Modeling and Analysis of Engine Control Tasks Under Dynamic Priority Scheduling

Alessandro Biondi and Giorgio Buttazzo
Scuola Superiore Sant'Anna, Pisa, Italy
Email: {alessandro.biondi, giorgio.buttazzo}@santannapisa.it

***Abstract*—In automotive systems, engine control applications include computational activities that are triggered by specific rotation angles of the crankshaft, causing their activation rate to be proportional to the engine speed. In order to avoid overloads at high engine speeds, these tasks are implemented to adapt their functionality based on the angular velocity of the engine. This paper proposes a task model for expressing a number of realistic features of such engine control tasks and presents a real-time schedulability analysis for applications consisting of multiple engine control tasks and classical periodic/sporadic tasks scheduled by the Earliest Deadline First algorithm. Differently from other efforts spent in analyzing engine-control applications, the presented approach is focused on simplicity, providing linear-time and quadratic-time schedulability tests based on utilization bounds. Experimental results are finally presented to assess the performance of the presented analysis techniques.**

I. Introduction

Typical engine control software consists of different kinds of computational activities: *periodic tasks*, activated by a timer at fixed time intervals, and *angular tasks*, triggered at predefined rotation angles of the crankshaft [1]. Hence, angular tasks generate a dynamic workload that strictly depends on the engine speed: the higher the speed, the higher the activation rate.

The major problem with the angular tasks is that, at high engine speeds, they are activated very frequently, thus generating a computational demand that may overload the processor of the *electronic control unit* (ECU). If not properly handled, this phenomenon could introduce large delays in control actions, causing a significant performance degradation of one or more control functions [2].

To prevent such overload conditions, a common solution adopted in automotive applications is to make angular tasks *adaptive*, so that their computational demand can be reduced at high speeds by disabling some functionality or switching to simpler control algorithms [3]. This is typically done by defining a set of execution modes, each operating within a given speed range. This approach is also compatible with the engine dynamics: in fact at high speeds the engine is more stable and hence simpler control algorithms can be applied.

The timing properties of the control software determine the performance of the engine with respect to several indexes like power, fuel efficiency, and pollutant generation [1], [4]. Analyzing and understanding the timing behavior of such software systems is therefore of paramount importance for designing and validating engine control systems.

The peculiarity of engine control software makes classical approaches developed by the real-time systems community ineffective for analyzing its timing properties with a reasonable degree of accuracy. Their variable rate of activation, together with their speed-dependent adaptive behavior, originate considerable challenges that cannot be solved by classical techniques without incurring in excessive pessimism. Furthermore, considering that the timing behavior of angular tasks is strictly dependent on the engine dynamics (specifically, the rotation of the crankshaft), the physical constraints of the system play a key role and cannot be neglected in the analysis. All these facts determine the need for novel analysis techniques, which represent important building blocks for developing new and accurate design methodologies for engine control software.

Previous works [5], [6] identified that the exact analysis of engine control applications requires the adoption of convoluted techniques (with respect to the most popular analysis approaches), which are complex both in terms of computational cost and human intuition. Motivated by this fact, this paper aims at taking another look at the analysis of engine control applications, focusing on *simplicity* as a primary objective.

Paper contributions. This work presents linear-time and quadratic-time schedulability tests for engine-control applications under *earliest-deadline first* (EDF) scheduling. The tests are based on *utilization bounds* that explicitly account for the physical constraints of the considered systems. The proposed analysis techniques are supported by a task model that allows expressing a number of features that are part of realistic designs of engine-control applications. Finally, the paper reports on an experimental study that has been conducted to evaluate the performance of the proposed approaches.

Paper structure. Section II presents the system model and some background. Section III proposes an utilization bound for an AVR task under dynamic conditions. Section IV proposes an analysis based on utilization bounds for a set of AVR tasks activated by the same rotation source. Section V reports some experimental results carried out to evaluate the proposed approaches. Section VI discusses the related work and Section VII concludes the paper.

II. System Model and Background

In this work, the rotation source (the crankshaft of the engine) will be characterized by the following state variables: θ the current rotation angle of the crankshaft; ω the current angular speed of the crankshaft; and α the current angular acceleration of the crankshaft. We also assume that the speed ω is limited within a given range $[\omega_{min}, \omega_{max}]$ and the acceleration α is limited within a given range $[\alpha^{-}, \alpha^{+}]$, as also true for real-world engines.

A. Task model

The considered engine-control applications consist of a set $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$ of n real-time preemptive tasks. Each task can be either *periodic* (i.e., activated at fixed time intervals), *sporadic* (i.e., activated with a minimum inter-arrival time) or an *angular task* (i.e., activated at specific crankshaft rotation angles). Since angular tasks adapt their workload for different speeds, they are also referred to as *adaptive variable-rate* (AVR) tasks. In the following, the subset of regular periodic/sporadic tasks is denoted as Γ_P and the subset of angular AVR tasks is denoted as Γ_A, so that $\Gamma = \Gamma_P \cup \Gamma_A$ and $\Gamma_P \cap \Gamma_A = \emptyset$. The overall utilization of Γ_P is denoted as U_P. For the sake of clarity, whenever needed, an AVR task may also be denoted as τ_i^*. Both types of tasks generate an infinite sequence of *jobs* (i.e., task instances).

Both types of tasks are characterized by a *worst-case execution time* (WCET) C_i, an inter-arrival time (or period) T_i, and a relative deadline D_i. However, while for regular periodic/sporadic tasks such parameters are fixed, for angular tasks they depend on the engine rotation speed ω. In particular, an angular task τ_i^* is characterized by an *angular period* Θ_i and an *angular phase* Φ_i, so that its jobs are activated at the following angles:

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

This means that, when the engine is rotating at a fixed speed ω, the inter-arrival time of an AVR task is *inversely proportional* to the engine speed and can be expressed as $T_i(\omega) = \Theta_i/\omega$. The angular phase Φ_i is relative to a reference position called *Top Dead Center* (TDC) corresponding to the crankshaft angle for which at least one piston is at the highest position in its cylinder. Without loss of generality, the TDC position is assumed to be at $\theta = 0$. An angular task τ_i^* is also characterized by a relative *angular deadline* Δ_i expressed as a fraction δ_i of the angular period ($\delta_i \in [0, 1]$). In the following, $\Delta_i = \delta_i\Theta_i$ represents the relative angular deadline.

For the purpose of analyzing the timing properties of engine control applications that include AVR tasks, it is crucial to characterize the relation between the AVR task parameters and the dynamics of the crankshaft.

Suppose that a job $J_{i,k}$ of an AVR task τ_i^* is released at time t_k with instantaneous engine speed ω_k. Following standard physical equations (e.g., as described in [7]), the release time t_{k+1} of the next AVR job $J_{i,k+1}$ assuming constant acceleration α_k during $(t_k, t_{k+1}]$ can be computed as $t_{k+1} = t_k + T_i(\omega_k, \alpha_k)$, where

$$T_i(\omega_k, \alpha_k) = \frac{\sqrt{\omega_k^2 + 2\Theta_i\alpha_k} - \omega_k}{\alpha_k}. \tag{1}$$

Given the acceleration bounds α^+ and α^-, the above equation allows computing the *minimum* and the *maximum* inter-arrival times that an AVR task τ_i^* can experience after the release of a job at a given speed ω, which are $T_i(\omega, \alpha^+)$ and $T_i(\omega, \alpha^-)$, respectively. In a similar way, the instantaneous engine speed $\omega_{k+1} = \Omega(\omega_k, \alpha_k)$ at the release of the next job $J_{i,k+1}$ can be computed as $\omega_k + \alpha_k T_i(\omega_k, \alpha_k)$, which gives:

$$\Omega_i(\omega_k, \alpha_k) = \sqrt{\omega_k^2 + 2\Theta_i\alpha_k}. \tag{2}$$

Mode change. An AVR task τ_i^* is typically implemented [3] as a set $\mathcal{M}_i$ of M_i execution modes with decreasing functionality, each operating in a predetermined range of rotation speeds. Mode m of an AVR task τ_i^* is characterized by a WCET C_i^m and is valid in a speed range $(\omega_i^{m+1}, \omega_i^m]$, where $\omega_i^{M_i+1} = \omega^{min}$ and $\omega_i^1 = \omega^{max}$. Hence, the set of modes of task τ_i^* can be expressed as $\mathcal{M}_i = \{(C_i^m, \omega_i^m), m = 1, 2, \ldots, M_i\}$.

The WCET $C_{i,k}$ of an arbitrary AVR job $J_{i,k}$ can be expressed as a non-increasing step function $\mathcal{C}_i(\omega)$ of the instantaneous speed ω *at its release*, that is,

$$C_{i,k} = \mathcal{C}_i(\omega) \in \{C_i^1, \ldots, C_i^{M_i}\}. \tag{3}$$

The implementation of AVR tasks can be performed as a sequence of conditional `if` statements, each executing a specific subset of functions [3], [7] (also denoted as *runnables* in the automotive domain).

B. EDF scheduling of AVR tasks

Note that, in engine control applications, the rate-monotonic priority assignment (which assigns higher priority to tasks having higher rate) has little sense, because AVR tasks are, by definition, activated at a variable rate.

For example, consider an AVR task with angular period $\Theta = 2\pi$ and a typical production car, where engine speed ranges from $\omega^{min} = 500$ RPM to $\omega^{max} = 6500$ RPM, leading to inter-arrival times from $T^{min} \approx 10\ ms$ to $T^{max} = 120\ ms$. Since generally engine control applications include periodic tasks with periods in the latter range [1], any fixed priority assignment (including Rate Monotonic) will be not optimal for some engine speeds. The situation gets worse when considering more than one AVR task.

This motivates the investigation of different priority assignments that take engine speed into account to support AVR tasks. The EDF scheduling algorithm assigns dynamic (job-level) priorities to tasks and is known to be optimal on uniprocessor systems [8]. Under EDF, an absolute deadline d must be assigned to each job at its release time t to be scheduled. For standard periodic tasks, the absolute deadline is computed as $d = t + D$, where D is the relative task deadline. For an AVR task, such a rule must be adapted, since the relative deadline is not constant, but depends on the dynamics of the rotation source triggering the task. In particular, the relative deadline is a function of the engine state at the release of a job and the *future* evolution of the rotation source in terms of acceleration, as it is illustrated in Figure 1. This reasoning brings to the conclusion that computing the *exact* relative deadline of an upcoming job of an AVR task requires clairvoyance, thus preventing optimality even under EDF.

Nevertheless, it is possible to achieve a safe schedule by assigning each job the earliest deadline among those that are compatible with the instantaneous speed at the job activation (i.e., the one derived assuming the maximum acceleration α^+ from the task release on). Using such a rule, the relative deadline of an AVR task τ_i^* released at the instantaneous engine speed ω results to be

$$D_i(\omega) = \frac{\sqrt{\omega^2 + 2\Delta_i\alpha^+} - \omega}{\alpha^+}. \tag{4}$$

The above equation is obtained as a special case of Equation (1). Please note that since EDF is a job-level fixed-

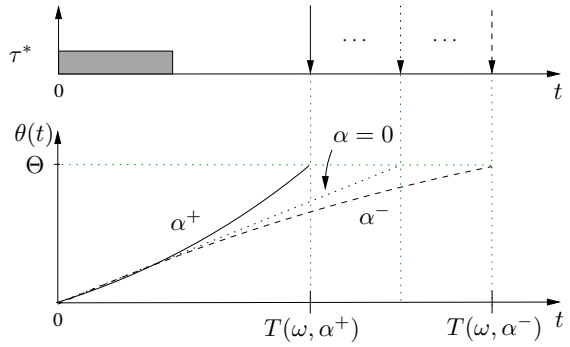

Figure 1. Possible deadlines of an AVR job activated at speed ω.

priority scheduling policy, such a deadline will be *fixed* for the whole execution of the upcoming job. Biondi et al. [5] showed that, with this deadline assignment, EDF scheduling can achieve near-optimal performance in scheduling engine-control applications. Guo and Baruah [9] confirmed this finding from a more theoretical perspective, showing that the speed-up bound of EDF in the presence of AVR tasks is close to one under realistic acceleration bounds.

III. Utilization Bound of an AVR Task

According to a well-established result [10], a task set composed of implicit-deadline periodic/sporadic tasks is schedulable under EDF *if and only if* the sum of the utilizations of all the tasks does not exceed one. Besides being exact, this test is very simple, very fast to execute (linear-time complexity), and therefore useful for several practical purposes, including the development of efficient design methodologies.

Unfortunately, this powerful result cannot be directly applied to engine-control applications. One may still treat engine-triggered tasks as sporadic tasks with WCET equal to the largest WCETs of their modes and inter-arrival time equal to the one corresponding to the highest engine speed: however, this approach generally leads to excessive pessimism and cannot be considered as a realistic solution to analyze engine-control applications (interested readers can refer to [11], Chapter 10). However, considering the simplicity of utilization-based tests, it would be good to derive a similar result to analyze engine-control applications without incurring in excessive pessimism, hence *explicitly* considering both mode-changes and physical constraints in the derivation of the test.

Once this bound is obtained, a mixed task set consisting of **(i)** a subset Γ_P of classical implicit-deadline periodic/sporadic tasks and **(ii)** a subset Γ_A of AVR tasks with implicit angular deadline can be deemed schedulable under EDF if

$$U_P + U_A \leq 1 \tag{5}$$

where U_P is the utilization of Γ_P and U_A is the utilization bound of Γ_A.

When looking at steady-state conditions, i.e., assuming a fixed speed ω, the utilization of an AVR task τ_i^* can simply be computed as the ratio between the WCET of τ_i^* at speed ω and the corresponding inter-arrival time Θ_i/ω, that is

$$u_i(\omega) = \frac{C_i(\omega)\Theta_i}{\omega}. \tag{6}$$

Indeed, for a fixed speed, an AVR task behaves as a standard periodic task. An example of function $u_i(\omega)$ is illustrated in Figure 2 (solid line) for an AVR task with three modes. Being all speeds $\omega \in [\omega^{min}, \omega^{max}]$ valid engine speeds, the maximum steady-state utilization can be computed as

$$U_i = \max_{\omega \in [\omega^{min}, \omega^{max}]} \{u_i(\omega)\},$$

which, due to the monotonicity of function $C_i(\omega)$, can be computed in a closed-form as

$$U_i = \max_{m=1,\ldots,M_i} \{u_i(\omega^m)\}. \tag{7}$$

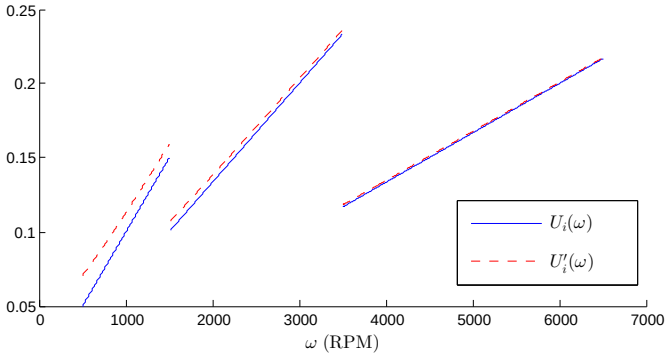

Figure 2. Utilization bounds for an AVR task with three modes $\{C^1 = 1\ ms, \omega^1 = \omega^{max} = 6500$ RPM$\}$,$\{C^1 = 2\ ms, \omega^2 = 3500$ RPM$\}$ and $\{C^3 = 3\ ms, \omega^3 = 1500$ RPM$\}$, computed in steady-state (solid line) and dynamic (dashed line) conditions.

However, when considering dynamic conditions (i.e., in the presence of acceleration), function $u_i(\omega)$ does not provide a safe bound for being used in an utilization-based schedulability test like Equation (5). In fact, accelerations of the engine can determine an instantaneous utilization peak that is higher than U_i. To better understand this phenomenon, consider a job $J_{i,k}$ of an AVR task τ_i^* that is released at an arbitrary speed ω. Such a job can execute for at most $C_i(\omega)$ time units. If, immediately after the release of $J_{i,k}$, the engine accelerates with maximum acceleration α^+ for an angular distance of Θ_i degrees, then the inter-arrival time to the next job $J_{i,k+1}$ results $T_i(\omega, \alpha^+)$ time units, which is clearly *shorter* than the inter-arrival time of the job in steady-state conditions (equal to Θ_i/ω time units). As a consequence, under dynamic conditions, AVR tasks can generate *utilization peaks* that are higher than their maximum computational load demanded at fixed speed: hence function $u_i(\omega)$ does not provide a safe utilization bound.

To cast this phenomenon to an example, consider a simple AVR task τ_i with two modes $\mathcal{M}_i = \{(5\ \text{ms}, 3000\ \text{RPM}), (2\ \text{ms}, 6000\ \text{RPM})\}$[1] and $\Theta_i = 2\pi$. Suppose also that the maximum acceleration is able to reduce the inter-arrival time at 3000 RPM from 20 ms (fixed speed) to 15 ms. While the maximum steady-state utilization of τ_i is equal to $5/20 = 0.25$, dynamic conditions can lead to an instantaneous load equal to $5/15 = 0.3$, which occurs when the engine accelerates with maximum acceleration after the release of a job at 3000 RPM.

To overcome this issue, a conservative bound can easily

[1] In pure physical units, 3000 RPM and 6000 RPM correspond to about 314.159 rad/s and 628.319 rad/s, respectively.

be obtained (see [7]) by considering the reduction of the inter-arrival time due to the effect of the acceleration. This reasoning leads to the definition of a new utilization bound, that is

$$u'_i(\omega) = \frac{C_i(\omega)}{T_i(\omega, \alpha^+)} \tag{8}$$

represented by the dashed line in Figure 2. Note that this bound is valid for any $\Theta_i > 0$. For any job $J_{i,k}$ of τ_i^* released at speed ω, function $u'_i(\omega)$ provides a safe bound on the maximum instantaneous processor load that $J_{i,k}$ can demand. Similarly as argued above for Equation (7), the overall bound U'_i on the maximum utilization of τ_i^* can be computed as

$$U'_i = \max_{m=1,\ldots,M_i} \{u'_i(\omega^m)\}. \tag{9}$$

This bound can finally be used to construct a safe schedulability test by simply computing $U_A = \sum_{\tau_i^* \in \Gamma_A} \{U'_i\}$, which can then be used as an utilization bound in Equation (5).

The resulting test has a *linear-time complexity* both as a function of the number of tasks and the number of modes of the AVR tasks. Please note that, differently from the case of sporadic/periodic tasks where the utilization of a task set can be used to implement an exact schedulability test, the bound U'_i for AVR tasks allows deriving a *sufficient* analysis only. Intuitively, this can be concluded with the following reasoning. Let m' be the particular mode for which $U'_i = u'_i(\omega^{m'})$. This means that the maximum instantaneous utilization of τ_i^* is generated when the task releases a job J_k at speed $\omega^{m'}$ and, immediately after, the engine accelerates with maximum acceleration α^+, thus releasing the next job J_{k+1} after $T(\omega^{m'}, \alpha^+)$ time units. However, this scenario may not happen again for the next job: J_{k+1} is (by construction) released at a speed higher than $\omega^{m'}$ but in a mode different than m' that must have WCET $C'' < C_i(\omega^{m'})$ (being $\omega^{m'}$ a switching speed), hence it is possible that $C''/T(\Omega(\omega^{m'}, \alpha^+), \alpha^+) < U'_i$. In other words, τ_i^* may not be able to generate a constant computational load equal to U'_i in the long run.

IV. Utilization Bound of Multiple AVR Tasks Activated by the Same Rotation Source

Typical engine control applications include multiple AVR tasks with different angular periods activated by the rotation of the crankshaft (e.g., see [1]). Therefore, this section considers a set of synchronous angular tasks with implicit angular deadlines ($\forall \tau_i^* \in \Gamma_A$, $\Phi_i = 0$ and $\delta_i = 1$) that are activated by the same rotation source (i.e., the same crankshaft). Moreover, as true in many engine control applications, we assume that each angular period Θ_i is a submultiple of a full crankshaft revolution, that is $\Theta_i = 2\pi/q$, for some positive integer q.

The utilization bound provided in the previous section can still be used to analyze such task sets: however, being the AVR tasks not independent (as they are activated by the same rotation source), this approach can easily result in an over-pessimistic analysis.

A. Analysis

Observe that variable ω in Equation (8) denotes the instantaneous speed at the *activation* time of τ_i. Also note that if all the AVR tasks have the same angular period, their are always activated at the same time, at the same instantaneous speed. Hence, the overall task set utilization can be computed by summing their utilization bounds for each ω, having

$$U_A = \max_{\omega \in [\omega^{min}, \omega^{max}]} \left\{ \sum_{\tau_i \in \Gamma_A} u'_i(\omega) \right\}.$$

When AVR tasks have different angular periods, they can be activated at different rotation speeds due to engine acceleration or deceleration. For instance, consider the case of two AVR tasks τ_A and τ_B with angular period $\Theta_A = 2\pi$ and $\Theta_B = \pi$, respectively. While τ_A has only a single activation per revolution of the rotation source, τ_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 τ_A and τ_B are activated at the same speed $\widehat{\omega}$, the second job of τ_B can be released at different instantaneous speeds with respect to $\widehat{\omega}$. As a consequence, the overall processor load generated by τ_A and τ_B within a revolution can be higher than $u'_A(\widehat{\omega}) + u'_B(\widehat{\omega})$.

To solve this problem, an upper-bound of the utilization of each AVR task is computed within 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.

Since the engine is subject to accelerations or decelerations, an AVR job activated between two consecutive TDCs can be characterized by an *interval* of possible speeds at its activation. This interval increases with the angular shift from the TDC: the higher the shift the larger the interval. Figure 3 illustrates the situation for an AVR task τ_A with $\Theta_A = 2\pi$ and a generic AVR task τ_i with $\Theta_i < 2\pi$.

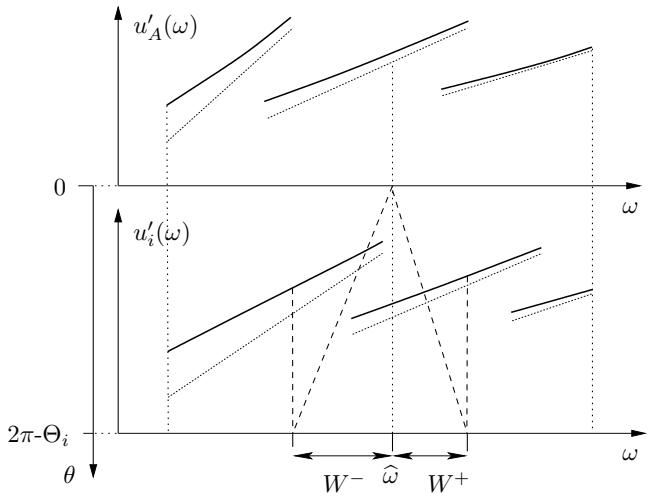

Figure 3. Interval of possible speeds to be considered at the activation of a task with $\Theta_i < 2\pi$. The interval increases as $2\pi - \Theta_i$ increases. The proposed approach aims at identifying the maximum utilization $u'_i(\omega)$ within such an interval.

Because this interval increases with the angle from the TDC, the largest speed interval associated with τ_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 maximum (resp., minimum) speed reachable from $\widehat{\omega}$ under maximum

acceleration (resp., deceleration) α in an angular space θ, that is $\sqrt{\widehat{\omega}^2 + 2\theta\alpha}$ as reported in Equation (2).

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

$$W^+(\widehat{\omega}, \Theta) = \left\{\omega \mid \omega \in \left[\widehat{\omega}, \sqrt{\widehat{\omega}^2 + 2(2\pi - \Theta)\alpha^+}\right]\right\}. \quad (10)$$

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

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

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

$$W(\widehat{\omega}, \Theta) = W^-(\widehat{\omega}, \Theta) \cup W^+(\widehat{\omega}, \Theta). \quad (12)$$

It is worth observing that for AVR tasks with angular period $\Theta_i = 2\pi$, 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 τ_i within a single revolution started at speed $\widehat{\omega}$ is upper-bounded by

$$\widehat{u}_i(\widehat{\omega}) = \max_{\omega \in W(\widehat{\omega}, \Theta_i)} u'_i(\omega). \quad (13)$$

Proof: Let $J_i^1, \ldots, J_i^{k_i}$ be the jobs of τ_i in a revolution, with $k_i = 2\pi/\Theta_i$. 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^ℓ. Note that, since the absolute deadline of each J_i^ℓ 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^ℓ even when considering decelerations.

According to the physical constraints determined by the acceleration bounds of the engine, all possible values of instantaneous speeds $\omega^{(\ell)}, \ell = 1, \ldots, k_i$ are included in the set $W(\widehat{\omega}, \Theta_i)$. Hence the overall processor load requested by all jobs $J_i^\ell, \ell = 1, \ldots, k_i$ can be upper-bounded by the maximum utilization $u'_i(\omega)$, with $\omega \in W(\widehat{\omega}, \Theta_i)$. ∎

An example of function $\widehat{u}_i(\widehat{\omega})$ is illustrated in Figure 4.

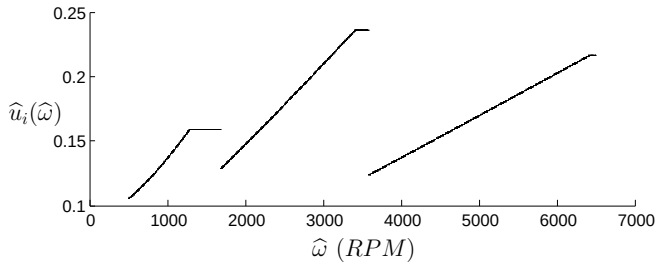

Figure 4. Example of function $\widehat{u}_i(\widehat{\omega})$ for the same AVR task of Figure 2.

Using Theorem 1, an upper-bound of the utilization in a full revolution beginning at speed $\widehat{\omega}$ can be computed by adding up the contribution of each AVR task, that is:

$$\widehat{U}(\widehat{\omega}) = \sum_{\tau_i^* \in \Gamma_A} \widehat{u}_i(\widehat{\omega}). \quad (14)$$

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 Γ_A can be computed as:

$$U_A = \max_{\widehat{\omega} \in [\omega^{min}, \omega^{max}]} \widehat{U}(\widehat{\omega}). \quad (15)$$

An example of function $\widehat{U}(\widehat{\omega})$ is illustrated in Figure 5, together with the corresponding utilization bound U_A.

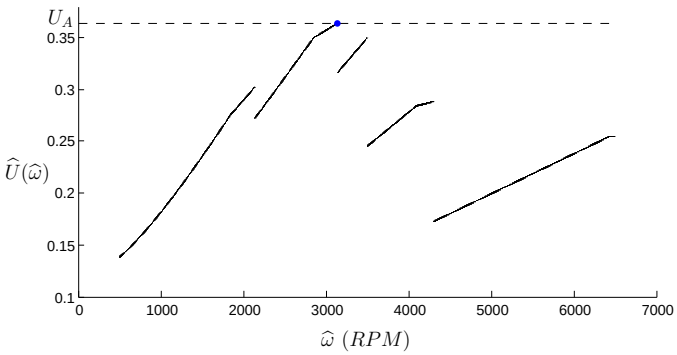

Figure 5. Example of function $\widehat{U}(\widehat{\omega})$ for set Γ_A of three AVR tasks. The utilization bound is $U_A = 0.364$ (dashed line).

B. Implementation

Despite providing an analytical expression for the utilization bound U_A, Equation (15) does not allow computing the bound in a closed-form, as it involves maximum operators over the continuous (i.e., infinite) domain of the engine speed. In the following, a closed-form expression for U_A is derived by exploiting some properties of the equations introduced in the previous section.

First of all, note that function $u'_i(\omega)$ is step-wise monotone: local maxima occur in correspondence of its discontinuities, which occur for values of ω equal to the switching speeds of the considered AVR task. As a consequence, the maximum over a continuous set in Equation (13) can be rewritten by looking at (i) the switching speeds into the interval $W(\widehat{\omega}, \Theta_i)$ and (ii) the maximum speed of the same interval, that is $\sqrt{\widehat{\omega}^2 + 2(2\pi - \Theta_i)\alpha^+}$, so obtaining

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

$$\max\left\{\max_{m=1,\ldots,M_i}\left\{u'_i(\omega_i^m) \mid \omega_i^m \in W(\widehat{\omega}, \Theta_i)\right\}, X\right\}, \quad (16)$$

where

$$X = u'_i\left(\sqrt{\widehat{\omega}^2 + 2(2\pi - \Theta_i)\alpha^+}\right).$$

Second, also Equation (15) can be solved by considering a limited set of speeds, as expressed by the following theorem.

Theorem 2: The expressions

$$U_A = \max_{\widehat{\omega} \in [\omega^{min}, \omega^{max}]} \widehat{U}(\widehat{\omega})$$

and

$$U_A = \max_{\widehat{\omega} \in \mathcal{W} \cup \{\omega^{max}\}} \widehat{U}(\widehat{\omega}), \quad (17)$$

are equivalent, where

$$\mathcal{W} = \bigcup_{\tau_i^* \in \Gamma_A} \left\{ \bigcup_{m=2,\ldots,M_i} \left\{ \sqrt{(\omega^m)^2 - 2(2\pi - \Theta_i)\alpha^-} \right\} \right\}. \quad (18)$$

Proof: First note that the bound U_A provided by Equation (15) is computed as the maximum of the sum of functions $\widehat{u}_i(\widehat{\omega})$ (for all AVR tasks) over all possible valid speeds $\widehat{\omega}$. Hence, its value must be originated by a particular speed $\widehat{\omega}'$ for which at least one of the functions $\widehat{u}_i(\widehat{\omega})$ exhibits a local maximum in $\widehat{\omega}'$, for some AVR task τ_i^*.

Now, consider one arbitrary AVR task $\tau_i^* \in \Gamma_A$. By studying its corresponding function $\widehat{u}_i(\widehat{\omega})$, it is possible to observe that it includes plateaus of local maxima for values in the neighborhood of speeds $\widehat{\omega}$ that correspond to the switching speeds of τ_i^* (see Figure 4). After each plateau, the function exhibits a discontinuity. Consider one of such plateaus and let $[\widehat{\omega}_p, \widehat{\omega}_{p+1}]$ the interval of speeds for which it occurs. Since functions $\widehat{u}_j(\widehat{\omega}), \forall \tau_j^* \in \Gamma_A$ are also piece-wise non-decreasing, by *increasing* speed $\widehat{\omega} \in [\widehat{\omega}_p, \widehat{\omega}_{p+1}]$ the value of $\widehat{U}(\widehat{\omega})$ can either **(i)** increase (or remain stable) or **(ii)** immediately decrease in correspondence of a discontinuity of some function $\widehat{u}_j(\widehat{\omega})$ (for some AVR task τ_j^*). In case (i), speed $\widehat{\omega}_{p+1}$ is a candidate value for U_A, while in case (ii) it must be that the local maxima of $\widehat{U}(\widehat{\omega})$ occurred in correspondence of the maximum speeds within plateaus of some other functions $\widehat{u}_j(\widehat{\omega})$.

Therefore, the maximum value of $\widehat{U}(\widehat{\omega})$ must occur for a speed $\widehat{\omega}'$ that is the upper-bound of one interval of speeds for which some function $\widehat{u}_j(\widehat{\omega})$ exhibits a plateau. By looking at Equations (12) and (11), it is possible to observe that each plateau occurs until reaching particular speeds $\widehat{\omega}'$ such that $\min W(\widehat{\omega}', \Theta_i) = \omega_i^m$ for some mode $m = 1, \ldots, M_i$. By expanding the latter equation, it follows that

$$\min W(\widehat{\omega}', \Theta_i) = \sqrt{\widehat{\omega}'^2 + 2(2\pi - \Theta_i)\alpha^-} = \omega_i^m,$$

which gives

$$\widehat{\omega}' = \sqrt{(\omega^m)^2 - 2(2\pi - \Theta_i)\alpha^-}.$$

As a consequence, the union of such speeds $\widehat{\omega}'$ for all tasks $\tau_i^* \in \Gamma^A$ and for all their corresponding modes $m = 1, \ldots, M_i$, gives the set of speeds to check for finding the maximum value of $\widehat{U}(\widehat{\omega})$. Such speeds are the ones included in the definition of the set $\mathcal{W}$ in Equation (18). Hence the theorem follows. ■

Based on Theorem 2 and Equation (16), it is possible to conclude that Equation (17) provides a closed-form expression for computing the utilization bound U_A. This formulation allows implementing a schedulability test for engine-control applications under EDF scheduling. Note that Equation (17) requires computing $\mathcal{O}(|\Gamma_A| \cdot M^{\text{MAX}})$ values, where $M^{\text{MAX}} = \max_{\tau_i^* \in \Gamma_A}\{M_i\}$; by looking at Equations (14) and (16), each of such values can in turn be computed in $\mathcal{O}(|\Gamma_A| \cdot M^{\text{MAX}})$ time. As a consequence, the proposed test has a *quadratic-time complexity*, both as a function of the number of AVR tasks and the number of their modes.

V. Experimental Results

This section presents a set of experimental results aimed at evaluating the schedulability performance of the schedulability tests presented in this paper.

Systems with a single rotation source (one engine) have been considered. The speed of the rotation source is assumed to be limited between $\omega^{min} = 500$ RPM and $\omega^{max} = 6500$ RPM[2] (typical values for a production car engine). The acceleration bounds have been selected such that the engine is able to reach the maximum speed starting from the minimum one in 35 revolutions [12], obtaining $\alpha^+ = -\alpha^- = 1.62\ 10^{-4}$ rev/ms^2.

Two experimental studies have been conducted. The first one, discussed in Section V-A, focuses on the assumptions considered in this paper, i.e., multiple AVR tasks triggered by the same rotation source. To ensure a broader comparison with the state of the art, a second study is presented in Section V-B, where the approach proposed in this paper is compared against other analysis techniques for both fixed-priority (FP) and EDF scheduling under a common setting in which there is a single AVR task.

A. Multiple AVR tasks

The following four analysis approaches have been evaluated: **(i)** the utilization bound provided in Section III, denoted as U-INDEP, which serves to compare with the case where all AVR tasks are pessimistically assumed to be independent; **(ii)** the utilization bound provided by Equation (17), denoted as U-SYNC, which takes into account the fact that AVR tasks are activated by the same rotation source; **(iii)** the maximum utilization in steady-state conditions considering the dependency of the AVR tasks on the same rotation source, i.e., $\max_\omega\{\sum_{\tau_i^* \in \Gamma_A}\{u_i(\omega)\}\}$, denoted as steady-state; and **(iv)** as a baseline for comparison, the transformation of each AVR task into a sporadic task, which is performed by taking the maximum WCET (last mode) and the minimum inter-arrival time (maximum engine speed). This approach is denoted as sporadic. It is worth repeating that the steady-state analysis approach provides an *unsafe* schedulability test.

To our records, no other analysis techniques are available to cope with the assumptions adopted in this work (i.e., EDF scheduling and multiple AVR tasks triggered by the same rotation source).

1) Workload generation: Task sets composed of n periodic/sporadic tasks and three AVR tasks have been considered. As a representative configuration for engine control applications [1], the AVR tasks are respectively activated *one*, *two* and *four* times per crankshaft revolution. All the three AVR tasks are activated at the TDC, so that they have no angular phases and angular periods 2π, π and $\pi/2$, respectively.

As a design choice of the workload generator, we required the presence of a parameter that controls the system load. Unfortunately, we found difficulty in generating a set of AVR tasks without exceeding a *given* value of processor utilization. This is because the actual processor load generated by AVR tasks depends in a *non-trivial manner* on the speed ranges of their modes—specifically, on how they overlap in the speed domain—and the acceleration bounds.

To overcome this issue in generating the task sets, we introduce the notion of *synthetic utilization*, denoted as U^{synth}

[2]In pure physical units, 500 RPM and 6500 RPM correspond to about 52.36 rad/s and 680.68 rad/s, respectively.

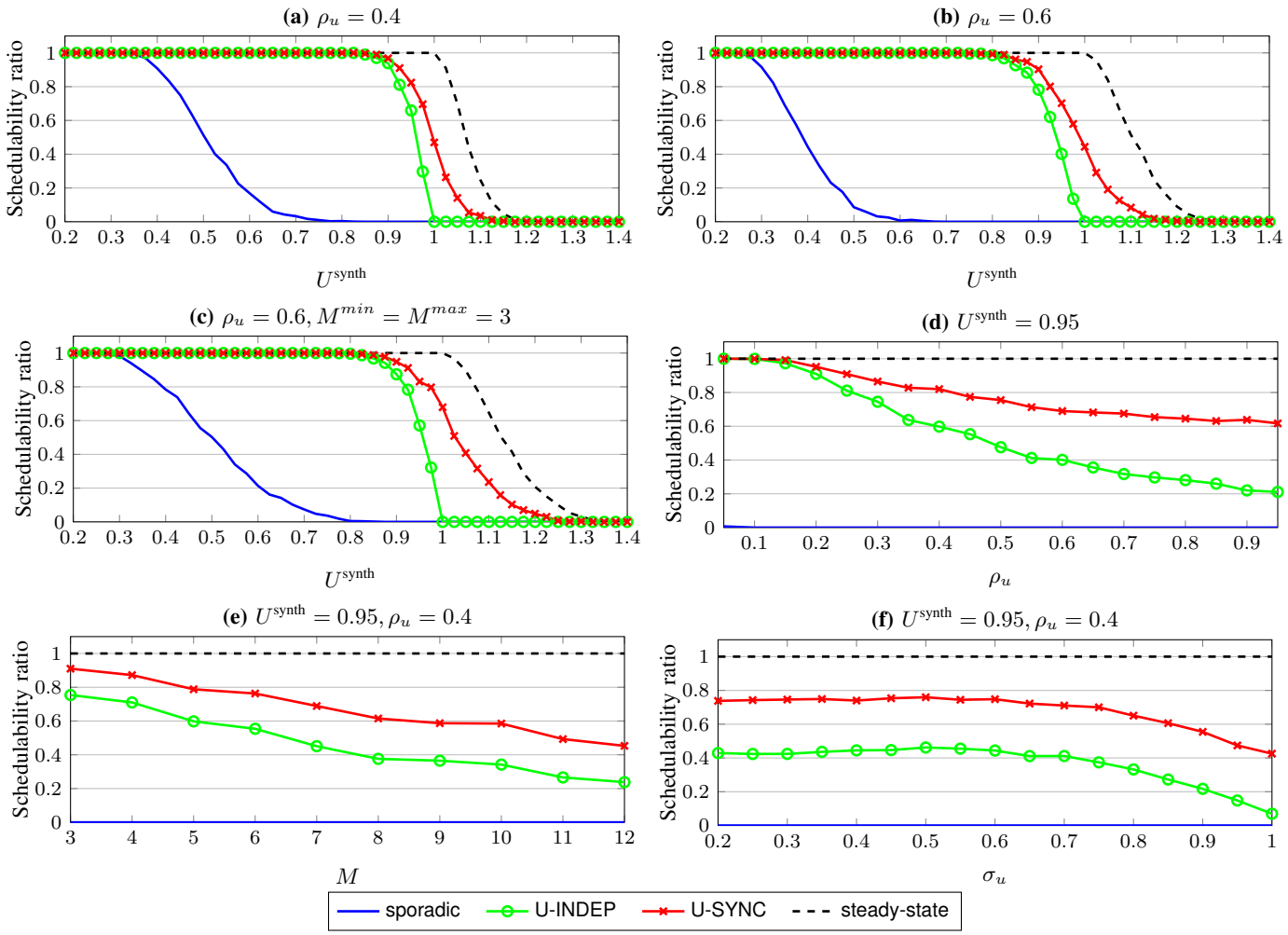

Figure 6. Experimental results for task sets that include multiple AVR tasks under six different configurations. The parameters of the configurations are reported in the captions above the figures.

and defined as follows:

$$U^{\text{synth}} = U_P + \sum_{\tau_i^* \in \Gamma_A} \max_{\omega \in [\omega^{min}, \omega^{max}]} \{U_i(\omega)\},$$

where U_P is the utilization of the periodic/sporadic tasks. As it can be noted from its analytical definition, the synthetic utilization copes with the maximum steady-state load that the AVR tasks can generate as if they would be triggered by *independent* rotation sources.

This parameter has two convenient properties: first, it provides an intuitive "knob" for varying the system load: the higher the value the higher the actual maximum system utilization; second, it simplifies the task set generation, as the WCETs of each AVR task can be independently generated without introducing biases.

A parameter $\rho_u \in [0, 1]$ has also been used to control the percentage of synthetic utilization reserved to AVR tasks. The periodic/sporadic tasks are generated with the UUnifast algorithm [13] so that their overall utilization resulted $U_P = (1 - \rho_u) \cdot U^{\text{synth}}$. The period (or minimum inter-arrival time) of such tasks has been randomly selected in the range $[3, 100]$ ms with uniform distribution.

The remaining portion of synthetic utilization (equal to $\rho_u \cdot U^{\text{synth}}$) has been divided between the AVR tasks by applying the UUnifast algorithm (which guarantees an uniform distribution), so obtaining the synthetic utilization U_i^{synth} of each AVR task. Then, each AVR task τ_i^* has been generated as follows. The number of modes M has been randomly generated in the range $[M^{min}, M^{max}]$. The values defining the range are parameters for the definition of the experimental set. A random mode m' is selected to have the maximum utilization $u_i(\omega^{m'}) = U_i^{\text{synth}}$. The utilization U^m of the other modes $m \neq m'$ are randomly generated in the range $[\sigma_u \cdot U_i^{\text{synth}}, U_i^{\text{synth}}]$, with $\sigma_u \in [0, 1]$ being another parameter to control the generation. The maximum speed ω^m of each mode $m < M$ is randomly generated in the range $[1000, 6000]$ RPM. The maximum speed for mode 1 is always set at the maximum speed ω^{max}. A minimum separation $3000/M$ RPM between any two switching speeds has been enforced. The WCET C^m of each mode m is computed as $C^m = U^m \cdot (\Theta / w^m)$. Monotonicity of the WCETs has been enforced.

2) Experiments: A first experiment has been carried out to measure the schedulability ratio of the considered analysis approaches as a function of the overall synthetic utilization U^{synth}. The experiments considered task sets composed of $n = 5$ periodic tasks and AVR tasks with $M^{min} = M^{max} = 5$ modes generated with $\sigma_u = 0.5$. The synthetic utilization has

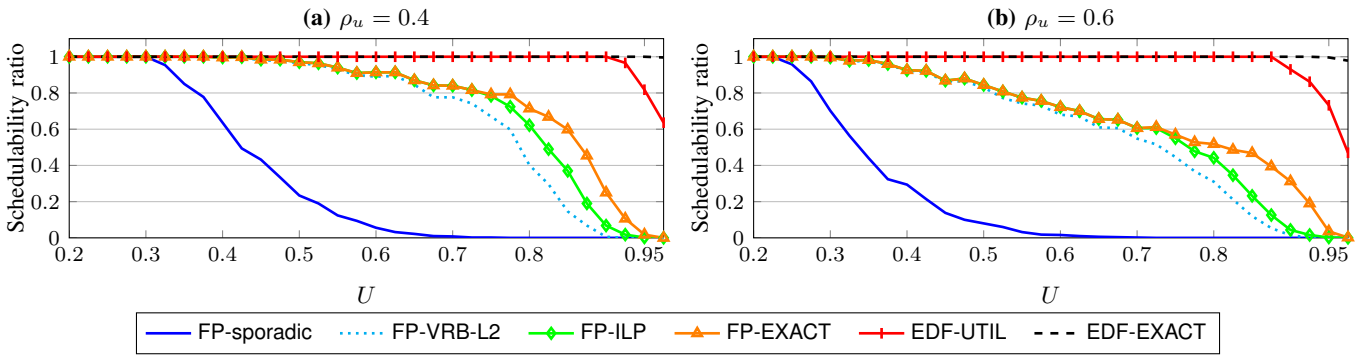

Figure 7. Experimental results for task sets that include a single AVR task. The comparison includes schedulability tests for fixed-priority scheduling and an exact test for EDF scheduling.

been varied from 0.3 to 1.4 with step 0.025, generating 1000 task sets for each value.

Figure 6-a reports the results for $\rho_u = 0.4$ while Figure 6-b reports the results for $\rho_u = 0.6$. As it can be noted from the graphs, U-SYNC provides significant benefits for high values of synthetic utilization, which correspond to scenarios in which the system has also a high actual load. It is worth observing that, since the utilization bound used in U-INDEP treats AVR tasks as being independent, it cannot guarantee task sets with $U^{\text{synth}} > 1$. Conversely, U-SYNC is able to exploit the fact that the AVR tasks are activated by the same rotation source, thus enabling a more accurate test. In fact, it allows accepting task sets with values of synthetic utilization up to about 1.15.

Note also that, in both the graphs, the transformation of AVR tasks to sporadic tasks leads to extremely poor performance, while the analysis in steady-state conditions can result very optimistic, especially for high values of U^{synth}. By varying the number of modes of the AVR tasks, it has been observed that the performance gap between U-SYNC and U-INDEP increases as the number of modes decreases. The results for a representative configuration are illustrated in Figure 6-c for $\rho_u = 0.6$ and $M^{min} = M^{max} = 3$.

A second experiment has been carried out to investigate on the dependency of the schedulability ratio on the processor load determined by AVR tasks, which has been controlled by varying the parameter ρ_u from 0.05 to 0.95 with step 0.05. For each value of ρ_u, 1000 task sets have been tested. The other parameters have been configured as in the first experiment. The results for task sets with $U^{\text{synth}} = 0.95$ are reported in Figure 6-d. As clear from the graph, both U-INDEP and U-SYNC tend to show lower performance as ρ_u increases. However, the latter one is always performing best with a performance gap up to about 40% for $\rho_u > 0.8$. Note also that, independently of the value of ρ_u, the analysis in steady-state conditions accepted all the tested task sets, while the transformation to sporadic tasks resulted completely ineffective.

A similar trend has been observed by varying the number of modes $M = M^{min} = M^{max}$ from 3 to 12. The results for $U^{\text{synth}} = 0.95$ and $\rho_u = 0.4$ are reported in Figure 6-e. Similarly, by adopting a variable range for the number of modes (i.e., $M^{min} < M^{max}$), no significant differences have been observed in the experiments above.

Finally, another experiment has been carried out to evaluate the dependency of the schedulability ratio on the σ_u parameter. The experimental results for task sets with $U^{\text{synth}} = 0.95$ and $\rho_u = 0.4$ are reported in Figure 6-f. The σ_u parameter, which controls the variability among the steady-state utilization of the modes of AVR tasks, has been varied from 0.2 to 1.0 with step 0.05. For each value, 2000 task sets have been tested. Task sets have been generated with $M^{min} = M^{max} = 5$.

As it emerges from the graph, the performance gap between U-SYNC and U-INDEP is not particularly affected by the σ_u parameter. However, the performance of both the bounds tends to decrease as σ_u increases for values higher than 0.6.

B. Single AVR task

This section reports other experimental results aimed at comparing the approach proposed in this paper against the following five analysis techniques: **(i)** the exact EDF analysis presented in [5], denoted as EDF-EXACT; **(ii)** the exact analysis for FP scheduling presented in [6], denoted as FP-EXACT; **(iii)** the ILP-based analysis for FP scheduling proposed in [12], denoted as FP-ILP; **(iv)** the VRB-L2 test for FP scheduling proposed in [12], denoted as FP-VRB-L2; and **(v)** the transformation to sporadic tasks under FP scheduling, which is denoted as FP-sporadic. To enable a comparison with all such techniques, a common setting targeting task sets with a single AVR task has been considered. The same strategy reported in Section V-A1 has been adopted for generating the tasks. The AVR task has been generated with angular period equal to 2π. Note that, under the assumption of task sets with a single AVR task, the synthetic utilization defined in the previous section is equal to the utilization U. In the considered setting, the utilization bounds presented in Sections III and IV are the same: the corresponding schedulability test is denoted as EDF-UTIL.

Figure 7 reports some results for task sets composed of $n = 5$ periodic tasks and an AVR task with $M^{min} = M^{max} = 6$ modes generated with $\sigma_u = 0.85$, where the utilization U has been varied from 0.2 to 0.975 with step 0.025, generating 1000 task sets for each value.

From Figure 7-a it emerges that the EDF-UTIL test shows the same performance of the EDF-EXACT test except for high utilization values, exhibiting a significant performance gap only for $U > 0.95$. A similar trend emerges from Figure 7-b, where the system load related to the AVR task is higher ($\rho_u = 0.6$). Analogously as it has been observed in [5], the EDF-EXACT test shows near-optimal performance and a significant improvement over the schedulability tests for FP scheduling.

VI. Related work

A task model suitable for engine control tasks has been first proposed by Kim et al. [14], who derived preliminary schedulability results under simple assumptions. In particular, their analysis applies to a single AVR task with inter-arrival time always smaller than the periods of the other tasks, and running at the highest priority level. In 2012, Negrean et al. [15] discussed the problem of analyzing the mode-changes of engine-triggered tasks by means of standard mode-change analysis techniques. Pollex et al. [16] presented a sufficient schedulability analysis under fixed priorities, assuming a constant angular velocity. The dynamic behavior of AVR tasks under fixed-priority scheduling has been analyzed by Davis et al. [12], who proposed a sufficient test based on an Integer Linear Programming (ILP) formulation and quantization of the speed domain. An exact response-time analysis of engine-control applications under fixed priorities has been proposed by Biondi et al. [6], [17]. 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. Feld and Slomka [18] derived an analysis that supports arbitrary angular phases. However, their approach only works for homogeneous tasks sets with no periodic tasks. Huang and Chen [19] addressed the analysis of sporadic tasks with mode-changes under fixed-priority scheduling: their approach also allows for different priorities for each mode.

Considering EDF scheduling, the schedulability analysis of a mixed set of AVR and implicit-deadline periodic tasks has first been addressed by Buttazzo et al. [7]. Both steady-state and dynamic conditions have been considered for deriving utilization bounds of AVR tasks. All the AVR tasks have been assumed to be triggered by independent rotation sources. Biondi and Buttazzo [20] proposed a schedulability analysis for a set of engine-triggered tasks activated by the same rotation source. Their analysis is based on utilization bounds, which however have been only formulated over the continuous domain of the engine speed. This paper extends [20] by including the following main additional contributions: **(i)** a *closed-form* formulation of the presented analysis technique, which enables the implementation of a schedulability test; **(ii)** an experimental evaluation, which was completely missing in [20]; and **(iii)** a detailed discussion for explaining the foundations of EDF scheduling of engine-triggered tasks. In 2015, Guo and Baruah [9] proposed sufficient analysis methods for systems of AVR and sporadic tasks that are managed under EDF scheduling. The authors proposed a pragmatic approach based on the transformation of AVR tasks to *digraph real-time* (DRT) tasks [21]. Speed-up factors of the proposed schedulability tests have also been presented. Still in 2015, Biondi et al. [5] presented an exact feasibility analysis for AVR tasks under EDF scheduling, where the notion of dominant speeds has been used to compute the worst-case workload generated by an AVR task in a given time window. Their analysis is limited to AVR tasks with the same angular period. Recently, Mohaqeqi et al. [22] presented an alternative exact analysis method for EDF scheduling where AVR tasks are transformed into DRT tasks: the resulting analysis is similar to the one proposed [5].

Motivated by the theoretical benefits provided by dynamic-priority scheduling, Apuzzo et al. [23] proposed an OSEK-like kernel support for engine-control applications that implements EDF scheduling. A design methodology for selecting the switching speeds of AVR tasks to optimize the engine performance is also available [24].

VII. Conclusion and Future work

This paper presented schedulability tests for engine control applications under EDF scheduling. The tests are based on utilization factors that have been obtained by integrating physical constraints in the derivation of upper-bounds on the workload generated by AVR tasks. The bounds have been first presented as a function of a continuous speed domain; then, closed-form expressions have been provided to actually implement efficient schedulability tests. Experimental results have been presented to assess the performance of the proposed tests, which confirmed their effectiveness even at high values of processor utilization. Future work includes the derivation of new analysis techniques to cope with arbitrary angular phases, the study of partitioning methodologies for multicore systems, and methods to handle transient overloads.

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] V. Hillier, P. Coombes, and D. Rogers, *Hillier's Fundamentals of Motor Vehicle Technology (Book 2)*. Nelson Thornes, 2006.

[5] A. Biondi, G. Buttazzo, and S. Simoncelli, "Feasibility analysis of engine control tasks under EDF scheduling," in *Proc. of the 27th Euromicro Conference on Real-Time Systems (ECRTS 2015)*, Lund, Sweden, July 8-10, 2015.

[6] A. Biondi, M. D. Natale, and G. Buttazzo, "Response-time analysis for real-time tasks in engine control applications," in *Proceedings of the 6th International Conference on Cyber-Physical Systems (ICCPS 2015)*, Seattle, Washington, USA, April 14-16, 2015.

[7] 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.

[8] M. Dertouzos, "Control robotics: the procedural control of physical processes," *Information Processing*, vol. 74, 1974.

[9] Z. Guo and S. Baruah, "Uniprocessor EDF scheduling of AVR task systems," in *Proc. of the ACM/IEEE 6th International Conference on Cyber-Physical Systems (ICCPS 2015)*, Seattle, USA, April 2015.

[10] 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.

[11] A. Biondi, "Analysis and design optimization of engine control software," Ph.D. dissertation, Scuola Superiore Sant Anna, Pisa, Italy, 2017.

[12] 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 2014.

[13] E. Bini and G. C. Buttazzo, "Measuring the performance of schedulability tests," *Real-Time Systems*, vol. 30, no. 1-2, 2005.

[14] 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 2012, pp. 28–38.

[15] M. Negrean, R. Ernst, and S. Schliecker, "Mastering timing challenges for the design of multi-mode applications on multi-core real-time embedded systems," in *Proceedings of the Embedded Real Time Software and Systems 2012 (ERTS2)*, Toulouse, France, Feb. 1-3 2012.

[16] 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.

[17] 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.

[18] T. Feld and F. Slomka, "Sufficient response time analysis considering dependencies between rate-dependent tasks," in *Proc. of the International Conference on Design, Automation and Test in Europe (DATE 2015)*, Grenoble, France, March 9-13, 2015, pp. 519–524.

[19] W.-H. Huang and J.-J. Chen, "Techniques for schedulability analysis in mode change systems under fixed-priority scheduling," in *Proc. of the IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2015)*, Hong Kong, China, August 2015.

[20] A. Biondi and G. Buttazzo, "Engine control: Task modeling and analysis," in *Proc. of the International Conference on Design, Automation and Test in Europe (DATE 2015)*, Grenoble, France, March 9-13, 2015.

[21] M. Stigge and W. Yi, "Combinatorial abstraction refinement for feasibility analysis," in *Proceedings of the 34nd IEEE Real-Time Systems Symposium (RTSS 2013)*, 2013.

[22] M. Mohaqeqi, J. Abdullah, P. Ekberg, and W. Yi, "Refinement of workload models for engine controllers by state-space partitioning," in *Proc. of the 29th Euromicro Conference on Real-Time Systems (ECRTS 2017)*, Dubrovnik, Croatia, June 2017.

[23] V. Apuzzo, A. Biondi, and G. Buttazzo, "OSEK-like kernel support for engine control applications under EDF scheduling," in *Proceedings of the 22nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2016)*, Vienna, Austria, April 11-14, 2016.

[24] A. Biondi, M. D. Natale, and G. Buttazzo, "Performance-driven design of engine control tasks," in *Proceedings of the 7th International Conference on Cyber-Physical Systems (ICCPS 2016)*, Vienna, Austria, April 11-14, 2016.

Alessandro Biondi is Assistant Professor at the Real-Time Systems (ReTiS) Laboratory of the Scuola Superiore Sant'Anna. He graduated (cum laude) in Computer Engineering at the University of Pisa, Italy, within the excellence program, and received a Ph.D. in computer engineering at the Scuola Superiore Sant'Anna under the supervision of Prof. Giorgio Buttazzo and Prof. Marco Di Natale. In 2016, he has been visiting scholar at the Max Planck Institute for Software Systems (Germany). His research interests include design and implementation of real-time operating systems and hypervisors, schedulability analysis, cyber-physical systems, synchronization protocols, and component-based design for real-time multiprocessor systems. He was recipient of four Best Paper Awards, and an Outstanding Paper Award.

Giorgio Buttazzo is Full Professor of computer engineering at the Scuola Superiore Sant'Anna of Pisa. He graduated in electronic engineering at the University of Pisa in 1985, received a M.S. degree in computer science at the University of Pennsylvania in 1987, and a Ph.D. in computer engineering at the Scuola Superiore Sant'Anna of Pisa in 1991. From 1987 to 1988, he worked on active perception and real-time control at the G.R.A.S.P. Laboratory of the University of Pennsylvania, Philadelphia. He has been Program Chair and General Chair of the major international conferences on real-time systems and Chair of the IEEE Technical Committee on Real-Time Systems. He is Editor-in-Chief of Real-Time Systems, Associate Editor of the ACM Transactions on Cyber-Physical Systems, and IEEE Fellow since 2012. He has authored 7 books on real-time systems and over 200 papers in the field of real-time systems, robotics, and neural networks.