# Engine Control: Task Modeling and Analysis

Alessandro Biondi and Giorgio Buttazzo
Scuola Superiore Sant'Anna, Pisa, Italy
Email: alessandro.biondi@sssup.it, g.buttazzo@sssup.it

***Abstract*—Engine control is characterized by computational activities that are triggered by specific crankshaft rotation angles and are designed to adapt their functionality based on the angular velocity of the engine. Although a few models have been proposed in the literature to handle such tasks, most of them are quite simplistic and do not allow expressing features that are presently used by the automotive industry. This paper proposes a new task model for expressing realistic features of engine control tasks and presents a sufficient real-time analysis for applications consisting of multiple engine control tasks and classical periodic/sporadic tasks scheduled by EDF.**

## I. Introduction

Automotive applications include tasks that are activated with different mechanisms. Some engine control tasks are activated by a timer at a fixed rate (*periodic tasks*), whereas other tasks are linked to the rotation of the crankshaft and are activated at specific rotation angles (*angular tasks*) [1]. The activation rate of such angular tasks is therefore proportional to the engine speed: the higher the engine speed, the higher the activation rate.

A problem with such a type of activities is that, for high engine speeds, the system utilization can increase beyond a limit, generating an overload condition on the processor of the engine control unit (ECU) executing the application. 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 [2].

To prevent overload conditions, a common practice adopted in automotive applications is to implement angular tasks in such a way they automatically decrease their computational requirements (by changing their functionality) for increasing speeds [3]. To change their behavior, angular tasks are implemented as a set of execution modes, each operating within a specified range of rotation speeds. As a result, angular tasks have a variable activation rate (dependent on the engine speed) and a self-adaptive behavior implemented through a set of mode changes. For this reason, they are often referred to as *adaptive variable-rate tasks* (or AVR tasks). In this paper, the name AVR task is used as a synonym for angular task.

Given the peculiar characteristics of AVR tasks, the schedulability analysis of engine control applications cannot be addressed using classical real-time approaches. For instance, the elastic task model proposed by Buttazzo et al. [4], [5] is not suited to handle AVR tasks. In fact, in the elastic model, an overload condition is not handled by the task itself (by self-scaling its functionality), but through a global resource manager, which reduces the utilization of all the tasks by properly enlarging their periods. Similarly, classical mode change analysis [6], [7], [8] is not suited for engine control applications, because the activation rate of an AVR task changes continuously, thus an infinite number of modes would be required to describe all possible situations.

The problem of analyzing the real-time properties of engine control applications including AVR tasks has recently been considered in the literature by several authors, under different models and assumptions.

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 [9], 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. [10] presented a sufficient schedulability analysis under fixed priorities, assuming a constant angular velocity. The analysis is formulated using continuous intervals, hence it cannot be immediately translated into a practical schedulability test, whose complexity has not been evaluated.

The dynamic behavior of AVR tasks under fixed-priority scheduling has been analyzed by Davis et al. [11], who proposed a sufficient test based on an Integer Linear Programming (ILP) formulation. Besides being only sufficient, their approach is based on a quantization of the instantaneous crankshaft rotation speed, which may introduce additional pessimism in the analysis to guarantee the safety of the test.

The exact interference of an AVR task under fixed priorities has been analyzed by Biondi et al. [12]. Here, the interference is analyzed using a search approach in the speed domain, where the complexity is contained by deriving a set of dominant speeds, which also avoid quantizing the instantaneous speed considered in the analysis.

The analysis of a mixed set of classical and AVR tasks under Earliest Deadline First (EDF) scheduling has been addressed by Buttazzo, Bini, and Buttle [13], but for AVR tasks related to independent rotation sources. They also provided a design method that allows computing the set of switching speeds at which modes have to be changed to keep the overall utilization below a desired bound.

Although the results produced in the previous papers represent important milestones for the analysis of engine control systems, the task models used for the analysis are not always able to capture features that are currently adopted by the automotive industry in the implementation of AVR tasks. For example, in some work [13], tasks are considered to be linked

978-3-9815370-4-8/DATE15/©2015 EDAA

to independent rotation sources, while in reality all the angular tasks related to engine control are linked to the same rotation speed and may be triggered at different rotation angles.

A typical engine control application includes both classical periodic tasks (with period ranging from a few milliseconds up to 100 ms) and a number of angular tasks activated every single, half, and quarter engine revolution (as reported in [1], page 152). The assumption of independency clearly simplifies the analysis, but introduces an additional source of pessimism, considering situations that cannot actually occur when tasks are related to the same state variable. Also, all the previous papers consider mode transitions occurring at fixed speeds, while in practice, to avoid frequent mode transitions when the engine speed is close to a switching speed, mode changes are implemented with hysteresis; therefore, the speed at which an AVR task adapts its functionality is not the same in acceleration and deceleration.

*Contributions:* In this paper, we propose a new model for AVR tasks that allows expressing several realistic features of engine control systems. First, we consider multiple AVR tasks depending on the same rotation variable (the engine speed), but activated at different angles. Tasks can be specified to have different initial phases and angular periods and can be assigned an angular deadline less than or equal to the angular period.

Second, the speeds used by a task for switching its functionality are not the same in acceleration and deceleration. To avoid multiple transitions when the engine has a speed close to a switching speed, transitions are implemented with hysteresis.

Using the new task model, a schedulability analysis is proposed under EDF for AVR tasks with angular deadlines equal to angular periods. A reason for selecting EDF as a scheduler is that schedulability analysis of AVR tasks under fixed priorities is characterized by a very high computational complexity [11], [12].

Another important reason for using EDF in this context is that the interarrival period of an AVR task is subject to huge variations: engine speed typically ranges from 500 rotations per minute (rpm) to 6000 rpm, hence the interarrival period of an AVR task may vary from 120 ms to 10 ms, respectively (if the task is activated once per rotation). The consequence is that, in a task set consisting of multiple AVR tasks and normal periodic tasks, there will be several speeds at which any fixed priority assignment is far from being optimal, significantly penalizing the schedulability.

It is also worth observing that today EDF is actually available in a few operating systems (e.g., Erika Enterprise [14], which is also OSEK-compliant, and Linux, using the SCHED_DEADLINE scheduling class [15]).

*Paper structure:* The rest of the paper is organized as follows. Section II introduces the task model and the adopted notation. Section III illustrates a simple example of an AVR task. Section IV presents the schedulability analysis under EDF of a set of AVR tasks. The analysis is first presented without hysteresis, and later refined by taking hysteresis into account. Finally, Section V summarizes the results and concludes the paper highlighting some future research directions.

## II. Modeling

This section presents the models used for the rotation source and for the tasks.

### A. Rotation source model

For the purpose of the analysis, this paper considers a single rotation source (the engine) characterized by the following state variables:

- $\theta$ the current rotation angle of the crankshaft;
- $\omega$ the current angular speed of the crankshaft;
- $\alpha$ the current angular acceleration of the crankshaft.

We assume that the speed $\omega$ is limited within a given range $[\omega_{min}, \omega_{max}]$ and the acceleration $\alpha$ is limited within a given range $[\alpha^-, \alpha^+]$. Finally, for the purpose of the analysis, the acceleration is assumed to have a negligible variation during a full revolution of the engine.

### B. Task model

The computational activities considered in this paper are represented by a set of $n$ real-time preemptive tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$. Each task can be either a classical *periodic* task, activated at fixed time intervals, or an *angular task*, activated at specific crankshaft rotation angles. Since angular tasks have a variable interarrival time (inversely proportional to the engine speed $\omega$) and adapt their functionality for different speeds, they are also referred to as adaptive variable-rate (AVR) tasks. In the following, the subset of regular periodic tasks is denoted as $\Gamma_P$ and the subset of angular AVR tasks is denoted as $\Gamma_A$, so that $\Gamma = \Gamma_P \cup \Gamma_A$ and $\Gamma_P \cap \Gamma_A = \emptyset$. For the sake of clarity, whenever needed, a rate-adaptive task may also be denoted as $\tau_i^*$.

Both types of tasks are characterized by a worst-case execution time (WCET) $C_i$, an interarrival time (or period) $T_i$, and a relative deadline $D_i$. However, while for regular periodic tasks such parameters are fixed, for angular tasks they depend on the engine rotation speed $\omega$.

In particular, an angular task $\tau_i^*$ is characterized by an *angular period* $\Theta_i$ and an *angular phase* $\Phi_i$, so that it is activated at the following angles:

$$\theta_i = \Phi_i + k\Theta_i, \quad \text{for } k = 0, 1, 2, \ldots$$

This means that the period of a rate-adaptive task is inversely proportional to the engine speed $\omega$ and can be expressed as

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

An angular task $\tau_i^*$ is also characterized by a relative *angular deadline* $\Delta_i$ expressed as a fraction $\delta_i$ of the angular period ($\delta_i \in [0, 1]$). In the following, $\Delta_i = \delta_i\Theta_i$ represents the relative angular deadline.

Figure 3 graphically illustrates the parameters of an angular task, represented on a revolution. Note that the angular phase $\Phi_i$ is relative to a reference position called *Top Dead Center* (TDC) corresponding to the crankshaft angle for which the piston is at the highest position in the cylinder. In this paper, the TDC position is assumed to be at $\theta = 0$.

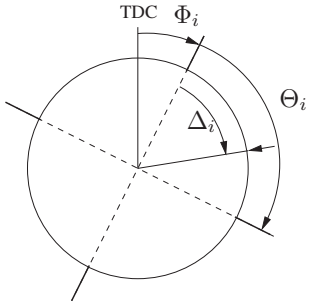


Figure 1. Parameters of an angular task, represented on a revolution.

When using EDF as a scheduler, an absolute deadline must be assigned to each job at its activation time in order to be scheduled. Although the angular deadline is constant, the temporal deadline is a function of $\omega$ and (for constant rotation speed) is equal to

$$D_i(\omega) = \frac{\Delta_i}{\omega} = \frac{\delta_i \Theta_i}{\omega} = \delta_i T_i(\omega). \quad (2)$$

However, note that for an incoming job, the next arrival time is not known, since $\omega$ may not be constant over time. To achieve a safe schedule, we make the conservative assumption to always assign each job the earliest possible deadline among those compatible with the speed at its activation, that is, the one derived assuming the maximum acceleration $\alpha^+$. A more precise deadline assignment could be achieved by considering the engine dynamics, but this is left to future work due to lack of space. In general, it is worth observing that the optimal deadline assignment for a job of an angular task requires clairvoyance to determine the exact interarrival period (which depends on the future engine evolution).

From the result presented in [13], the shortest activation interval an angular task can experience at a given speed $\omega$ is

$$T'_i(\omega) = \frac{\sqrt{\omega^2 + 2\Theta_i \alpha^+} - \omega}{\alpha^+}. \quad (3)$$

Replacing the value $T'_i(\omega)$ in Equation (2) we have:

$$D'_i(\omega) = \delta_i \frac{\sqrt{\omega^2 + 2\Theta_i \alpha^+} - \omega}{\alpha^+}, \quad (4)$$

which is the relative deadline assigned to an AVR task at its activation.

The execution time of an angular task $\tau_i^*$ is also a function of the rotation speed, since the task adapts its functionality to decrease its utilization at higher speeds. In the actual practice, angular tasks are implemented as a set of modes, each operating within a specified range of speeds; in addition, a hysteresis is introduced to avoid frequent mode changes when the engine speed is around to a switching speed. Therefore, the computation time of an angular tasks can be described by a step function consisting of $M_i$ execution modes, where each mode $m$ ($m = 1, \ldots, M_i$) is defined by a computation time $C_i^m$ and a speed range $[\omega_i^{m-}, \omega_i^{m+}]$.

Note that, when considering hysteresis in mode changes, the computation time of an AVR task depends not only on the value of $\omega$ but also on the current mode $m$, leading to two alternative mode-change behaviors, characterized by the following computation functions:

$$C_i^+(\omega) = C_i^m, \quad \forall \omega \in (\omega_i^{(m-1)+}, \omega_i^{m+}] \quad (5)$$

$$C_i^-(\omega) = C_i^m, \quad \forall \omega \in [\omega_i^{m-}, \omega_i^{(m+1)-}). \quad (6)$$

Similarly, the utilization of an AVR task with hysteresis (in steady-state conditions) can be expressed by the following two functions:

$$u_i^+(\omega) = \frac{C_i^+(\omega)}{T_i(\omega)} = \frac{\omega C_i^+(\omega)}{\Theta_i} \quad (7)$$

$$u_i^-(\omega) = \frac{C_i^-(\omega)}{T_i(\omega)} = \frac{\omega C_i^-(\omega)}{\Theta_i}. \quad (8)$$

In the following, to simplify the notation of the analysis with no hysteresis, we use $C_i(\omega)$ and $u_i(\omega)$ to denote $C_i^+(\omega)$ and $u_i^+(\omega)$, respectively.

## III. Example

Table I illustrates an example of an AVR task with three modes, specified for different (overlapping) speed intervals. Each mode $m$ executes a different function characterized by a computation time $C^m$ and mode transitions have a hysteresis of 500 rpm (for instance the transition from mode 1 to mode 2 occurs at 2000 rpm, whereas the reverse transition occurs at 1500 rpm).

| mode | $C^m$ | $[\omega^{m-}, \omega^{m+}]$ (rpm) | functionality |
|---|---|---|---|
| 1 | $C^1$ | [500, 2000] | `f1()` |
| 2 | $C^2$ | [1500, 4000] | `f2()` |
| 3 | $C^3$ | [3500, 6000] | `f3()` |

Table I. Example of an AVR task with three modes with different functionality.

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 [3], [13]. By defining an array of modes, however, an angular task can be efficiently implemented as illustrated in Figure 2. In this implementation, function `read_rotation_speed()` returns the instantaneous speed $\omega$ at the task activation time (not at the execution time of the function).

```
//Global variables
mode[M] = {f1(), f2(), f3()};
w_plus[M] = {2000, 4000, 6000};
w_minus[M] = {500, 1500, 3500};
m = 1;

task sample_angular_task {

    w = read_rotation_speed();

    while (w > w_plus[m]) m = m+1;
    while (w < w_minus[m]) m = m-1;
    execute(mode[m]);
}
```

Figure 2. Typical implementation of the AVR task shown in Table I.

Figure 3 graphically illustrates functions $C_i^-(\omega)$ and $C_i^+(\omega)$ for the AVR task shown in Table I, while Figure 4 illustrates the steady-state utilization functions $u_i^-(\omega)$ and $u_i^+(\omega)$.

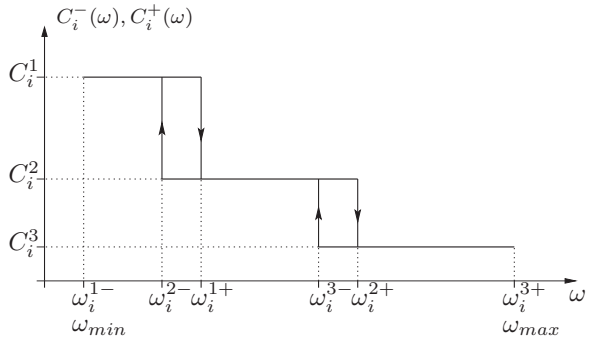


Figure 3. Task WCET as a function of the rotation speed $\omega$.

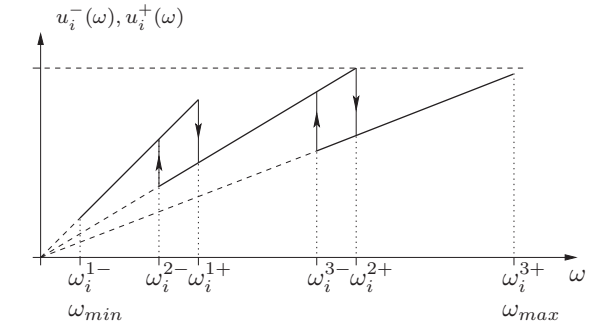


Figure 4. Task utilization in steady-state condition as a function of $\omega$.

## IV. Schedulability analysis

The objective of this section is to derive an upper bound $U_A$ for the overall utilization of a set $\Gamma_A$ of AVR tasks related to the same rotation source. In this way, a mixed set consisting of a subset $\Gamma_P$ of classical implicit-deadline periodic/sporadic tasks and a subset $\Gamma_A$ of AVR tasks can be schedulable under EDF if

$$U_P + U_A \leq 1 \tag{9}$$

where $U_P$ is the utilization of $\Gamma_P$. For the sake of clarity, we first present the schedulability analysis of AVR tasks under EDF without considering hysteresis into account, refining the results under hysteresis in Section IV-B. For the analysis we consider a set of synchronous angular tasks with implicit deadlines ($\forall \tau_i \in \Gamma_A$, $\Phi_i = 0$ and $\delta_i = 1$). Moreover, as true in many engine control applications, we assume that each angular period $\Theta_i$ is a submultiple of a full revolution, that is $\Theta_i = 2\pi/k$, for some positive integer $k$.

### A. Analysis with no hysteresis

As derived in [13], the utilization of an AVR-task under dynamic conditions (i.e., non-constant speed) is bounded by

$$u_i'(\omega) = \frac{C_i(\omega)}{T_i'(\omega)} = \frac{\alpha\, C_i(\omega)}{\sqrt{\omega^2 + 2\Theta_i\alpha^+} - \omega}. \tag{10}$$

The analysis presented in [13], however, assumes that all AVR tasks are triggered by independent rotation sources. As the following example shows, this assumption is pessimistic when applied to a set of AVR tasks related to the same rotation source.

*Example:* Consider two AVR tasks $\tau_1^*$ and $\tau_2^*$, both activated at the TDC ($\Phi_1 = \Phi_2 = 0$), with the same angular period $\Theta_1 = \Theta_2 = 2\pi$ and implicit angular deadlines ($\delta_i = 1$). Each task implements two modes and is triggered by the same rotation source. The task parameters are reported in Table II. Note that $\omega_1^{1-} = \omega_2^{1-}$ and $\omega_1^{2+} = \omega_2^{2+}$, since the same rotation source has clearly the same minimum and maximum speed. For this example, we consider an engine characterized by $\alpha^+ = -\alpha^- = 1.62 \cdot 10^4$ rev/ms$^2$.

| | | $C_i^m$ (ms) | $\omega_i^{m-}$ (rpm) | $\omega_i^{m+}$ (rpm) |
|---|---|---|---|---|
| $\tau_1$ | mode 1 | 2 | 500 | 2500 |
| | mode 2 | 1 | 2500 | 6500 |
| $\tau_2$ | mode 1 | 3 | 500 | 3500 |
| | mode 2 | 0.5 | 3500 | 6500 |

Table II. Task parameters for the example.

According to the schedulability test presented in [13], the maximum utilizations of the tasks taking acceleration into account are

$$U_1' = \max\left\{u_1'(\omega_1^{1+}), u_1'(\omega_1^{2+})\right\} = u_1'(\omega_1^{2+}) \approx 0.1084$$

$$U_2' = \max\left\{u_2'(\omega_2^{1+}), u_2'(\omega_2^{2+})\right\} = u_2'(\omega_2^{1+}) \approx 0.179.$$

Hence, the total utilization of the AVR-task set would be $U_A = U_1' + U_2' \approx 0.2874$.

Note that the value of $U_A$ is originated by the maximum utilization of $\tau_2$ at speed $\omega_2^{1+} = 3500$ rpm, plus the utilization of $\tau_1$ at speed $\omega_1^{2+} = 6500$ rpm. However, in practice, this scenario is actually impossible, because both tasks are triggered by the same rotation source.

To analyze a set of angular tasks related to the same rotation source, we first consider the simple case of tasks having the same angular period and then consider the general case of tasks with different periods.

First of all, observe that the variable $\omega$ in Equation (10) represents the instantaneous speed at the *activation* time of $\tau_i$. When all the AVR-tasks have the same angular period, their next activation will always occur at the same time, and hence at the same instantaneous speed. As a consequence, their utilizations can be added for each $\omega$ to obtain the overall task set utilization, so that $U_A = \max_\omega\{\sum_{\tau_i \in \Gamma_A} u_i'(\omega)\}$.

In the general case in which the AVR tasks have different angular periods, their activation may occur at different rotation speed due to engine acceleration or deceleration. For example, consider two AVR tasks $\tau_A$ and $\tau_B$ having $\Theta_A = 2\pi$ and $\Theta_B = \pi$, respectively. While $\tau_A$ has a single activation per revolution, $\tau_B$ has two activations per revolution. If $\widehat{\omega}$ is the instantaneous speed at the TDC, it is clear that while the first jobs of $\tau_A$ and $\tau_B$ are activated at the same speed $\widehat{\omega}$, the second job of $\tau_B$ can be released at different instantaneous speeds with respect to $\widehat{\omega}$, potentially increasing the utilization imposed in the revolution. In other words, for any given speed corresponding to the activation of $\tau_A$, a set of speeds has to be considered for $\tau_B$.

We address this issue by computing an upper-bound of the utilization of each AVR task in a full revolution ($\theta = 2\pi$), starting from the TDC at speed $\widehat{\omega}$. A full revolution is also the *angular hyperperiod* of the angular task set. Since $\Theta_i = 2\pi/k$, for some positive integer $k$, all the AVR tasks are synchronously activated at the TDC, which represents the *angular critical instant* of the task set, that is, the release scenario leading to the maximum workload in every time window.

Due to engine acceleration/deceleration, a job activated between two consecutive TDCs can be characterized by an interval of possible speeds at its activation. Such an interval increases with the angular shift from the TDC. Figure 5 graphically illustrates the situation for an AVR task $\tau_A$ with $\Theta_A = 2\pi$ and a generic AVR task $\tau_i$ with $\Theta_i < 2\pi$.

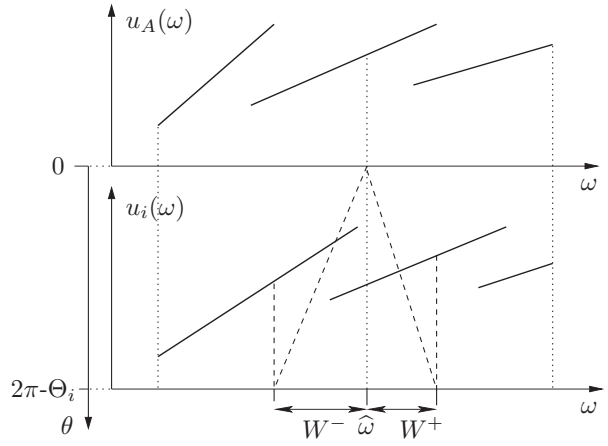


Figure 5. Interval of possible speeds to be considered at the activation of a task with $\Theta_i < 2\pi$.

Since such an interval increases with the angle from the TDC, the largest speed interval associated with $\tau_i$ is related to the last job activated during a revolution, that is the job activated at the angle $2\pi - \Theta_i$. The analytical expression of such a speed interval can be derived by computing the speed reachable from $\widehat{\omega}$ under a constant acceleration $\alpha$ in an angular space $\theta$, that is $\sqrt{\widehat{\omega}^2 + 2\theta\alpha}$.

Therefore, the set of speeds reachable from $\widehat{\omega}$ under acceleration $\alpha \in [0, \alpha^+]$ in an angular space $(2\pi - \Theta)$ is

$$W^+(\widehat{\omega}, \Theta) = \left\{ \omega \mid \omega \in \left[ \widehat{\omega}, \sqrt{\widehat{\omega}^2 + 2(2\pi - \Theta)\alpha^+} \right] \right\}. \quad (11)$$

Similarly, the set of speeds reachable in deceleration is given by:

$$W^-(\widehat{\omega}, \Theta) = \left\{ \omega \mid \omega \in \left[ \sqrt{\widehat{\omega}^2 + 2(2\pi - \Theta)\alpha^-}, \widehat{\omega} \right] \right\}. \quad (12)$$

The full range of speeds reachable from $\widehat{\omega}$ is then given by

$$W(\widehat{\omega}, \Theta) = W^-(\widehat{\omega}, \Theta) \cup W^+(\widehat{\omega}, \Theta). \quad (13)$$

Note that, for AVR tasks having $\Theta_i = 2\pi$, the set $W(\widehat{\omega}, \Theta)$ reduces to a single value, which is the instantaneous engine speed $\widehat{\omega}$ at the TDC.

Interval $W(\widehat{\omega}, \Theta)$ is used in the following theorem to derive an upper-bound of the utilization imposed by an AVR task during a revolution.

*Theorem 1:* The utilization of an AVR task $\tau_i$ within a single revolution is upper-bounded by

$$\widehat{u}_i(\widehat{\omega}) = \max_{\omega \in W(\widehat{\omega}, \Theta_i)} u'_i(\omega). \quad (14)$$

*Proof:* For each AVR task $\tau_i$, let $k_i = 2\pi/\Theta_i$ be the number of jobs of $\tau_i$ in a revolution, and let $J_i^1, \ldots, J_i^{k_i}$ be the jobs of $\tau_i$ in a revolution. Since we are considering dynamic conditions, the utilization of each job $J_i^\ell, \ell = 1, \ldots, k_i$ (as an upper-bound of its workload) is bounded by $u'_i(\omega^{(\ell)})$, where $\omega^{(\ell)}$ is the instantaneous engine speed at the activation of job $J_i^\ell$. Note that, since the absolute deadline of each $J_i^\ell$ is always assigned based on the maximum acceleration from speed $\omega^{(\ell)}$, $u'_i(\omega^{(\ell)})$ has to be used for computing the utilization upper bound of $J_i^\ell$ even when considering decelerations.

Due to engine acceleration/deceleration, all possible values of instantaneous speeds $\omega^{(\ell)}, \ell = 1, \ldots, k_i$ are included in the set $W(\widehat{\omega}, \Theta_i)$. Since an acceleration $\alpha \in [\alpha^-, \alpha^+]$ can always be found such that a particular $\omega^{(\ell)} \in W(\widehat{\omega}, \Theta_i)$ is the speed at the activation of a job $J_i^\ell$, the overall workload of all jobs $J_i^\ell, \ell = 1, \ldots, k_i$ can be upper-bounded by the maximum utilization $u'_i(\omega)$, with $\omega \in W(\widehat{\omega}, \Theta_i)$. ■

Using Theorem 1, an upper-bound of the utilization in a full revolution can be computed by summing the contribution of each AVR task, that is:

$$\widehat{U}(\widehat{\omega}) = \sum_{\tau_i \in \Gamma_A} \widehat{u}_i(\widehat{\omega}). \quad (15)$$

To cope with all possible scenarios determined by the different speeds $\widehat{\omega}$, an upper-bound $U_A$ on the total utilization of the set $\Gamma_A$ can be computed as:

$$U_A = \max_{\widehat{\omega}} \widehat{U}(\widehat{\omega}). \quad (16)$$

### B. Analysis with hysteresis

In this section, the utilization bound of Theorem 1 is refined by taking into account the hysteresis on mode changes.

As done for $u'_i(\omega)$, it is possible to derive a utilization bound taking into account the effect of the hysteresis in deceleration, that is

$$u'^-_i(\omega) = \frac{C_i^-(\omega)}{T'_i(\omega)} = \frac{\alpha\, C_i^-(\omega)}{\sqrt{\omega^2 + 2\Theta_i\alpha^+} - \omega}. \quad (17)$$

Note that, although considering decelerations in the analysis, all the jobs of an AVR task are assigned an absolute deadline based on $\alpha^+$, hence $T'_i(\omega)$ has to be considered in Equation (17).

To simplify the readability of the following theorem, we define a function $\mu_i(\omega, \widehat{\omega})$, which represents the worst-case utilization for $\tau_i$, activated at instantaneous speed $\omega$, where $\omega$ is reached with a deceleration from $\widehat{\omega}$.

$$\mu_i(\omega, \widehat{\omega}) = \begin{cases} u'^-_i(\omega) & \text{if } u'_i(\widehat{\omega}) = u'^-_i(\widehat{\omega}) \\ u'_i(\omega) & \text{otherwise.} \end{cases} \quad (18)$$

Further details about the conditional nature of this functions are reported in the proof of the following theorem.

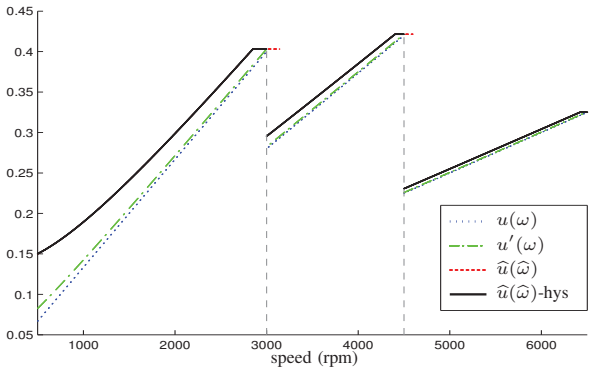


Figure 6. Comparison of the utilization bounds defined in the paper.

*Theorem 2:* The utilization of an AVR task $\tau_i$ within a single revolution during acceleration is upper-bounded by

$$\widehat{u}_i^{+}(\widehat{\omega}) = \max_{\omega \in W^{+}(\widehat{\omega},\Theta_i)} u'_i(\omega) \tag{19}$$

and during deceleration is upper-bounded by

$$\widehat{u}_i^{-}(\widehat{\omega}) = \max_{\omega \in W^{-}(\widehat{\omega},\Theta_i)} \mu_i(\omega,\widehat{\omega}). \tag{20}$$

*Proof:* The same considerations expressed in the proof of Theorem 1 holds in the case of acceleration during the revolution. When decelerations are considered, two scenarios can occur. If the revolution starts from a speed $\widehat{\omega}$ and $u'_i(\widehat{\omega}) > u'^{-}_i(\widehat{\omega})$, since nothing is assumed about the "previous history" of the engine speed, it is not possible to infer on the current state of hysteresis. Hence, a safe bound is computed by considering $u'_i(\widehat{\omega})$, since $\forall \omega \in [\omega_{min}, \omega_{max}] u'_i(\omega) \geq u'^{-}_i(\omega)$. In the other case in which $u'_i(\widehat{\omega}) = u'^{-}_i(\widehat{\omega})$, since we are assuming a deceleration, we can safely use function $u'^{-}_i(\widehat{\omega})$ exploiting the effect of hysteresis. ■

Using Theorem 2, an upper-bound of the utilization in a full revolution can be computed as

$$\widehat{U}(\widehat{\omega}) = \max\left\{ \sum_{\tau_i \in \Gamma_A} \widehat{u}_i^{+}(\widehat{\omega}), \sum_{\tau_i \in \Gamma_A} \widehat{u}_i^{-}(\widehat{\omega}) \right\}. \tag{21}$$

## V. Summary and Conclusions

In this paper we presented a new task model suited for describing and analyzing the temporal behavior of computational activities typically used in engine control applications, where task activations are linked to specific rotation angles of the engine. In addition, such tasks vary their functionality to adapt their computation load at different engine speeds. The proposed model includes the possibility of specifying several features that are commonly adopted in the implementation of such activities, as angular phases, angular periods, angular deadlines, and hysteresis in mode transitions. A conservative schedulability analysis has been derived under EDF for a set of synchronous angular tasks with angular deadlines equal angular periods and related to the same rotation source.

The analysis proposed in this paper has been developed based on a number of utilization functions, each derived under specific situations. In particular, $u_i(\omega)$ represents the utilization of an AVR task in steady-state conditions, $u'_i(\omega)$ is the one inflated by the effect of acceleration, and $\widehat{u}_i(\widehat{\omega})$, which represents the novel contribution of this work, takes into account the workload generated by all the jobs activated within a revolution, without and with hysteresis.

For comparison purpose, all these functions are plotted in Figure 6 for a sample AVR task $\tau$ with three modes and angular period $\Theta = \pi/2$. Note that the curve denoted as $\widehat{u}(\widehat{\omega})$-hys is computed by Equation (21) for $\Gamma_A = \tau$. As already observed in [13], the plots show that the effect of acceleration is more significant at lower speeds and is negligible at higher speeds. Also note that $\widehat{u}(\widehat{\omega})$ with and without hysteresis almost always overlap, except in intervals across the switching speeds, where $\widehat{u}(\widehat{\omega})$-his $< \widehat{u}(\widehat{\omega})$.

As a future work, we plan to extend the schedulability analysis for task sets with arbitrary deadlines and different phases, and improve the deadline assignment taking engine dynamics into account to further reduce the utilization bound.

## References


[1] L. Guzzella and C. H. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Springer-Verlag, 2010.

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

[3] D. Buttle, "Real-time in the prime-time," in *Keynote speech at the 24th Euromicro Conference on Real-Time Systems*, Pisa, Italy, July 12, 2012.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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 *Proc. of the Design, Automation and Test Conference in Europe*, Grenoble, France, March 18-22, 2013.

[11] R. I. Davis, T. Feld, V. Pollex, and F. Slomka, "Schedulability tests for tasks with variable rate-dependent behaviour under fixed priority scheduling," in *Proc. 20th IEEE Real-Time and Embedded Technology and Applications Symposium*, Berlin, Germany, April 15-17, 2014.

[12] 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)*, Madrid, Spain, July 8-11, 2014.

[13] G. Buttazzo, E. Bini, and D. Buttle, "Rate-adaptive tasks: Model, analysis, and design issues," in *Proc. of the Int. Conference on Design, Automation and Test in Europe*, Dresden, Germany, March 24-28, 2014.

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

[15] D. Faggioli, F. Checconi, M. Trimarchi, and C. Scordino, "An edf scheduling class for the linux kernel," in *Proc. of the 11th Real-Time Linux Workshop (RTLWS)*, Dresden, Germany, September 28-30, 2009.