Rate-Adaptive Tasks: Model, Analysis, and Design Issues

Giorgio C. Buttazzo*, Enrico Bini†*, Darren Buttle‡
*Scuola Superiore Sant'Anna, Pisa, Italy
†Lund University, Lund, Sweden
‡ETAS-PGA/PRM-E, Germany
Email: g.buttazzo@sssup.it, e.bini@sssup.it, darren.buttle@etas.com

Abstract

Several real-time systems include computational activities in which both activation rate and functionality are related to the value of a state variable. For instance, in automotive systems, some of the tasks are triggered by specific crankshaft rotation angles and change their functionality based on the angular velocity of the engine.

To deal with such a peculiar type of activities, this paper presents a new task model useful for specifying and analyzing real-time activities with variable computational requirements and activation rates. Schedulability analysis is presented under fixed and dynamic priorities for different scenarios, and design issues are discussed to determine the speeds for triggering mode transitions. Finally, a number of research directions are highlighted to extend the current results to more complex scenarios.

I. Introduction

Some embedded systems require the execution of periodic tasks whose activation rate depends on the value of a state variable. For example, in avionic systems, altimeters are acquired more frequently at low altitudes. Similarly, in mobile robot platforms, proximity sensors are acquired more frequently when the robot gets closer to an obstacle, so their activation rate results to be inversely proportional to the obstacle distance. In automotive applications, several tasks are linked to rotation (e.g., of the crankshaft, gears, or wheels), thus their activation rate is proportional to the angular velocity of a specific device.

A potential problem with such a type of activities is that, for high activation rates, the corresponding utilization can increase beyond a limit, eventually generating an overload condition on the processor. Under fixed priority systems, the overload caused by such a task could delay some lower priority tasks beyond their deadlines, or even prevent them to execute [1], thus leading to a functionality loss.

To avoid such problems, a common practice adopted in automotive applications is to properly design rotation dependent tasks so that they automatically decrease their functionality for increasing speeds [2]. In fact, it is often the case that at higher rotation speeds the system under control becomes inherently more stable, and therefore some functions that must execute at lower speeds do not need to run at higher speed. This can be exploited to reduce the execution time of rotation-driven tasks at higher rotation speeds. A discussion of the basic principles used in an Engine Control Unit (ECU) has been addressed by Kim et al. [3]. Table I illustrates an example of a task with four levels of functionality, specified for different speed intervals.

rotation (rmp)	functions to be executed
[0, 2000]	`f1(); f2(); f3(); f4(); f5();`
(2000, 4000]	`f1(); f2(); f3();`
(4000, 6000]	`f1(); f2();`
(6000, 8000]	`f1();`

Table I
EXAMPLE OF A TASK WITH A FUNCTIONALITY DECREASING WITH THE ROTATION SPEED.

The implementation of such a type of tasks is typically performed as a sequence of conditional `if` statements, each executing a specific subset of functions [2]. For example, Figure 1 shows the pseudo code implementing the task illustrated in Table I.

The research leading to these results was supported by the Linneaus Center LCCC and the Marie-Curie Intra European Fellowship within the 7th European Community Framework Programme.

```
task sample_task {
    omega1 = 2000;
    omega2 = 4000;
    omega3 = 6000;
    omega4 = 8000;

    omega = read_rotation_speed();

    if (omega ≤ omega4) {
       f1();
    }
    if (omega ≤ omega3) {
       f2();
    }
    if (omega ≤ omega2) {
       f3();
    }
    if (omega ≤ omega1) {
       f4();
       f5();
    }
}
```

Figure 1. Typical implementation of a task with a functionality variable with the rotation speed.

The schedulability analysis of a task set that includes such a type of tasks requires estimating the worst-case execution time (WCET) of each function and computing the overall task utilization for each rotation speed. In particular, for the sample task reported in Table I, four different WCETs must be estimated, one for each execution mode.

Under classical analysis, such modal relationships between activation period and WCET are difficult to model and introduce unnecessary pessimism in the analysis, For high-utilization applications found in motor management, such a pessimism can make the difference to the analysis result.

a) **Contributions:** In this paper we propose a new task model that formalizes the specification of rate-adaptive tasks that are activated as a function of a physical variable (e.g., the crankshaft rotation speed) and self-adapt their computational requirements to avoid overloading the system. Moreover, we describe a method for analyzing the schedulability of such systems under both fixed priorities and Earliest Deadline First (EDF) scheduling [4], for all possible values of the physical variable, not only in steady states conditions, but also taking system dynamics into account. Finally, using the analysis carried out under EDF, a design method is proposed to determine a set of safe rotation speeds at which mode switches must occur to avoid overload situations and keep a desired constant utilization.

b) **Paper structure:** The rest of the paper is organized as follows. Section II introduces the task model and the adopted notation. Section III presents the schedulability analysis of systems that include rate-adaptive tasks in quasi-static conditions in which the rotation speed is constant or changes slowly within a task period. The analysis is presented under both fixed priorities and Earliest Deadline First scheduling. Section IV analyzes the more realistic case in which the rotation speed can change according to the typical system dynamics. Section V computes the safe transition rates to keep the maximum utilization of a rate-adaptive task below a given bound. Section VI presents some related work. Section VII concludes the paper and highlights some future research directions.

II. Task model

In this paper, we consider a computing system that has to run a task set Γ of n tasks $\tau_1, \ldots, \tau_n$. Each task can belong to one of two different types: *regular periodic* or *rate-adaptive*. In the following, Γ_p denotes the subset of regular periodic tasks, while Γ_{ra} denotes the subset of rate-adaptive tasks ($\Gamma = \Gamma_p \cup \Gamma_{ra}$).

Both types of tasks are characterized by a worst-case execution time (WCET) C_i, a period T_i, and a relative deadline D_i. However, while for regular tasks such parameters are fixed, for rate-adaptive tasks all the three parameters depend on a system variable. Since this work is motivated by providing a suitable support to automotive applications, we consider the crankshaft rotation speed ω of the car engine as the dynamic variable that determines the actual parameters values of rate-adaptive tasks.

The activation period $T_i(\omega)$ of a rate-adaptive task is given by

$$T_i(\omega) = \frac{2\pi}{\omega}. \tag{1}$$

The execution time $C_i(\omega)$ of a rate-adaptive task τ_i can be model by defining a *set of switching rotation speeds* $\Omega_i = \{\omega_i^1, \ldots, \omega_i^{m_i}\}$, with m_i equal to the number of modes of task τ_i. Since a task has the same functionality in any interval $(\omega_i^{k-1}, \omega_i^k]$, the WCET of a rate-adaptive task can be defined as

$$C_i(\omega) = C_i(\omega_i^k), \quad \forall \omega \in (\omega_i^{k-1}, \omega_i^k], \tag{2}$$

for $k = 1, \ldots, m_i$, where $w_i^0 = 0$ and $C_i(0) = C_i(\omega_i^1)$.

In addition, to precisely define the relative deadline of a rate-adaptive task during rate variations, we also define a *normalized deadline* δ_i, expressed as a fraction of the period, independently of the current speed ω, thus

$$D_i(\omega) = \delta_i T_i(\omega). \tag{3}$$

Figure 2 graphically illustrates $C_i(\omega)$ for the sample task considered in Table I.

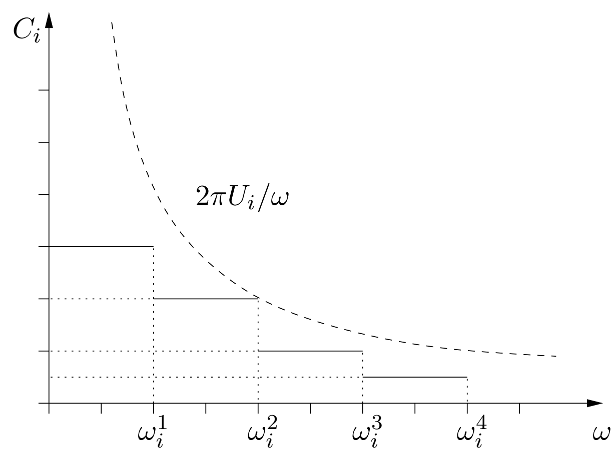

Figure 2. Task WCET as a function of the rotation speed ω.

The actual utilization $u_i(\omega)$ of a rate-adaptive task results to be

$$u_i(\omega) = \frac{C_i(\omega)}{T_i(\omega)} = \frac{\omega\ C_i(\omega)}{2\pi} \tag{4}$$

and has the shape illustrated in Figure 3.

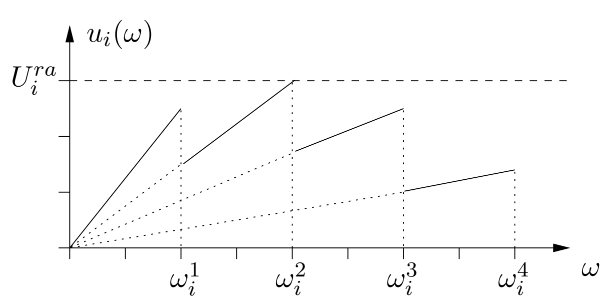

Figure 3. Task utilization as a function of the rotation speed ω.

If the schedulability of the task set must be guaranteed for all possible values of the rotation speed, then the processor utilization of a rate-adaptive task (U_i^{ra}) must be defined as the maximum utilization resulting for each rotation speed, that is

$$U_i^{ra} = \max_{\omega \leq \omega_i^{m_i}} \{u_i(\omega)\} = \max_{\omega \in \Omega_i} \left\{ \frac{\omega \; C_i(\omega)}{2\pi} \right\}. \tag{5}$$

It is worth observing that, by defining U_i^{ra} as in Equation (5), the step function describing $C_i(\omega)$ lies entirely below the hyperbole $2\pi U_i/\omega$ (represented by the dashed curve in Figure 2).

Note that $C_i(\omega)$ can also be expressed as a function of the activation period $T_i(\omega)$. In this case, there is a minimum period that is achieved at the maximum rotation speed ω_{max} reachable by the system:

$$T_{min} = \frac{2\pi}{\omega_{max}}. \tag{6}$$

For the sake of clarity, Figure 4 illustrates $C_i(\omega)$ as a function of $T_i(\omega)$ for the sample task considered in Table I. In this figure, U_i^{ra} coincides with the slope of the dashed line.

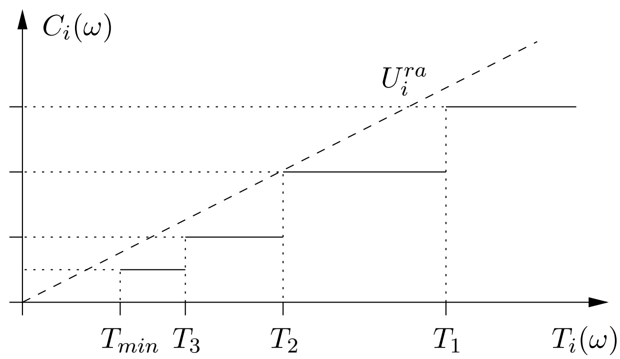

Figure 4. Task WCET as a function of the activation period.

III. Schedulability analysis

This section illustrates how to analyze the schedulability of a task set that includes normal and rate-adaptive tasks. We first consider the case in which tasks are scheduled using a fixed priority assignment and then the case in which they are executed by the Earliest Deadline First (EDF) scheduling algorithm [4]. The analysis presented in this section refers to steady-state conditions, where the rotation speed ω is constant, or changing very slowly within a period. The more general case in which the speed can change significantly within a task period is analyzed in the next section.

A. Fixed-priority scheduling

Under fixed priority scheduling, if a rate-adaptive task is assigned the same priority level for all possible rotation speeds, then task periods may not follow a Rate-Monotonic priority order [4] for all possible rates, thus the schedulability analysis has to be performed using the Response Time Analysis (RTA) [5], even for tasks with relative deadlines equal to periods. Note that schedulability methods based workload analysis [6], [7] are not suited for such types of tasks, because the set of test points in which the test must be performed changes with ω.

If the application includes a single rate-adaptive task, τ_a, with priority P_a, the response time of all tasks with priority higher than P_a can be computed with the classical iterative formula, without any change. The response time R_a of τ_a is a function of the rotation speed ω and, for a given ω, can be computed by iterating the following expression:

$$R_a(\omega) = C_a(\omega) + \sum_{j \in hp(a)} \left\lceil \frac{R_a(\omega)}{T_j} \right\rceil C_j \tag{7}$$

where $hp(a)$ denotes the set of tasks with priority higher than P_a.

The response time of a task τ_i with priority $P_i < P_a$ can be computed by iterating the following expression:

$$R_i(\omega) = C_i + \sum_{j \in hp(i)} \left\lceil \frac{R_i(\omega)}{T_j} \right\rceil C_j \tag{8}$$

where, if τ_j is rate-adaptive, both C_j and T_j are a function of ω.

Hence, a periodic task set with a single rate-adaptive task τ_a is schedulable under a fixed priority assignment if and only if

$$\forall\ i = 1, \ldots, n \quad \max_{\omega \in \Omega_i} R_i(\omega) \leq D_i(\omega). \tag{9}$$

Note that the complexity of the test expressed in Equation (9) is m_a times the complexity of the standard RTA test.

If more than one task are rate adaptive, the definition of the response time requires more computations. Let $\boldsymbol{\omega}_i$ be the vector $(\omega_1, \ldots, \omega_i)$ of rotation rates of the first i highest priority tasks, and let $\boldsymbol{\Omega}_i = \Omega_1 \times \ldots \times \Omega_i$.

Definition 1: The worst-case response time of a task τ_i is defined as

$$R_i = \max_{\boldsymbol{\omega}_i \in \boldsymbol{\Omega}_i} R_i(\boldsymbol{\omega}_i) \tag{10}$$

with $R_i(\boldsymbol{\omega}_i) = R_i(\omega_1, \ldots, \omega_i)$ defined as the fixed point of the following iteration

$$\begin{cases} R_i^{(0)} = C_i(\omega_i) \\ R_i^{(k+1)} = C_i(\omega_i) + \sum\limits_{j \in \mathsf{hp}(i)} \left\lceil \dfrac{R_i^{(k)}}{T_j(\omega_j)} \right\rceil C_j(\omega_j). \end{cases} \tag{11}$$

Unfortunately, computing the response-time in such a way has a complexity of $\prod_i |\Omega_i|$ times the complecity of the RTA, which is exponential in the number of tasks, due to the combinatorial number of all possible speed configurations.

In the following, we propose an alternative schedulability test with reduced complexity, using an upper bound of the response time, as stated in the next theorem.

1) Upper bound:

Theorem 1: Let Ω_i be the set of critical rotation rates of task τ_i and let $\overline{R}_i$ be the fixed point of the following iteration:

$$\begin{cases} \overline{R}_i^{(0)} = \max\limits_{\omega \in \Omega_i} C_i(\omega) \\ \overline{R}_i^{(k+1)} = \max\limits_{\omega \in \Omega_i} C_i(\omega) + \sum\limits_{j \in \mathsf{hp}(i)} \max\limits_{\omega \in \Omega_j} \left\{ \left\lceil \dfrac{\overline{R}_i^{(k)}}{T_j(\omega)} \right\rceil C_j(\omega) \right\}, \end{cases} \tag{12}$$

Then,

$$R_i \leq \overline{R}_i. \tag{13}$$

Proof: Let $\boldsymbol{\omega}^* = (\omega_1^*, \ldots, \omega_i^*)$ be the vector of rotation rates for which the maximum of (10) occurs, meaning that

$$R_i = \max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}_i} R_i(\boldsymbol{\omega}_i) = R_i(\boldsymbol{\omega}_i^*).$$

We prove Equation (13) by induction on the iteration counter of the response time recurrent definition.

a) Step 0: At the initial condition we have

$$R_i^{(0)} = C_i(\omega_i^*) \leq \max_{\omega \in \Omega_i} C_i(\omega) = \overline{R}_i^{(0)}.$$

b) Inductive step: From the inductive hypothesis we have that

$$R_i^{(k)} \le \overline{R}_i^{(k)}.$$

Since the ceiling function is non-decreasing, we have

$$\forall j, \omega, \ \left\lceil \frac{R_i^{(k)}}{T_j(\omega)} \right\rceil C_j(\omega) \le \left\lceil \frac{\overline{R}_i^{(k)}}{T_j(\omega)} \right\rceil C_j(\omega).$$

Then,

$$\begin{aligned} R_i^{(k+1)} &= C_i(\omega_i^*) + \sum_{j \in \mathsf{hp}(i)} \left\lceil \frac{R_i^{(k)}}{T_j(\omega_j^*)} \right\rceil C_j(\omega_j^*) \\ &\le C_i(\omega_i^*) + \sum_{j \in \mathsf{hp}(i)} \left\lceil \frac{\overline{R}_i^{(k)}}{T_j(\omega_j^*)} \right\rceil C_j(\omega_j^*) \\ &\le \max_{\omega_i \in \Omega_i} C_i(\omega_i) + \sum_{j \in \mathsf{hp}(i)} \max_{\omega_j \in \Omega_j} \left\lceil \frac{\overline{R}_i^{(k)}}{T_j(\omega_j)} \right\rceil C_j(\omega_j) \\ &= \overline{R}_i^{(k+1)} \end{aligned}$$

hence, at any iteration k, we have $R_i^{(k)} \le \overline{R}_i^{(k)}$.

Now, let k^* and $\overline{k}^*$ be the iteration index where the fixed point is reached for (11), with $\boldsymbol{\omega} = \boldsymbol{\omega}^*$, and for (12), respectively. Then,

$$R_i = R_i^{(k^*)} \le \overline{R}_i^{(k^*)} \le \overline{R}_i^{(\overline{k}^*)} = \overline{R}_i,$$

which concludes the proof. ■

Theorem 1 only proves that $\overline{R}_i \ge R_i$, so it could be that $\overline{R}_i$ is equal to the exact response time of (10). Unfortunately, the next theorem shows that this is not true.

Theorem 2: There are cases in which $\overline{R}_i > R_i$.

Proof: Let us consider two periodic tasks, τ_1 and τ_2, where τ_1 is rate adaptive, with parameters reported in Table II.

Task	C_i	T_i
$\tau_1(\omega_1)$	5	10
$\tau_1(\omega_2)$	2	4
τ_2	4	20

Table II
TASK SET FOR THE COUNTEREXAMPLE.

For $\omega = \omega_1$ we have (for the sake of clarity, the dependency on ω is not indicated in the formulas):

$$\begin{aligned} R_2^{(0)} &= C_1 + C_2 = 5 + 4 = 9 \\ R_2^{(1)} &= C_2 + \left\lceil \frac{R_2^{(0)}}{T_1} \right\rceil C_1 = 4 + \left\lceil \frac{9}{10} \right\rceil 5 = 9. \end{aligned}$$

Hence $R_2(\omega_1) = 9$. For $\omega = \omega_2$ we have:

$$\begin{aligned} R_2^{(0)} &= C_1 + C_2 = 2 + 4 = 6 \\ R_2^{(1)} &= C_2 + \left\lceil \frac{R_2^{(0)}}{T_1} \right\rceil C_1 = 4 + \left\lceil \frac{6}{4} \right\rceil 2 = 8 \\ R_2^{(2)} &= 4 + \left\lceil \frac{8}{4} \right\rceil 2 = 8. \end{aligned}$$

Hence $R_2(\omega_2) = 8$, and by Definition 1, the worst-case response time of τ_1 is

$$R_2 = \max\{R_2(\omega_1), R_2(\omega_2)\} = 9.$$

However, by applying the test in Theorem 1 we have:

$$\begin{aligned}
\overline{R}_2^{(0)} &= C_2 + \max_{\omega}\{C_1(\omega)\} = 4 + 5 = 9 \\
\overline{R}_2^{(1)} &= 4 + \max\left\{\left\lceil \frac{9}{4} \right\rceil 2, \left\lceil \frac{9}{10} \right\rceil 5\right\} = 10 \\
\overline{R}_2^{(2)} &= 4 + \max\left\{\left\lceil \frac{10}{4} \right\rceil 2, \left\lceil \frac{10}{10} \right\rceil 5\right\} = 10.
\end{aligned}$$

Hence $\overline{R}_2 > R_2$, as required. ∎

2) Lower bound: Here, a lower bound is provided for the task response time.

Definition 2: Let $\overline{k}^*$ be the index at which the iteration (12) reaches a fixed point, and let $\overline{\boldsymbol{\omega}}_i = (\overline{\omega}_1, \ldots, \overline{\omega}_i)$ be the set of rotation rates for which the maxima at the $\overline{k}^*$ iteration are reached.

We define a lower bound of the response time as

$$\underline{R}_i = R_i(\overline{\boldsymbol{\omega}}_i). \tag{14}$$

The proof that $\underline{R}_i$ is a lower bound to R_i of (10) is straightforward, since R_i is maximum over all possible rotation rates, while $\underline{R}_i$ is just the evaluation of the response time for one given set of rotation rates. Note that in the counterexample illustrated in the proof of Theorem 2 the response time lower bound of task τ_2 was $\underline{R}_2 = 8$, since:

$$\begin{aligned}
\underline{R}_2^{(0)} &= 4 \\
\underline{R}_2^{(1)} &= 4 + \left\lceil \frac{4}{4} \right\rceil 2 = 6 \\
\underline{R}_2^{(2)} &= 4 + \left\lceil \frac{6}{4} \right\rceil 2 = 8 \\
\underline{R}_2^{(3)} &= 4 + \left\lceil \frac{8}{4} \right\rceil 2 = 8 = \underline{R}_2.
\end{aligned}$$

B. EDF scheduling

If all tasks are scheduled by the EDF algorithms, then the case in which relative deadlines are equal to periods (that is, when $\delta_i = 1$ for all tasks) can easily be analyzed using the Liu and Layland schedulability test [4], where the utilization U_i^{ra} of the rate-adaptive tasks has to be computed by Equation (5):

$$\sum_{\tau_i \in \Gamma_p} U_i + \sum_{\tau_i \in \Gamma_{ra}} U_i^{ra} \le 1. \tag{15}$$

If tasks have relative deadlines less than periods (that is, if some task has $\delta_i < 1$), the schedulability analysis must be performed using the processor demand criterion [8], where, for a rate-adaptive task, the demand bound function becomes dependent on ω, as follows:

$$\begin{aligned}
\mathsf{dbf}_i^{ra}(t, \omega) &= \left\lfloor \frac{t + T_i(\omega) - D_i(\omega)}{T_i(\omega)} \right\rfloor C_i(\omega) \\
&= \left\lfloor \frac{\omega}{2\pi} t + 1 - \delta_i \right\rfloor C_i(\omega). \qquad (16)
\end{aligned}$$

For a single rate-adaptive task, τ_a, the feasibility test can easily be extended by checking whether, for all possible rotation speeds, the processor demand is no larger than the available time t for all $t \in \mathcal{D}$, where $\mathcal{D}$ is the set of absolute deadlines no larger than a given bound [9]. Note that $\mathcal{D}$ is also a function of ω and that, for any given ω_i^h $(1 < h \le m)$, $\mathsf{dbf}_a^{ra}(t, \omega)$ is a non decreasing function of ω in those intervals in which $C_a(\omega)$ is constant, that is

$$\begin{aligned}
&\forall\, \omega \in (\,\omega_a^{h-1},\ \omega_a^h] \\
&\qquad \forall\, t > 0 \quad \mathsf{dbf}_a^{ra}(t, \omega) \le \mathsf{dbf}_a^{ra}(t, \omega_a^h).
\end{aligned}$$

As a consequence, the feasibility test can be performed only for the given set of rotation rates $\omega \in \Omega_a$. Hence, a periodic task set with a single rate-adaptive task τ_a is schedulable by EDF if and only if

$$\forall \omega \in \Omega_a,\ \forall t \in \mathcal{D}(\omega),\ \sum_{\tau_i \in \Gamma_p} \mathsf{dbf}_i(t) + \mathsf{dbf}_a^{ra}(t, \omega) \leq t. \tag{17}$$

Note that the complexity of the test expressed in Equation (17) is m_a times the complexity of the standard Processor Demand Test. However, extending the exact feasibility test to a task set with k rate-adaptive tasks, each with m transition speeds, increases the complexity by a factor m^k, due to the combinatorial number of all possible speed configurations. Finding a sufficient test under EDF with reduced complexity is part of future work.

IV. Handling dynamic changes

In this section we analyze the case in which the rotation speed can change. The problem that may occur in this situation is that, if a task τ_i is activated at time t, when the rotation speed is ω, if ω increases, the task period may be shorter than $T_i(\omega)$. In fact, let θ_0 be the angle at which τ_i is triggered at time t_0, when the rotation speed is ω_0, and let $\theta_1 = \theta_0 + \Delta\theta$ be the next angular value at which τ_i is triggered again. The situation is depicted in Figure 5.

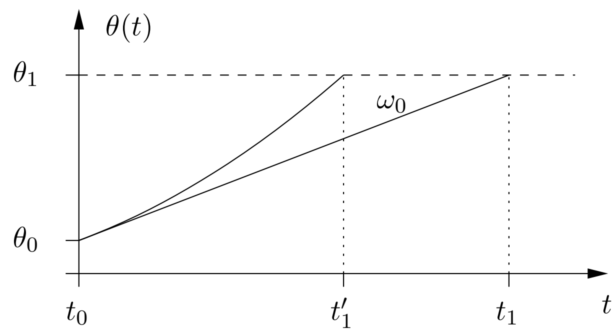

Figure 5. Shaft angle as a function of time.

If the speed is constant, the angle $\theta(t)$ will increase linearly as

$$\theta(t) = \theta_0 + \omega_0(t - t_0).$$

Hence, the value $\theta_1 = \theta_0 + \Delta\theta$ will be reached at time

$$t_1 = t_0 + \frac{\Delta\theta}{\omega_0}$$

leading to an activation period equal to

$$T_i(\omega_0) = t_1 - t_0 = \frac{\Delta\theta}{\omega_0}.$$

If the speed is increasing, the activation period will be shorter than $T_i(\omega_0)$. To compute the new period, $T'_i(\omega_0)$, let us assume that the rotation speed can increase at most with a maximum angular acceleration α, so that the angle $\theta(t)$ will increase at most as a quadratic function of time:

$$\theta(t) = \theta_0 + \omega_0(t - t_0) + \frac{1}{2}\alpha(t - t_0)^2.$$

Hence, the value $\theta_1 = \theta_0 + \Delta\theta$ will be reached after an interval of time x (from t_0) such that

$$\omega x + \frac{1}{2}\alpha x^2 = \Delta\theta.$$

Solving the equation above, we have

$$x_{1,2} = \frac{-\omega_0 \mp \sqrt{\omega_0^2 + 2\alpha\Delta\theta}}{\alpha}$$

and, discarding the negative solution, we find

$$T'_i(\omega_0) = x_2 = \frac{\sqrt{\omega_0^2 + 2\alpha\Delta\theta} - \omega_0}{\alpha}.$$

Hence, if the schedulability of the task set has to be guaranteed in the worst-case scenario, at a given speed ω, the shortest activation period is not $T_i(\omega) = \Delta\theta/\omega$ but

$$T'_i(\omega) = \frac{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega}{\alpha} = \frac{\Delta\theta}{\omega} - \frac{\Delta\theta^2}{2\omega^3}\alpha + o(\alpha). \tag{18}$$

Note that

$$\forall\omega > 0 \quad T'_i(\omega) < T_i(\omega).$$

However, as indicated by the first-order approximation in (18) and illustrated in Figure 6, where T_i and T'_i are plotted as a function of ω (for α = 50 rad/sec^2 and $\Delta\theta = 2\pi$), the difference is more significant for small rotation speeds. In particular, for $\omega = 0$ we have $T'_i(0) = \sqrt{2\Delta\theta/\alpha}$.

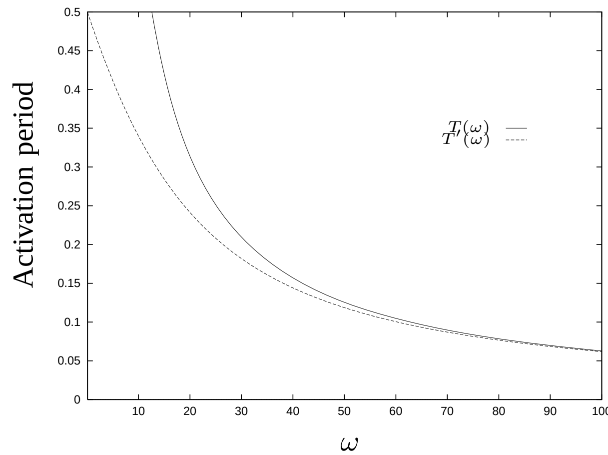

Figure 6. Periods T_i and T'_i as a function of ω.

As a consequence, the actual utilization at speed ω is not $u_i(\omega) = \omega C_i(\omega)/2\pi$, but

$$u'_i(\omega) = \frac{C_i(\omega)}{T'_i(\omega)} = \frac{\alpha\, C_i(\omega)}{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega}$$

Therefore, the utilization of a rate-adaptive task needs to be re-defined as

$$U_i^{ra}(\omega, \alpha) = \max_{\omega\in\Omega_i}\left\{\frac{\alpha\, C_i(\omega)}{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega}\right\}. \tag{19}$$

This result can be used at a design stage to set safe transitions rates such that the actual task utilization never exceeds a desired utilization U_i^d.

V. Setting safe transitions

This section exploits the results of the schedulability analysis under the dynamic case to compute the safe transition rates required to keep the task utilization below a given bound U_i^d, provided at design time.

In Section IV we showed that, if ω is the rotation speed detected at the activation time t, the actual period can be shorter than $2\pi/\omega$, because of the angular acceleration α. This means that, if a rate-adaptive task is required to have a maximum utilization U_i^d, then we can find the minimum $T'_i(\omega)$ that leads to U_i^d, that is

$$T'_i(\omega) = \frac{C_i}{U_i^d}.$$

Inverting Equation (18), we can derive ω as a function of $T'_i(\omega)$:

$$\omega = \frac{\Delta\theta}{T'_i(\omega)} - \frac{\alpha}{2}T'_i(\omega). \tag{20}$$

and imposing $T'_i(\omega) = C_i/U_i^d$ in the equation above we get

$$\omega = \frac{\Delta\theta}{C_i}U_i^d - \frac{\alpha}{2}\frac{C_i}{U_i^d}. \tag{21}$$

So, given a rate-adaptive task τ_i characterized by a set of m modes with WCETs $C_i^{(1)}, \ldots, C_i^{(m)}$, Equation (21) allows computing, for each computation time $C_i^{(k)}$, the maximum transition rate that guarantees not to exceed a desired utilization U_i^d:

$$\forall\ k = 1, \ldots, m \qquad \omega_i^{(k)} = \frac{\Delta\theta}{C_i^{(k)}} U_i^d - \frac{\alpha}{2} \frac{C_i^{(k)}}{U_i^d}. \tag{22}$$

It is worth observing that the first term of the right-hand side of Equation (22) represents the maximum rate for achieving utilization U_i^d in a steady state condition, that is, when the rotation speed is constant ($\alpha = 0$), whereas the second term represents the amount that must be subtracted to take into account the period shrinking due to the acceleration α.

Figure 7 provides a graphic understanding of this concept and illustrates the maximum transition rates that must be adopted for a given set of modes with WCETs $C_i^{(1)}, \ldots, C_i^{(m)}$ to keep the maximum task utilization no larger than a given U_i^d. The dashed curve (plotted for $\Delta\theta = 2\pi$) represents the maximum WCET $C_i(\omega) = \Delta\theta U_i^d/\omega$ allowed in steady state conditions for each ω, whereas the dotted curve represents the maximum WCET allowed in dynamic mode, with maximum acceleration α, given by the following function:

$$C_i(\omega) \ = \ T_i'(\omega)\, U_i^d \ = \ \frac{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega}{\alpha}\, U_i^d.$$

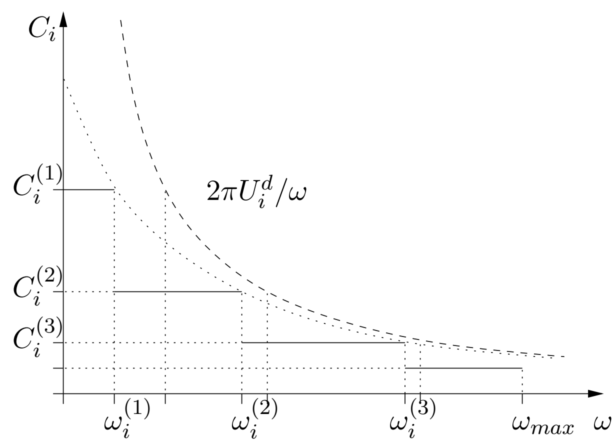

Figure 7. Maximum transition rates to keep a maximum utilization no larger than U_i^d.

Being $T_i(\omega) = \Delta\theta/\omega$, from Equation (22) we can also derive the minimum transition periods that guarantee not to exceed the desired utilization U_i^d:

$$T_i^{(k)} = \frac{2\Delta\theta\ C_i^{(k)}/U_i^d}{2\Delta\theta - \alpha\left(C_i^{(k)}/U_i^d\right)^2}. \tag{23}$$

Figure 8 provides a graphic understanding of this concept and illustrates the minimum transition periods that must be adopted for a given set of m modes with WCETs $C_i^{(1)}, \ldots, C_i^{(m)}$, in order to keep the maximum task utilization no larger than a desired U_i^d. The dotted line represents the function $C_i(\omega) = U_i^d T_i$ corresponding to the steady state condition.

VI. Related Work

Tasks with variable rates have been considered in the real-time literature by several authors, but computation times were typically considered to be constant for different rates.

Jeffay et al. [10], [11] proposed a rate-based execution abstraction, which generalizes the classical periodic and sporadic scheme. According to such a model, a task specifies its expected rate as the maximum number x of

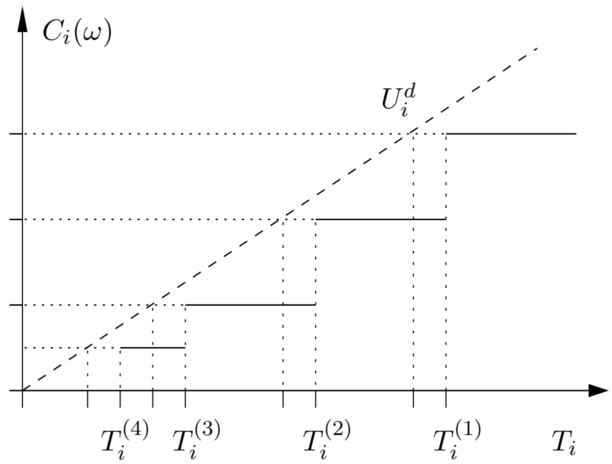

Figure 8. Minimum transition periods to keep a maximum utilization no larger than U_i^d.

executions expected to be requested in any interval of length y, however the maximum computation time required for any job of the task is fixed, while the actual distribution of events in time is arbitrary.

Velasco et al. [12] formulated the analysis for tasks activated by events linked to the dynamics of the plant to be controlled. In this case, the activation pattern of the task is related the system dynamics.

In the multi-frame task model proposed by Mok and Chen [13], tasks are activated periodically, but the execution time of each job varies according to a predefined pattern. Such a model has been later generalized by Baruah et al. [14] to allow jobs to be separated by a varying interarrival time. However, in both cases the activation pattern is known a priori and does not depend on any state variable.

Buttazzo et al. proposed the elastic task model [15], [16], where each task has a fixed computation time, but a variable period, which can vary in a given range $[T_i^{min}, T_i^{max}]$. In this approach, an overload condition generated by a period variation is not handled by the task itself (through a self scaling of its functionality), but at the system level. In particular, the overload is handled by properly compressing task utilizations as they were elastic springs with given elastic coefficients, expressing the availability of each task in changing its period.

Beccari et al. [17], [18] proposed other methods for coping with overload conditions through period adjustments, but task computation times do not adapt with the rate.

Tasks that adapt their computation times to cope with overload conditions have been considered by Abeni and Buttazzo [19], who proposed a hierarchical feedback scheme that combines a global bandwidth compression algorithm with a set of local (task-level) adaptation strategies, including multiple versions. However, each local strategy only considers varying a single task parameter.

Mode change analysis [20], [21] is not suited for describing rate-adaptive tasks, since an infinite number of modes would be required to describe such tasks for all possible rotation speeds.

The case of engine control tasks with activation rates and execution times depending on the angular velocity of the engine has been addressed by Pollex et al. [22], who presented a sufficient schedulability analysis under fixed priorities. The analysis, however, assumes that all the tasks with a variable rate depend on the same angular velocity, which can be arbitrary, but fixed; also, the analysis is formulated using continuous intervals, hence it cannot be immediately translated into a practical schedulability test, whose complexity has not been evaluated.

VII. Conclusions

Many real-time applications require the execution of periodic activities whose rate depends on the system state. In automotive systems, a typical example of such a type of activity is represented by a task activated a specific angles of the crankshaft, hence the higher the rotation speed, the higher the activation rate. The problem introduced by such tasks is that for high activation rates their utilization can quite high, possibly creating an overload condition that, especially under fixed priority scheduling, could prevent the execution of lower priority tasks, so losing some functionality.

To handle such a problem and prevent overload situations, a common practice adopted by software engineers is to disable some functions at high rates, so creating a task that change mode as a function of its activation rate.

To analyze these type of systems and predict their timing behavior for all possible conditions, this paper proposed a new task model in which a task that needs to change its activation rate can run in different modes, each characterized by a specific functions and computational requirements. Each mode is active in a predefined range of values of the variable (e.g., the crankshaft rotation speed) that determines the task activation period.

We have shown that the exact schedulability analysis of a system including rate-adaptive tasks can easily be extended both under fixed priority scheduling and EDF, however the complexity of the resulting schedulability test heavily depend on the specific assumptions. We have also shown how to determine the transition speeds that keep the task utilization below a desired value.

Concerning the schedulability analysis, we have shown that, under fixed priority scheduling, the utilization test [4] cannot be used, even when deadlines are equal to periods, because a Rate-Monotonic order could not be preserved for all possible rates. On the other hand, workload analysis [6], [7] become too complex, since the test points change as a function of the rotation speed. For a single rate-adaptive task, the Response Time Analysis [5] can easily be extended by computing the response time for each critical rotation speed. For k rate-adaptive tasks with m critical speeds, however, the complexity of the Response Time Analysis increases by a factor m^k. To reduce such a complexity, we proposed a simpler but only sufficient test to derive un upper bound of task response times, with a complexity higher than RTA by a factor m.

Under EDF, we first have shown how to extend the utilization test for tasks with relative deadlines are equal to periods. For tasks with constrained deadlines, the we have shown how to extend the processor demand criterion [8], but additional work has to be done to propose a sufficient method with reduced complexity.

Although some results have been presented in this paper for analyzing a system including rate-adaptive tasks, several problems remain to be solved to exploit such a task model in practical situations, including schedulability tests with reduced complexity and their extensions under shared resources and multi-core platforms.

REFERENCES

[1] G. C. Buttazzo, "Rate monotonic vs. EDF: Judgment day," *Real-Time Systems*, vol. 29, no. 1, pp. 5–26, January 2005.

[2] D. Buttle, "Real-time in the prime-time," Keynote speeach given at the 24th Euromicro Conference on Real-Time Systems (ECRTS 2012), Pisa, Italy, July 12th, 2012.

[3] J. E. Kim, O. Rogalla, S. Kramer, and A. Hamann, "Extracting, specifying and predicting software system properties in component based real-time embedded software development," in *Proceedings of the 31st International Conference on Software Engineering (ICSE 2009)*, Vancouver, BC, Canada, May 16-24, 2009, pp. 28–38.

[4] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, January 1973.

[5] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Software Engineering Journal*, vol. 8, no. 5, pp. 284–292, September 1993.

[6] J. Lehoczky, L. Sha, and Y. Ding, "The rate monotonic scheduling algorithm: Exact characterization and average case behavior," in *Proceedings of the 10th IEEE Real-Time Systems Symposium (RTSS'89)*, Santa Monica, CA, USA, December 5-7, 1989, pp. 166–171.

[7] E. Bini and G. C. Buttazzo, "Schedulability analysis of periodic fixed priority systems," *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1462–1473, 2004.

[8] S. Baruah, L. Rosier, and R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Journal of Real-Time Systems*, vol. 2, 1990.

[9] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Third Edition*. New York: Springer, 2011.

[10] K. Jeffay and D. Bennett, "A rate-based execution abstraction for multimedia computing," in *Proceedings of the Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Lecture Notes in Computer Science, Springer-Verlag, Vol. 1018*, Durham, NH, April 1995, pp. 67–78.

[11] K. Jeffay and S. Goddard, "A theory of rate-based execution," in *Proceedings of the 20th IEEE Real-Time Systems Symposium*, Phoenix, AZ, December 1999, pp. 304–314.

[12] M. Velasco, P. Martí, and E. Bini, "Control-driven tasks: Modeling and analysis," in *Proceedings of the* 29^{th} *IEEE Real-Time Systems Symposium*, Barcelona, Spain, November 30 - December 3, 2008.

[13] A. K. Mok and D. Chen, "A multiframe model for real-time tasks," *IEEE Transactions on Software Engineering*, vol. 23, no. 10, pp. 635–645, October.

[14] S. K. Baruah, D. Chen, S. Gorinsky, and A. K. Mok, "Generalized multiframe tasks," *Real-Time Systems*, vol. 17, no. 1, pp. 5–22, July 1999.

[15] G. Buttazzo, L. Abeni, and G. Lipari, "Elastic task model for adaptive rate control," in *IEEE Real Time System Symposium*, Madrid, Spain, December 1998.

[16] G. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, March 2002.

[17] G. Beccari, S. Caselli, M. Reggiani, and F. Zanichelli, "Rate modulation of soft real-time tasks in autonomous robot control systems," in *IEEE Proceedings of the 11th Euromicro Conference on Real-Time Systems*, June 1999.

[18] G. Beccari, S. Caselli, and F. Zanichelli, “A technique for adaptive scheduling of soft real-time tasks,” *Real-Time Systems*, vol. 30, no. 3, pp. 187–215, July 2005.

[19] L. Abeni and G. Buttazzo, “Hierarchical QoS management for time sensitive applications,” in *Proceedings of the IEEE Real-Time Technology and Applications Symposium*, Taipei, Taiwan, May 2001.

[20] J. Real and A. Crespo, “Mode change protocols for real-time systems: A survey and a new proposal,” *Real-Time Systems*, vol. 26, no. 2, pp. 161–197, March 2004.

[21] N. Stoimenov, S. Perathoner, and L. Thiele, “Reliable mode changes in real-time systems with fixed priority or EDF scheduling,” in *Proceedings of the Design, Automation and Test Conference in Europe (DATE 2009)*, Nice, France, April 20-24, 2009.

[22] 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,” in *Proceedings of the Design, Automation and Test Conference in Europe (DATE 2013)*, Grenoble, France, March 18-22, 2013.