Real-Time Analysis of Engine Control Applications with Speed Estimation

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

***Abstract*—Engine control applications include computational activities that adapt their behavior as a function of the engine speed, referred to as adaptive variable-rate (AVR) tasks. Although a substantial amount of work has been done to analyze the timing behavior of real-time applications with AVR tasks, most of the authors assumed the knowledge of the instantaneous engine speed at any instant. In practice, however, the instantaneous engine speed is not known and can only be estimated by various techniques, which hence introduce an error with respect to the ideal case of perfect knowledge. If not properly handled, such an error can result in a potentially unsafe analysis. This paper proposes a general approach to include speed estimators in the analysis of engine control applications and shows two particular examples using common speed estimators. Finally, estimators are also characterized through a numerical evaluation and experimental results are presented to evaluate their impact in terms of system schedulability.**

I. Introduction

Engine control systems typically include *periodic tasks*, activated by a timer at a fixed rate, and *angular tasks*, activated at specific rotation angles of the crankshaft [1]. As a consequence, the activation rate of angular tasks, as well as their computational load, is proportional to the engine speed. This means that, at high engine speeds, angular tasks could possibly cause an overload condition on the engine control unit (ECU) executing the application. If not properly handled, overloads may introduce long delays on task executions that can significantly degrade the system performance or even cause a loss of certain functions [2]. To avoid such problems, angular tasks are often implemented to adapt their behavior as a function of the speed, by switching to execution modes with lower computational demand at predetermined rotation speeds [3]. For this reason, angular tasks are also referred to as *adaptive variable-rate* (AVR) tasks.

The schedulability analysis of engine control applications with AVR tasks has been addressed by several authors under different models and assumptions. Kim, Lakshmanan, and Rajkumar [4] derived a preliminary schedulability test under a simplified model including a single AVR task running at the highest priority and having a period always smaller than those of other periodic tasks. In addition, the analysis was derived assuming implicit deadlines and priorities assigned by the Rate-Monotonic algorithm.

A sufficient analysis under fixed priorities has been proposed by Pollex et al. [5], but only in steady-state conditions (constant angular velocity). A sufficient schedulability test under fixed-priority scheduling and dynamic behavior has been proposed by Davis et al. [6] considering a formulation based on Integer Linear Programming (ILP).

The exact characterization of an AVR task under fixed priorities was first derived by Biondi et al. [7] using a search approach in the speed domain, where the concept of dominant speeds are used to contain the complexity and avoid speed quantization. Then, the exact response time analysis for both AVR and periodic tasks has been presented by Biondi et al. [8]. Feld and Slomka [9] extended the analysis for AVR tasks having arbitrary angular phases, but only for homogeneous tasks sets with no periodic tasks.

The analysis of engine control applications under Earliest Deadline First (EDF) scheduling has been addressed by Buttazzo, Bini, and Buttle [10], but assuming that AVR tasks are linked to independent rotation sources. A sufficient EDF analysis of AVR tasks triggered by a common rotation source has been derived by Biondi and Buttazzo [11]. Then, Guo and Baruah [12] proposed a speedup factor analysis and sufficient schedulability tests under EDF. An exact test of engine control applications under EDF for AVR tasks triggered by a common rotation source was first derived by Biondi et al. [13].

In most of the works considered above, the analysis is performed assuming a precise knowledge of the instantaneous engine speed at any instant. In practice, however, the instantaneous engine speed is not known and can only be estimated by various techniques, which hence introduce an error with respect to the ideal case of perfect knowledge. According to our records, the only work in which the analysis is based on a speed estimator is the one by Davis et al. [6]. In this work, the engine speed at the task activation time is estimated as the ratio of the angular period of the AVR task and its last interarrival time. However, the proposed analysis is tailored to that specific estimator. Considering the importance of such an aspect, the goal of this work is to present a general approach for including a speed estimator in the schedulability analysis of engine control applications and characterize the corresponding error introduced by the estimation.

Contributions: This paper has the following contributions:

- A generic approach is proposed to integrate the speed estimation in the real-time analysis of engine control applications.
- Two examples of common speed estimators are studied and analyzed: the first one computes the average engine speed over a fixed crankshaft angular interval, whereas the second one computes the average speed over a fixed time interval.
- Finally, the presented estimators are characterized through a numerical evaluation and a set of experiments aimed at evaluating their impact on the system schedulability.

978-3-9815370-6-2/DATE16/ ©2016 EDAA

Paper structure: The rest of the paper is organized as follows. Section II introduces the adopted terminology and the model used for the rotation source, the tasks, and the speed estimator. Section III describes how to integrate the speed estimation in the real-time analysis of engine control applications. Section IV presents two specific examples of speed estimators and illustrates how to account for them in the schedulability analysis of AVR tasks. Section V reports a set of experimental results aimed at comparing the two estimators and quantifying their impact in the schedulability analysis. Finally, Section VI states our conclusions and future directions.

II. System model

This section introduces the models used for the rotation source, the tasks, and the speed estimator.

A. Rotation source

It is assumed that all AVR tasks are triggered by a single rotation source (the engine), which is described by the following variables:

- θ angle of the crankshaft;
- ω speed of the crankshaft;
- α acceleration of the crankshaft.

To limit the pessimism of the analysis and reflect the characteristics of a real engine, we assume that both ω and α are constrained in given intervals, that is, $\omega \in [\omega_{min},\ \omega_{max}]$ and $\alpha \in [\alpha^-,\ \alpha^+]$.

To simplify the description of the analysis, the following functions will be used in the paper, as derived by Buttazzo et al. [10]:

$$T(\Theta, \omega, \alpha) = \frac{\sqrt{\omega^2 + 2\Theta\alpha} - \omega}{\alpha} \tag{1}$$

$$\Omega(\Theta, \omega, \alpha) = \sqrt{\omega^2 + 2\Theta\alpha}. \tag{2}$$

Function (1) gives the time needed by the engine to cover an angular interval Θ under constant acceleration α when the current instantaneous speed is ω. Function (2) gives the instantaneous rotation speed Ω after an angular interval Θ under constant acceleration α when the current instantaneous speed is ω.

B. Task model

A typical engine control application is modeled as a set of n real-time preemptive tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$, where each task can be activated at fixed time intervals (*periodic*) or at specific crankshaft rotation angles (*AVR task*). For the sake of clarity, a generic AVR task will be denoted as τ_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 periodic tasks such parameters are fixed, for angular tasks they depend on the engine rotation speed ω.

In particular, an AVR task τ_i^* can be described by an *angular period* Θ_i and an *angular phase* Φ_i, so that it is activated at the following angles:

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

Hence, the period of an AVR task is inversely proportional to the engine speed ω and can be expressed as

$$T_i(\omega) = \frac{\Theta_i}{\omega}. \tag{3}$$

The angular phase Φ_i is relative to a reference position called *Top Dead Center* (TDC) corresponding to the crankshaft angle for which the piston reaches the highest position in the cylinder. In this paper, 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 ($\delta_i \in [0, 1]$), that is, $\Delta_i = \delta_i \Theta_i$.

To obtain a self-adaptive behavior, an AVR task τ_i^* is typically implemented as a set $\mathcal{M}_i$ of M_i execution modes with decreasing computational demand, each operating in a specific range of rotation speeds. In general, mode m 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

$$\mathcal{M}_i = \{(C_i^m, \omega_i^m), m = 1, 2, \ldots, M_i\}.$$

As a result, the computation time of a generic AVR task τ_i^* can be described by a non-increasing step function $C_i(\omega)$ of the instantaneous speed ω at its release, that is,

$$C_i(\omega) \in \{C_i^1, \ldots, C_i^{M_i}\}. \tag{4}$$

An example of $C_i(\omega)$ function is illustrated in Figure 1.

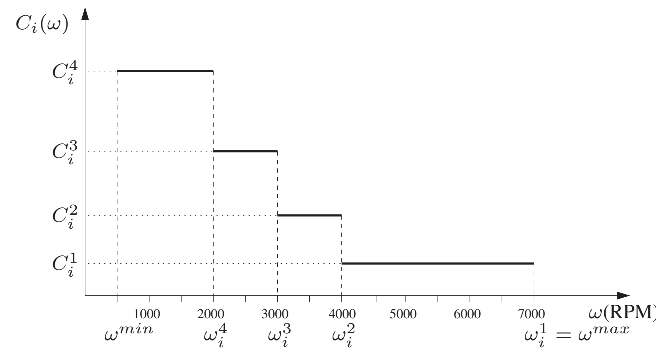

Figure 1. WCET of an AVR task as a function of the instantaneous engine speed.

Such tasks can be implemented as a sequence of `if` statements, each executing a specific subset of functions [3], [10]. As an example, Figure 2 illustrates the pseudo code of an AVR task implementing the modes reported in Figure 1, where function `read_rotation_speed()` returns the speed estimate of the rotation source available at the time at which the task is activated (not at the time at which the function is executed).

C. Speed estimator model

As already mentioned in Section I, the exact instantaneous speed at a task activation is not known and can only be estimated by various techniques. In this paper, a generic speed estimator is characterized by two functions $E_i^+(\widetilde{\omega})$ and $E_i^-(\widetilde{\omega})$, providing the *maximum* and *minimum* instantaneous speed of the rotation source for a given speed estimate $\widetilde{\omega}$ at the activation of task τ_i^*.

```
#define omega1 7000
#define omega2 4000
#define omega3 3000
#define omega4 2000

TASK (sample_task) {
      omega = read_rotation_speed();
      f0();
      if (omega ≤ omega4) f1();
      if (omega ≤ omega3) f2();
      if (omega ≤ omega2) f3();
      if (omega ≤ omega1) f4();
}
```

Figure 2. Typical implementation of an AVR task.

III. Integrating speed estimation in schedulability analysis

Note that function $C_i(\omega)$ describes the WCET of the AVR task τ_i^* as a function of the *instantaneous* speed ω at its release. Since such a speed is not known for any time instant, an *estimate* must be produced by a proper estimation algorithm. More specifically, suppose that an AVR task τ_i^* is released at time t and let $\widetilde{\omega}$ be the *speed estimate* available in the system at time t. Then, the upcoming job of τ_i^* is released in a mode that could be different than the mode that would have been activated at speed ω. If not properly handled, this phenomenon could result in an unsafe system analysis.

To cope with the error introduced by speed estimators, this paper proposes a task set transformation such that the existing analysis methods can be applied as they are to obtain a safe schedulability test. The underlying idea is to replace each AVR task τ_i^* with another AVR task ${\tau_i}'^*$ whose WCET $C_i'(\omega)$ accounts for the maximum WCET of the jobs that can be released considering all possible speed estimates that are allowed at speed ω. Note that the task set transformation is only applied for the analysis, not in the real system implementation.

Task set transformation. The proposed transformation replaces each AVR task τ_i^* with a new AVR task ${\tau_i}'^*$, such that

$$C_i'(\omega) = \max\left\{C_i(\widetilde{\omega}) \mid \omega \in [E_i^-(\widetilde{\omega}), E_i^+(\widetilde{\omega})]\right\}. \quad (5)$$

Being $C_i(\omega)$ a piece-wise constant non increasing function, the transformation can easily be implemented by increasing the maximum switching speed ω_i^m of each mode m. This can be obtained by computing

$${\omega_i}'^m = E_i^+(\omega_i^m),$$

for each mode $m = 2, 3, \ldots, M$. Note that mode $m = 1$ is excluded from this computation because it is not possible to overestimate the maximum speed ω_{max} of the rotation source.

By construction, given an arbitrary instantaneous speed ω, each AVR task ${\tau_i}'^*$ (obtained with the proposed transformation) releases a job having a WCET that is greater than or equal to *every* possible job released by the original AVR task τ_i^*. Due to the sustainability of uniprocessor fixed-priority and EDF scheduling [14], this property determines that applying existing analyses on ${\tau_i}'^*$ will result in a safe schedulability test.

IV. Two Speed Estimators

This section analyzes two simple and common speed estimators based on the computation of the *average speed* of the rotation source, defined as the ratio of a *space* interval over a *time* interval:

- *angular estimator* - it computes the average speed by measuring the time needed by the rotation source to cover a *fixed* angular distance Θ^E;
- *periodic estimator* - it computes the average speed by sampling the rotation source angle every *fixed* periodic interval T^E.

A. Angular estimator

This estimator measures the time T needed by the rotation source to cover a *fixed* angular distance Θ^E and then computes the average speed as $\widetilde{\omega} = \frac{\Theta^E}{T}$.

A possible implementation of the angular estimator is reported in Figure 3. In this example $\Theta^E = \pi$ and it is assumed that an interrupt service routine (ISR) is executed everytime the crankshaft angle becomes 0 or 180 degrees. The last time instant at which the ISR is activated is stored in a global variable `lastTime`. The ISR computes the time T needed to cover the angular distance Θ^E as the difference between the current time and the one stored in `lastTime`. Then, it updates the speed estimation as $\widetilde{\omega} = \frac{\Theta^E}{T}$ and the value of `lastTime`.

```
typeTime    lastTime, currTime;
typeSpeed   speed;

ISR  (crankAngle_0_180) {
      currTime = getCurrentTime();
      speed = ANGLE_180/(currTime-lastTime);
      lastTime = currTime;
}
```

Figure 3. Example of implementation of the angular estimator.

This estimator is now analyzed by using the generalized model presented in Section II-C. Fot the sake of clarity, the estimator error is split in two components:

- *intrinsic error*, that is, the error introduced by estimating the instantaneous speed through an average measurement over a fixed angular distance;
- *freshness error*, that is, the error related to the lag in the use of the speed estimate with respect to the time instant at which it has been produced.

First, the intrinsic error is derived by the functions $E^+(\widetilde{\omega})$ and $E^-(\widetilde{\omega})$, which represents the maximum and minimum instantaneous speed for an estimate $\widetilde{\omega}$, respectively. Then, such functions are refined by introducing the freshness error, so obtaining functions $E_i^+(\widetilde{\omega})$ and $E_i^-(\widetilde{\omega})$ that characterize the estimator according to the model presented in Section II-C.

Intrinsic error. Suppose that at time t the estimator has computed a time interval T to cover the angular distance Θ^E, thus leading to the speed estimate $\widetilde{\omega} = \Theta^E/T$. Without loss of generality, we set $\theta(t) = \Theta^E$ and $t = T$, as reported in Figure 4. The maximum instantaneous speed $E^+(\widetilde{\omega})$ at time t is obtained in the case where the rotation source has been subjected to a maximum constant acceleration α^+ in the angular interval $[0, \Theta^E]$. In this case there exists an instantaneous speed ω_0 at $\theta = 0$ such that the time needed to move from $\theta = 0$ to $\theta = \Theta^E$ is exactly T. Following Equation (1), the time T can be computed as

$$T = T(\Theta^E, \omega_0, \alpha^+) = \frac{\sqrt{\omega_0^2 + 2\Theta^E\alpha^+} - \omega_0}{\alpha^+}. \quad (6)$$

Rewriting [1] Equation (6) to extract speed w_0 we obtain

$$w_0 = \frac{2\Theta^E\alpha^+ - (\alpha^+T)^2}{2\alpha^+T} = \frac{\Theta^E}{T} - \frac{\alpha^+T}{2}. \quad (7)$$

Hence, the maximum speed $E^+(\widetilde{\omega})$ can be computed as the instantaneous speed obtained by imposing a maximum acceleration α^+ in the angular interval $[0, \Theta^E]$, that is given by applying Equation (2) so obtaining

$$E^+(\widetilde{\omega}) = \Omega(\Theta^E, \omega_0, \alpha+) = \sqrt{\omega_0^2 + 2\Theta^E\alpha^+}. \quad (8)$$

Replacing Equation (7) in Equation (8), we obtain

$$E^+(\widetilde{\omega}) = \sqrt{\left(\frac{\Theta^E}{T}\right)^2 + \left(\frac{\alpha^+T}{2}\right)^2 + \Theta^E\alpha^+} = \\ = \sqrt{\left(\frac{\Theta^E}{T} + \frac{\alpha^+T}{2}\right)^2} = \frac{\Theta^E}{T} + \frac{\alpha^+T}{2}. \quad (9)$$

Finally, knowing that $T = \Theta^E/\widetilde{\omega}$, we have

$$E^+(\widetilde{\omega}) = \widetilde{\omega} + \alpha^+\frac{\Theta^E}{2\widetilde{\omega}}. \quad (10)$$

Following a similar reasoning it is possible to compute the minimum speed $E^-(\widetilde{\omega})$, that is

$$E^-(\widetilde{\omega}) = \widetilde{\omega} + \alpha^-\frac{\Theta^E}{2\widetilde{\omega}}. \quad (11)$$

Note that both functions $E^+(\widetilde{\omega})$ and $E^-(\widetilde{\omega})$ are directly proportional to the angular distance Θ^E, and for Θ^E tending to zero the estimator tends to be exact (i.e., producing the value of the exact instantaneous speed), as it can be intuitively guessed.

Freshness error. This error is introduced because an AVR task can be activated at angular positions for which the speed estimate in not updated. To derive a safe value for functions $E_i^+(\widetilde{\omega})$ and $E_i^-(\widetilde{\omega})$ we have to consider the worst-case in which a task is activated just before the update of the speed estimate. In this case, the rotation source has covered an angular distance $\Theta^E - \epsilon$ (with ϵ arbitrarily small) from the previous speed estimate. This reasoning determines that we have to consider an additional angular space of Θ^E in which the rotation source can have accelerated (increasing the maximum speed $E^+(\widetilde{\omega})$) or decelerated (decreasing the minimum speed $E^-(\widetilde{\omega})$). Hence for an AVR task τ_i we have

$$E_i^+(\widetilde{\omega}) = \sqrt{(E^+(\widetilde{\omega}))^2 + 2\Theta^E\alpha^+} \quad (12)$$

and

$$E_i^-(\widetilde{\omega}) = \sqrt{(E^-(\widetilde{\omega}))^2 + 2\Theta^E\alpha^-}. \quad (13)$$

These functions account for the worst-case freshness error, independently of the relation between the angular period Θ^E and the AVR task angular parameters; this generic situation is denoted as *unrelated estimation*. In the particular case in which the AVR task τ_i^* has activation synchronous with the update of the estimate (e.g., τ_i^* has an angular period $\Theta_i = k\Theta^E, k = 1, 2, \ldots$ and no angular phases are present) the freshness error is zero[2], thus $E_i^+(\widetilde{\omega}) = E^+(\widetilde{\omega})$ and $E_i^-(\widetilde{\omega}) = E^-(\widetilde{\omega})$. This particular situation is denoted as *in-phase estimation*.

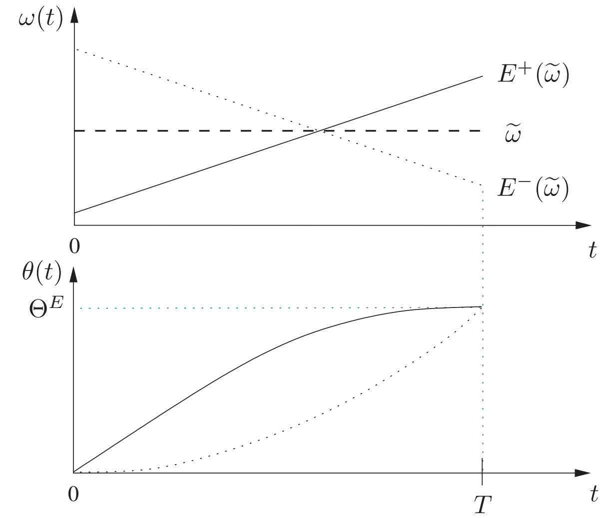

Figure 4. Worst-case scenarios for the angular estimator. The dashed line represents the average speed (estimate); the speed and angular evolution leading to the maximum and minimum speed are represented by continuous and dotted lines, respectively.

B. Periodic estimator

This estimator periodically samples the angular position $\theta(t)$ of the crankshaft with a fixed period T^E, computes the angular distance Θ from the last activation of the estimator and then computes the average speed as $\widetilde{\omega} = \frac{\Theta}{T^E}$.

In engine control systems, the angular position of the crankshaft is identified trough a crankshaft position sensor which is generally affected by a resolution error [15]. To account for such an additional error, we introduce Θ^{res} as the resolution in reading the angular position $\theta(t)$. As done for the angular estimator, the estimation error is split between *intrinsic error* and *freshness error*.

Intrinsic error. To study this estimator through the generalized model presented in Section II-C we can follow the same approach used for the angular estimator in Section IV-A. The maximum speed $E^+(\widetilde{\omega})$ can be determined by modifying Equation (9) replacing the parameters of the periodic estimator, so obtaining:

$$E^+(\widetilde{\omega}) = \frac{\Theta + \frac{\Theta^{\text{res}}}{2}}{T^E} + \frac{\alpha^+T^E}{2}. \quad (14)$$

Note that, to cope with the resolution error, the maximum speed for an estimate $\widetilde{\omega}$ is obtained by including the maximum

[1]This step can be performed by rewriting Equation (6) as $(\alpha^+T + \omega_0) = \sqrt{\omega_0^2 + 2\Theta^E\alpha^+}$ and then square both left and right hand terms.

[2]This is true assuming that the ISR in charge of updating the speed estimate has precedence over the AVR task where the speed estimate is used.

resolution error $\Theta^{\text{res}}/2$. Since $\widetilde{\omega} = \Theta/T^E$, we obtain

$$E^+(\widetilde{\omega}) = \widetilde{\omega} + \frac{\Theta^{\text{res}}}{2T^E} + \alpha^+ \frac{T^E}{2}. \quad (15)$$

Following a similar reasoning it is possible to compute the minimum speed $E_i^-(\widetilde{\omega})$ as

$$E^-(\widetilde{\omega}) = \widetilde{\omega} - \frac{\Theta^{\text{res}}}{2T^E} + \alpha^- \frac{T^E}{2}. \quad (16)$$

Freshness error. Since the rotation source triggering the AVR tasks is independent of the temporal source (e.g., a timer) activating the periodic computational activities, an AVR task can be activated at *any* time instant, independently of the execution of the periodic estimator. Specifically, assume that the speed estimate is updated at time t. In the worst-case an AVR task τ_i can be activated at time $t + T^E - \epsilon$ (with ϵ arbitrarily small), hence its mode-change is based on a speed estimate that is not "fresh", being updated $T^E - \epsilon$ time units before. To cope with this kind of error, we have to account for all possible speed evolutions in the time interval $[t, t + T^E]$.

Let $\widetilde{\omega}$ be the speed estimate at time t, then the current maximum instantaneous speed is given by $\omega = E^+(\widetilde{\omega})$. We now compute the angular distance Θ^+ covered in any time interval of length T^E starting from an instantaneous speed ω, under constant acceleration α^+. Similarly, as done in Section IV-A for Equation (7), it is possible to rewrite Equation (1) extracting the angular distance Θ, so obtaining

$$\Theta^+ = E^+(\widetilde{\omega})T^E + \alpha^+ \frac{T^E}{2}. \quad (17)$$

Hence, given a current speed estimate $\widetilde{\omega}$, the maximum speed $E_i^+(\widetilde{\omega})$ at the activation of an AVR task τ_i can be obtained by applying Equation (2) from speed $E^+(\widetilde{\omega})$ with angular distance Θ^+, that is

$$E_i^+(\widetilde{\omega}) = \sqrt{E^+(\widetilde{\omega})^2 + 2\Theta^+\alpha^+}. \quad (18)$$

The minimum speed $E_i^-(\widetilde{\omega})$ can be computed in a similar way and results

$$E_i^-(\widetilde{\omega}) = \sqrt{E^-(\widetilde{\omega})^2 + 2\Theta^-\alpha^-}, \quad (19)$$

with $\Theta^- = E^-(\widetilde{\omega})T^E + \alpha^- \frac{T^E}{2}$.

Note that the error introduced by the periodic estimator is not directly proportional to the period T^E, that is, a very fast update of the estimate can lead to a large error! This is due to the resolution error in the determination of the angular position $\theta(t)$. The derivation of the optimal period T^E can be found by computing the minimum in the error functions. Due to the high number of involved parameters, it is not easily presentable in an analytical form, but given the system parameters, it can be computed numerically.

V. Experimental results

A. Numerical evaluation

To evaluate the errors introduced by speed estimators, we carried out a numerical evaluation of the error for both the angular and periodic estimator under different configuration parameters. As done by Davis et al. [6], the acceleration range has been set so that the rotation source is able to reach ω_{max} starting from ω_{min} in 35 revolutions, so obtaining $\alpha^+ = -\alpha^- = 1.62 \cdot 10^{-4}$ rev/msec^2. The rotation speed ranges from $\omega^{min} = 500$ RPM to $\omega^{max} = 6500$ RPM, which are typical values for a production car engine.

Figure 5 reports the errors related to the angular estimator. In particular, the continuous lines report the in-phase estimation error $E(\widetilde{\omega}) = E^+(\widetilde{\omega}) - \widetilde{\omega}$ (i.e., the one considering the case where the AVR tasks are synchronously activated with the update of the estimate), whereas the dashed lines refer to the unrelated estimation error given by $E_i(\widetilde{\omega}) = E_i^+(\widetilde{\omega}) - \widetilde{\omega}$. Both errors have been evaluated for two different angular periods, $\Theta^E = 2\pi$ and $\Theta^E = \pi$. As clear for the graph, the error introduced by both estimators decreases for increasing speeds. The angular estimator with in-phase estimation shows a small error for speed greater than 3000 RPM. Note that the linear decrease of the error observed at high speeds is due to a saturation effect introduced by the maximum speed ω^{max}.

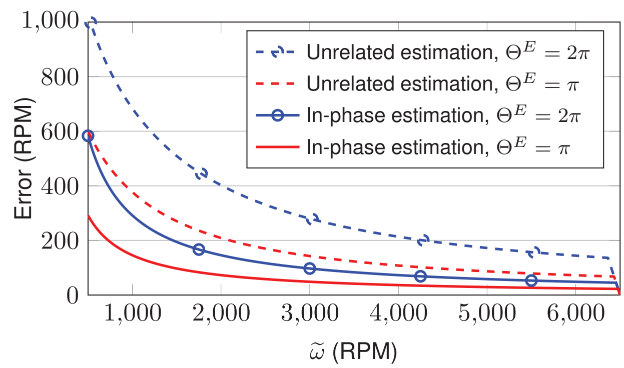

Figure 5. Errors for the angular estimator as a function of the estimate $\widetilde{\omega}$.

Figure 6 reports the error $E(\widetilde{\omega}) = E_i^+(\widetilde{\omega}) - \widetilde{\omega}$ related to the periodic estimator for different values of the period T^E. The optimal period T^E_{opt} has been numerically computed by differentiating the error function $E(\widetilde{\omega})$, and resulted to be $T^E_{\text{opt}} \approx 5.9$ ms. The angular resolution has been set to $\Theta^{\text{res}} = \pi/30$, which is a typical value in most of real engines relying only on the tooth wheel connected to the crankshaft [15]. The graph shows that the error has a little dependency on the speed (originated by the freshness error) and highlights that a fast estimation rate leads to a large error (see the curve $T^E = 1$ ms), as already explained in Section IV-B. Also in this case, the linear decrease of the error observed at high speeds is due to the saturation effect introduced by the maximum speed ω^{max}.

In general, the errors reported in the graphs show that no estimation technique dominates the other for all possible configuration parameters. However, under in-phase estimation (whenever allowed by the application), the angular estimator shows the smallest error. The next experiment evaluates the impact of the errors in terms of schedulability.

B. Schedulability experiments

The impact of the speed estimators on the schedulability of a task set has been evaluated by running other experiments based on the schedulability test presented in [8], which is related to fixed-priority scheduling. Due to lack of space, the task set generation strategy used for this experiment is not reported here but is the same as the one adopted in [8] (Section 5.1).

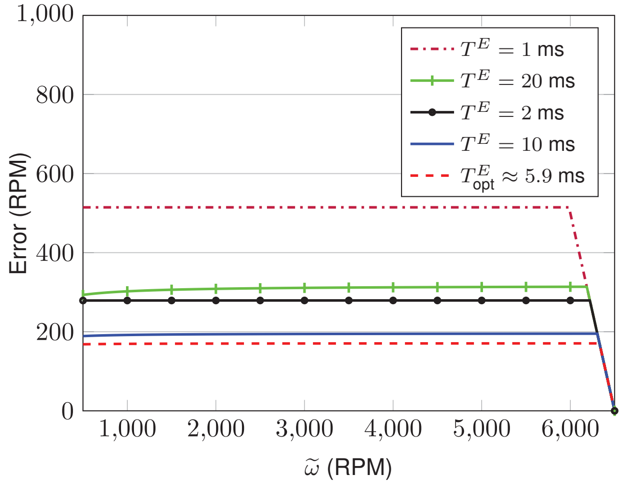

Figure 6. Errors for the periodic estimator as a function of the estimate $\widetilde{\omega}$.

Figure 7 reports the results for task sets including 5 periodic tasks and one AVR task with angular period $\Theta_i = 2\pi$ and a random number of modes M uniformly distributed between 5 and 8. Schedulability ratio has been measured as a function of the total system utilization U, varied with step 0.025 between 0.7 and 1.0; we decided to limit the graph to these utilization values because we observed that the curves have no differences for utilization values $U \leq 0.7$. The upper curve refers to the ideal case of no estimator, that is, the case in which the actual instantaneous speed is assumed to be available at any time instant; the dashed curve refers to the angular estimator with in-phase estimation; the dotted curve refers to the periodic estimator running with the optimal period $T^E_{\text{opt}} \approx 5.9$; and, finally, the lower curve refers to the angular estimator in the case of unrelated estimation. To better evaluate the impact of the estimators on schedulability performance, task sets have been generated so that the maximum utilization of the AVR task is a significant percentage (60%) of the total task set utilization. Note that the overhead introduced by the estimators is not considered in this experiment, because it is assumed that the speed estimate is performed by a dedicated processing unit, like the Time Processing Unit (TPU) introduced by Freescale [15]. The results were obtained over 1000 randomly generated task sets for each tested utilization value. As can be observed from the graph, the angular estimator with in-phase estimation shows the best performance, accepting always more than 90 percent of the task sets with respect to the case without estimator.

Overall, this experiment shows that even a simple speed estimation technique, like the angular estimator with in-phase estimation, does not significantly jeopardize the schedulability performance with respect to the ideal case of perfect knowledge of the instantaneous speed.

VI. Conclusions and Future Work

In this paper we proposed a general approach to account for speed estimators in the real-time analysis of engine control applications and we studied the error introduced by two simple speed estimators, both based on the computation of the average speed. The considered estimators have been evaluated through a numerical evaluation of their error, under different configurations. Additionally, experimental results have been carried out to measure the impact of the estimators in terms of schedulability ratio. Experimental results show that simple speed estimation techniques do not significantly penalize the system schedulability. As a future work we aim to investigate the impact of speed estimators under EDF scheduling and extend the experimental evaluation to build a broader view on the impact of speed estimators on system schedulabilty.

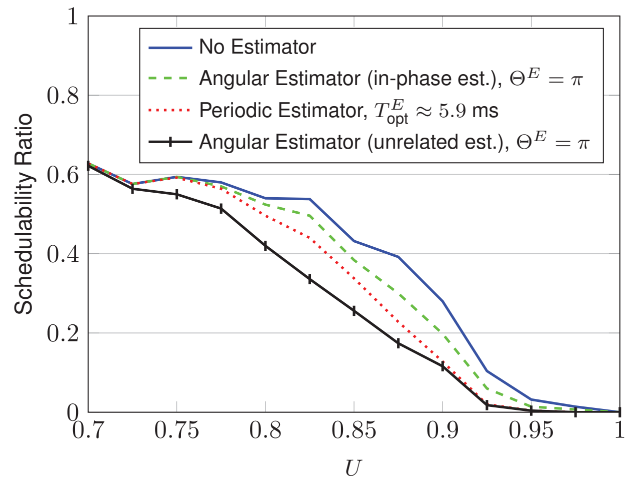

Figure 7. Schedulability ratio as a function of the system utilization U.

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] 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.

[5] 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.

[6] 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.

[7] A. Biondi, A. Melani, M. Marinoni, M. D. Natale, and G. Buttazzo, "Exact interference of adaptive variable-rate tasks under fixed-priority scheduling," in *Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS 2014)*, Madrid, Spain, July 9-11, 2014.

[8] 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.

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

[10] 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 (DATE 2014)*, Dresden, Germany, March 24-28, 2014.

[11] 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.

[12] 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.

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

[14] A. Burns and S. Baruah, "Sustainability in real-time scheduling," *Journal of Computing Science and Engineering*, vol. 2, no. 1, pp. 74–97, 2008.

[15] Freescale semiconductor Application Note AN3769. Using the engine position (CRANK and CAM) eTPU functions. [Online]. Available: http://cache.freescale.com/files/32bit/doc/app_note/AN3769.pdf