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, bini@control.lth.se, darren.buttle@etas.com

Abstract **In automotive systems, some of the engine control tasks are triggered by speciŒ crankshaft rotation angles and are designed to adapt their functionality based on the angular velocity of the engine. This paper proposes a new task model for specifying such a type of real-time activities and presents an approach for analyzing the system feasibility under deadline scheduling for different scenarios. In particular, a feasibility test is derived for tasks under steady-state conditions (constant speed), as well as in dynamic conditions (constant acceleration). A design method is also discussed to determine the most suitable switching speeds for adapting the functionality of tasks without exceeding a desired utilization. Finally, a number of research directions are highlighted to extend the current results to more complex and realistic scenarios.**

I. INTRODUCTION

Several control systems include periodic tasks whose activation rate depends on the value of a state variable. For example, in automotive systems, engine control tasks are linked to the rotation of the crankshaft, thus their activation is triggered by the hardware at specific rotation angles. Other computational activities are linked to the rotation speed of other subsystems (e.g., wheels, gears, engine, cooling fan), thus their activation rate is proportional to the angular velocity of a specific device. In avionic systems, altimeters are acquired more frequently at low altitudes and, similarly, in mobile robot platforms, proximity sensors are acquired more frequently when the robot gets closer to obstacles, so their activation rate results to be inversely proportional to the obstacle distance.

A potential problem with such a type of activities is that, for high activation rates, the system utilization can increase beyond a limit, generating an overload condition on the control processor. If not properly handled, an overload can have disruptive effects on the controlled system, introducing unbounded delays on the computational activities or even leading to a complete functionality loss [1].

To prevent such problems, a common practice adopted in automotive applications is to properly design rotation-dependent tasks so that they automatically decrease their computational requirements (and 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].

In this paper, such self-adjusting activities are referred to as *rate-adaptive tasks*, because they adapt their functional requirements based on their activation rate (which is controlled by the hardware). Table I illustrates an example of a rate-adaptive task with four levels of functionality, specified for different speed intervals, expressed in rotations per minute (rpm). 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]. Figure 1 shows the pseudo code implementing the task illustrated in Table I.

crankshaft rotation speed (rpm)	functions to be executed in a given speed range
[0, 2000]	`f1(); f2(); f3(); f4();`
(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 schedulability analysis of systems that include 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 a difference to the result of the feasibility analysis.

The research leading to these results was supported by the Linneaus Center LCCC, the ELLIIT Excellence Center, and the Marie-Curie Intra European Fellowship within the 7th European Community Framework Programme.
978-3-9815370-2-4/DATE14/©2014 EDAA

```
#define omega1 2000
#define omega2 4000
#define omega3 6000
#define omega4 8000

task  sample_task {
      omega = read_rotation_speed();

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

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

A. 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. [4], [5] proposed a rate-based execution abstraction that generalizes the classical periodic and sporadic scheme. According to such a model, a task specifies its expected rate as the maximum number x of executions expected to be requested in any interval of length y, however the maximum computation time required by any job of the task is fixed, while the actual distribution of events in time is arbitrary.

Velasco et al. [6] formulated the analysis for tasks activated by events linked to the dynamics of the plant to be controlled, but computation times are fixed for each job.

In the multi-frame task model proposed by Mok and Chen [7], 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. [8] 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 [9], [10], where each task has a fixed computation time, but a variable period, which can vary in a given range. 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 by a global resource manager. 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 to change its period.

Beccari et al. [11], [12] 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 [13], 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 [14], [15], [16] is not suited for describing rate-adaptive tasks, because their activation periods change continuously, thus an infinite number of modes would be required to describe all possible situations.

A task model suitable for engine control tasks with activation rates and execution times depending on the angular velocity of the engine has been proposed for the first time by Kim, Lakshmanan, and Rajkumar [17], who derived preliminary schedulability results under simple assumptions. In particular, their analysis applies to a single rate-adaptive task with a period always smaller than the periods of the other tasks, and running at the highest priority level. In addition, they assume that all relative deadlines are equal to periods and priorities are assigned based on the Rate-Monotonic algorithm.

Pollex et al. [18] also presented a sufficient schedulability analysis under fixed priorities, but they assumed that all the tasks with a variable rate depend on the same angular velocity, which can be arbitrary, but fixed. Moreover, the analysis is formulated using continuous intervals, hence it cannot be immediately translated into a practical schedulability test, whose complexity has not been evaluated.

Contributions: This paper addresses both schedulability analysis and design issues of rate-adaptive tasks that are activated as a function of physical variables and self-adjust their computational requirements to avoid overloading the system. The proposed approach extends the state of the art [17], [18] in several directions:

1) It considers multiple rate-adaptive tasks, each of which can be activated by an independent system variable.
2) Schedulability results are derived under Earliest Deadline First (EDF) scheduling, for all possible values of the physical variable, not only in steady states conditions, but also taking system dynamics into account.
3) The period of each rate-adaptive task is not constrained as in [17] to be smaller than those of regular periodic tasks, but can change continuously in a given arbitrary range.
4) Finally, a design method is proposed to determine, for each rate-adaptive task, a set of switching speeds at which the functionality can be adjusted to keep the task utilization below a desired value.

The reason for selecting EDF as a scheduler is that an exact feasibility analysis of rate-adaptive tasks under fixed priorities is characterized by a very high computational complexity. In fact, using the response time analysis (RTA), given m rate-adaptive tasks using k execution modes, feasibility must be checked for all possible combinations of switching speeds, leading to k^m tests, each having pseudo-polynomial complexity.

A similar complexity arises under EDF for tasks with relative deadlines different than periods, since exact tests require checking the overall computational demand in a set of test points that, for rate-adaptive tasks, depend on the value of the state variable. On the other hand, for tasks with implicit deadlines, the exact analysis under EDF has a linear complexity, hence it can effectively be used in such a scenario. In addition, OSEK-compliant kernels that support EDF as a task scheduler exist today [19], so the results presented in this paper can actually be implemented with a limited effort.

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 under EDF for fixed rotations speeds. Section IV analyzes the case in which rotation speeds can change with a given maximum acceleration. Section V discusses how to compute the set of transition rates to keep the maximum utilization of a rate-adaptive task below a given desired bound. Finally, Section VI concludes the paper highlighting some future research directions.

II. Task Model

In this paper, we consider a computing system running a set of n real-time preemptive tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$. Each task can belong to one of two different types: regular *periodic* or *rate-adaptive*. In the following, the subset of regular periodic tasks is denoted as Γ^P and the subset of rate-adaptive tasks is denoted as Γ^R, so that $\Gamma = \Gamma^P \cup \Gamma^R$ and $\Gamma^P \cap \Gamma^R = \emptyset$. For the sake of clarity, whenever needed, a rate-adaptive task may also be denoted as τ_i^*.

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-related variable, ω_i, also referred to as *angular velocity* or *rotation speed* associated with task τ_i^*. In particular, we assume that the period of a rate-adaptive task is inversely proportional to its related angular velocity ω_i:

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

Note that, although ω_i is related to a system state variable, it may properly be redefined for each specific task to allow more flexibility while keeping a uniform interface for all rate-adaptive tasks. For instance, let ω_c be the angular velocity of the crankshaft and suppose that the application includes two rate-adaptive tasks, τ_1^* and τ_2^*, where τ_1^* is activated every quarter turn of the crankshaft, while τ_2^* is activated every half turn. Hence, it will be $\omega_1 = 4\omega_c$ and $\omega_2 = 2\omega_c$. If there is another task τ_3^* whose activation period is related to the obstacle distance d detected by the parking sensor (e.g., $T_3 = Kd$), then T_3 can still be equal to $2\pi/\omega_3$ by defining $\omega_3 = 2\pi/Kd$. As done by Kim et al. [17], we assume $D_i = T_i$ for all the tasks.

The execution time $C_i(\omega_i)$ of a rate-adaptive task τ_i^* can be modeled 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

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

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

It is worth observing that the rate-adaptive task model does not necessarily assume that WCETs must decrease for higher switching speeds. Although this is a common practice in some application domains to control the task utilization, the proposed analysis applies to arbitrary WCET values.

III. Schedulability Analysis

This section illustrates how to analyze the schedulability of a task set that includes normal and rate-adaptive tasks, in a steady-state situation in which all state variables are arbitrary but fixed. The dynamic case in which they are not constant is addressed in Section IV. Under EDF scheduling, if relative deadlines are equal to periods, the task set feasibility can easily be analyzed using the Liu and Layland schedulability test [20] as a function of task utilizations. While a regular periodic task has a constant utilization $U_i = C_i/T_i$, for a rate-adaptive task the utilization is a function of ω_i and results to be

$$u_i(\omega_i) = \frac{C_i(\omega_i)}{T_i(\omega_i)} = \frac{\omega_i\ C_i(\omega_i)}{2\pi}. \tag{3}$$

If the schedulability of the task set must be guaranteed for all possible values of ω_i, then the maximum processor utilization U_i^* of a rate-adaptive task must be defined as

$$U_i^* = \max_{\omega_i \le \omega_i^{m_i}} \{u_i(\omega_i)\} = \max_{\omega_i \in \Omega_i} \left\{ \frac{\omega_i\ C_i(\omega_i)}{2\pi} \right\}. \tag{4}$$

Figure 2 graphically illustrates $C_i(\omega_i)$ for the sample task considered in Table I, while Figure 3 illustrates its actual utilization as a function of ω_i.

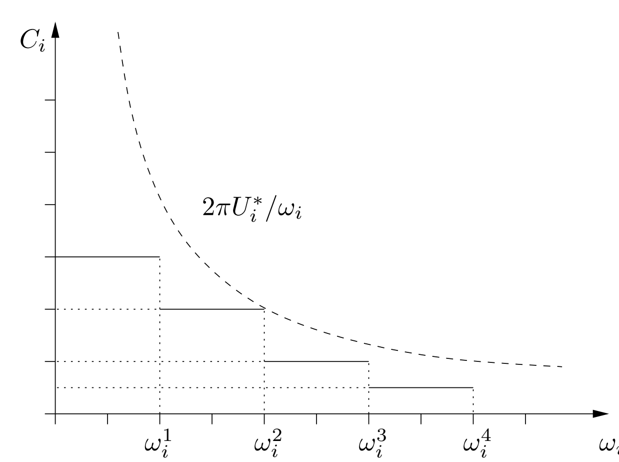

Figure 2. Task WCET as a function of the rotation speed ω_i.

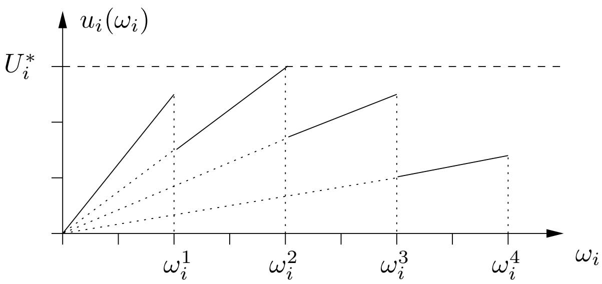

Figure 3. Task utilization as a function of the rotation speed ω_i.

It is worth observing that, by defining U_i^* as in Equation (4), the step function describing $C_i(\omega_i)$ lies entirely below the hyperbole $2\pi U_i^*/\omega_i$ (represented by the dashed curve in Figure 2).

Hence, a task set Γ is schedulable under EDF if and only if

$$\sum_{\tau_i \in \Gamma^P} U_i + \sum_{\tau_i^* \in \Gamma^R} U_i^* \leq 1. \tag{5}$$

where U_i^* is computed by Equation (4).

Note that, if state variables are arbitrary and independent, the test expressed in Equation (5) is necessary and sufficient, because the case in which all tasks experience their maximum utilization can actually occur in practice.

IV. Handling dynamic changes

This section analyzes the worst-case utilization of a rate-adaptive task in the case in which the rotation speed ω_i can change over time, but its acceleration is limited by a maximum value α_i. To explain the problem that may occur in this situation, we consider a generic rate-adaptive task τ_i^* activated at time t_0, when the crankshaft rotation angle is equal to θ_0 and its rotation speed is ω_0. For the sake of simplicity, and for the explanation of the problem, the index i is omitted from the variables ω and α. Let $\theta_1 = \theta_0 + \Delta\theta$ be the next angle at which τ_i^* is triggered again.

If the crankshaft rotation speed is constant and equal to ω_0, the rotation angle $\theta(t)$ will increase linearly as a function of time as

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

Thus, 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

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

However, if ω increases over time, the activation period will be shorter than $T_i(\omega_0)$. The situation is depicted in Figure 4.

To compute the shortest period $T_i'(\omega)$ at which τ_i^* can be activated when the rotation speed is allowed to change, we assume that $\omega(t)$ can increase at most with a maximum acceleration α:

$$\omega(t) = \omega_0 + \alpha(t - t_0).$$

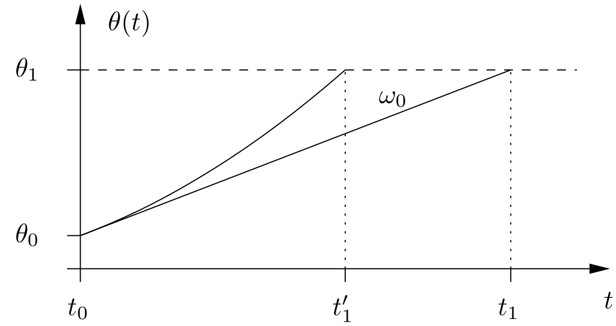

Figure 4. Shaft angle as a function of time.

As a consequence, the angle $\theta(t)$ will increase as

$$\theta(t) = \theta_0 + \int_{t_0}^{t} \omega(t)dt = \theta_0 + \omega_0(t - t_0) + \frac{\alpha}{2}(t - t_0)^2$$

and the value $\theta_1 = \theta_0 + \Delta\theta$ will be reached at time t_1' such that

$$\Delta\theta = \omega_0(t_1' - t_0) + \frac{\alpha}{2}(t_1' - t_0)^2.$$

Hence, for any given ω, the shortest activation period for τ_i^* can be computed as $T_i' = t_1' - t_0$ and it is such that:

$$\omega T_i' + \frac{\alpha}{2}(T_i')^2 = \Delta\theta.$$

Solving the equation above (and discarding the negative solution) we find:

$$T_i'(\omega, \alpha) = \frac{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega}{\alpha}. \tag{6}$$

Figure 5 illustrates T_i' as a function of ω for different values of α (rad/sec^2) and $\Delta\theta = 2\pi$.

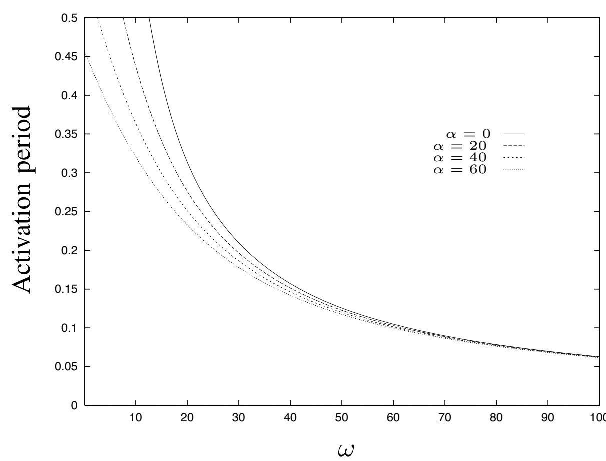

Figure 5. Period T_i' as a function of ω for different values of α (rad/sec^2) and $\Delta\theta = 2\pi$.

Note that $T_i'(\omega, 0) = T_i(\omega)$ and

$$\forall\omega > 0,\ \forall\alpha > 0 \quad T_i'(\omega, \alpha) < T_i(\omega).$$

However, the difference $T_i - T_i'$ is more significant for small rotation speeds, as also indicated by the first-order approximation of Equation (6):

$$T_i'(\omega, \alpha) = \frac{\Delta\theta}{\omega} - \frac{\Delta\theta^2}{2\omega^3}\alpha + o(\alpha). \tag{7}$$

In particular, for $\omega = 0$ we have $T_i'(0,\alpha) = \sqrt{2\Delta\theta/\alpha}$.

As a consequence, the actual utilization of a rate-adaptive task when the speed ω is not constant is

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

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

$$U_i^*(\alpha) = \max_{\omega\in\Omega_i} \left\{ \frac{\alpha\, C_i(\omega)}{\sqrt{\omega^2 + 2\alpha\Delta\theta} - \omega} \right\}. \tag{8}$$

Using this result in Equation (5), a task set Γ can be guaranteed by EDF under dynamic conditions. The schedulability test, however, becomes only sufficient, because Equation (8) pessimistically assumes that, in each mode, all jobs periods are shrunk by acceleration while keeping the same computation time of the first job. In practice, when a rate adaptive task switches to a new mode due to a positive acceleration, the next job will run with the computation time associated with the new mode. Hence, there can be cases in which the schedulability test is not satisfied, but the system is actually schedulable.

Equation (8) is also used in the next section to define for each rate-adaptive task τ_i^* the set switching speeds Ω_i such that its maximum utilization never exceeds a desired value U_i^d provided at design time.

V. Deriving the set of switching speeds

In Section IV we showed that, if ω is the rotation speed detected at the task activation time t, the actual period can be shorter than $\Delta\theta/\omega$, 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' that leads to U_i^d, that is

$$T_i' = \frac{C_i}{U_i^d}.$$

Inverting Equation (6), we can derive ω as a function of T_i':

$$\omega = \frac{\Delta\theta}{T_i'} - \frac{\alpha}{2} T_i'. \tag{9}$$

and imposing $T_i' = C_i/U_i^d$ in the equation above we get

$$\omega = \frac{\Delta\theta}{C_i} U_i^d - \frac{\alpha}{2}\frac{C_i}{U_i^d}. \tag{10}$$

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 (10) allows computing, for each computation time $C_i^{(k)}$, the maximum transition rate that guarantees not to exceed a desired utilization U_i^d:

$$\forall\, k = 1,\ldots,m \qquad \omega_i^{(k)} = \frac{\Delta\theta}{C_i^{(k)}} U_i^d - \frac{\alpha}{2}\frac{C_i^{(k)}}{U_i^d}. \tag{11}$$

It is worth observing that the first term of the right-hand side of Equation (11) 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 6 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 desired utilization 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:

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

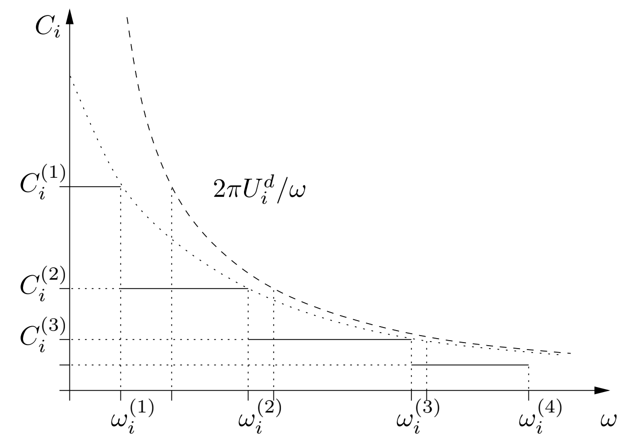

Figure 6. Maximum transition rates to keep a maximum utilization no larger than U_i^d.

Being $T_i(\omega) = \Delta\theta/\omega$, from Equation (11) we can also derive the minimum transition periods that guarantee not to exceed the desired utilization U_i^d:

$$T_i^{(k)} = \frac{2\Delta\theta\, C_i^{(k)}/U_i^d}{2\Delta\theta - \alpha\left(C_i^{(k)}/U_i^d\right)^2}. \tag{12}$$

Figure 7 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 dashed line represents the function $C_i(\omega) = U_i^d T_i$ corresponding to the steady state condition.

VI. Conclusions

In this paper we have shown how to model and analyze periodic tasks whose activation rates can vary with a system variable, while computation times can be adapted by the tasks themselves, by disabling/enabling a predetermined set of functions, to balance the overall workload.

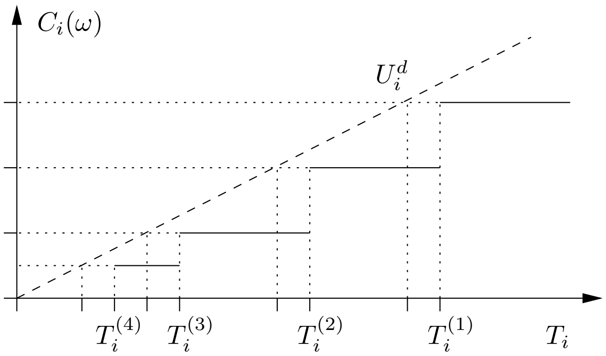

Figure 7. Minimum transition periods to keep a maximum utilization no larger than U_i^d.

In particular, exact schedulability analysis has been derived under EDF for the case in which relative deadlines are equal to periods, and we illustrated how to compute the worst-case utilization of such tasks under both steady-state conditions (where state variables are constant) and dynamic situations where system variables can change at a maximum bounded rate α.

Finally, given for each rate-adaptive task a set modes with known computation times, a design method has been presented to determine the set of safe transition speeds that keep the task utilization below a desired value.

As a future work, we plan to extend the schedulability analysis under fixed priorities and for tasks with relative deadlines different than periods.

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 speech 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] 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.

[5] 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.

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

[7] A. K. Mok and D. Chen, "A multiframe model for real-time tasks," *IEEE Transactions on Software Engineering*, vol. 23, no. 10, pp. 635–645, October 1997.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] L. Sha, R. Rajkumar, J. P. Lehoczky, and K. Ramamritham, "Mode change protocols for priority-driven preemptive scheduling," *Real-Time Systems*, vol. 1, no. 3, pp. 243–264, December 1989.

[15] 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.

[16] 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.

[17] J. Kim, K. Lakshmanan, and R. Rajkumar, "Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems," in *Proceedings of the Third IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS 2012)*, Beijing, China, April 17-19, 2012, pp. 28–38.

[18] 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.

[19] "Erika enterprise: an OSEK compliant real-time kernel." [Online]. Available: http://erika.tuxfamily.org/drupal/

[20] 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.