^+ = 100\ rps$ (equates to a range $[600rpm, 6000rpm]$).
- The work cycle comprises two revolutions: $\zeta = 2$.
- The number of events per work cycle were randomly generated for each task between 1 and 4.
- The angles per work cycle $\phi_1 \dots \phi_k$ were chosen at random according to a uniform distribution.
- Task execution times were set based on the utilisation if the system is operating at a randomly chosen angular velocity w: $c_{\tau_i}(w) = U_i \cdot \frac{\zeta}{w \cdot k}$. The execution times for the remaining w are set with a factor f: $c_{\tau_i}(w) = f \cdot U_i \cdot \frac{\zeta}{w \cdot k}$
- The scaling factor f, representing the variation of the execution time over the angular velocity, is varied from 0.25 to 1 random according to a uniform distribution.
- The deadlines were set to the angle difference to the next event: $d_{\tau_i,l} = \phi_{i,l+1} - \phi_{i,l}$
- The task set cardinality is by default 10.

In our experiment, we compared the performance of the schedulability tests via a metric referred to as the success ratio, i.e. the proportion of randomly generated task sets that are schedulable in each case. In this experiment, 1000 task sets were used for each utilisation level.

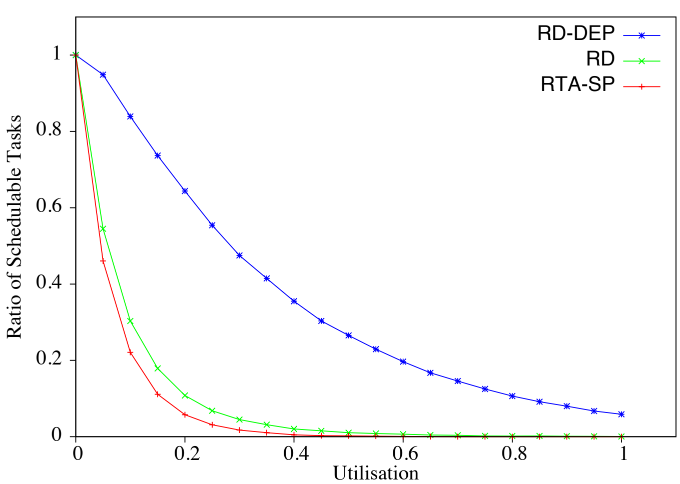

Figure 6: Schedulability rate over Utilisation

Figure 6 shows that the schedulability test with consideration of task-dependencies (RD-DEP) significantly improves upon the default approach (RTA-SP) of treating rate-dependent tasks as if they were sporadic tasks and assuming worst-case parameters and upon the schedulability test without considering dependencies (RD).

V. Summary

In this paper a sufficient response time analysis for rate-dependent tasks was presented. The varying angular velocities over time and the effects on the rate-dependent tasks are taken into account. Additionally dependencies between rate-dependent tasks are considered. Finally the analysis was applied to a big number of randomly generated task sets.

Our experimental results clearly illustrate that the consideration of dependencies improves the accuracy of our sufficient schedulability test.

As a future work, we plan to extend the schedulability analysis for task sets consisting of a combination of time-triggered and rate-dependent tasks.

References

[1] V. Pollex, T. Feld, F. Slomka, U. Margull, R. Mader, and G. Wirrer, “Sufficient real-time analysis for an engine control unit,” 2013.

[2] S. Kollmann, V. Pollex, K. Kempf, and F. Slomka, “A scalable approach for the description of dependencies in hard real-time systems,” pp. 397–411, 10 2010.

[3] J. Kim, K. Lakshmanan, and R. Rajkumar, “Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems,” in *Third International Conference on Cyber-Physical Systems (ICCPS)*, 2012.

[4] V. Pollex, T. Feld, F. Slomka, U. Margull, R. Mader, and G. Wirrer, “Sufficient Real-Time Analysis for an Engine Control Unit with Constant Angular Velocities,” pp. 1335–1338, 2013.

[5] R. 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)*, 2014.

[6] G. Buttazzo, E. Bini, and D. Buttle, “Rate-adaptive tasks: Model, analysis and design-issues,” in *Proceedings of the International Conference on Design, Automation and Test in Europe Conference and Exhibition (DATE)*, 2014.

[7] A. Biondi, A. Melani, M. Marinoni, M. D. Natale, and G. Buttazzo, “Exact interference of adaptive variable-rate tasks under fixed-priority scheduling,” in *Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS)*, 2014.

[8] K. Tindell, “Adding time-offsets to schedulability analysis,” University of York, Computer Science Dept, YCS-94-221, Tech. Rep., 1994.

[9] J. Palencia and M. G. Harbour, “Schedulability analysis for tasks with static and dynamic offsets,” in *The 19th IEEE Real-Time Systems Symposium*, 1998.

[10] R. Pellizzoni and G. Lipari, “Feasibility analysis of real-time periodic tasks with offsets,” *Real-Time Systems*, vol. 30, no. 1-2, pp. 105–128, 2005.

[11] K. Traore, E. Grolleau, A. Rahni, and M. Richard, “Response-time analysis of tasks with offsets,” in *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, 2006.

[12] S. Kollmann, K. Albers, and F. Slomka, “Limiting event streams: A general model to describe dependencies in distributed hard real-time systems,” 2009.

[13] R. Henia and R. Ernst, “Context-aware scheduling analysis of distributed systems with tree-shaped task-dependencies,” in *Proceedings of the conference on Design, Automation and Test in Europe - Volume 1*, ser. DATE ’05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 480–485.

[14] O. Redell, “Analysis of tree-shaped transactions in distributed real time systems,” in *16th Euromicro Conference on Real-Time Systems (ECRTS)*, 2004.

[15] N. C. Audsley, A. Burns, M. Richardson, and A. J. Wellings, “Applying new scheduling theory to static priority pre-emptive scheduling,” *Software Engineering Journal*, 1993.

[16] E. Bini and G. Buttazzo, “Measuring the performance of schedulability tests,” in *Real-Time Systems*, 2005, pp. 129 – 154.