Journal of Systems and Software, Volume 138, April 2018.

A Survey of Schedulability Analysis Techniques for Rate-Dependent Tasks

Timo Feld[a,*], Alessandro Biondi[b], Robert I. Davis[c], Giorgio Buttazzo[b], Frank Slomka[a]

[a] *Institute of Embedded Systems/Real-Time Systeme, Ulm University, Germany*
[b] *Scuola Superiore SantAnna, Pisa, Italy*
[c] *Real-Time Systems Research Group, University of York, UK; Inria, France*

Abstract

In automotive embedded real-time systems, such as the engine control unit, there are tasks that are activated whenever the crankshaft arrives at a specific angular position. As a consequence the frequency of activation changes with the crankshaft's angular speed (i.e., engine rpm). Additionally, execution times and deadlines may also depend on angular speeds and positions. This paper provides a survey on schedulability analysis techniques for tasks with this rate-dependent behaviour. It covers different task-models and analysis methods for both fixed priority and earliest deadline first scheduling. A taxonomy of the different analysis methods, classifying them according to the assumptions made and the precision of the analysis, is provided at the end of the paper.

Keywords: Real-Time Analysis, Schedulability Test, Automotive, Engine Control Unit, Rate-dependent, Adaptive Variable-Rate

1. Introduction

Real-time systems are characterised by the need for both functional and timing correctness. The system must produce the correct responses to input stimuli within specified time constraints or deadlines. Real-time functionality is typically decomposed into a set of tasks that are either activated periodically in time, or directly in response to events in their environment. Considering engine management systems in the automotive domain, there are functions that need to be executed with a specific *time period* for example 5, 10, 20, 50, 100 ms. In addition, there are functions related to controlling the engine behaviour (fueling, ignition timing, and so on) that are triggered by the crankshaft rotation. Such tasks have an *angular period* measured in degrees of rotation and are triggered at specific *angular positions* or phases. Since the engine speed, measured in revolutions per minute (rpm), may vary over a wide range, from 500 rpm to more than 6500 rpm, the *angular speed* of the crankshaft and hence the rate at which these tasks are triggered varies widely. The deadline for such *rate-dependent* tasks is also measured in terms of angular rotation. For example, in a 4-cylinder petrol engine [26], the task that computes the quantity of fuel to be injected must execute every 180 degrees of rotation, with a deadline of 120 degrees. Thus at 1000 rpm, the inter-arrival time of this task is 30 ms and the relative deadline is 20 ms, whereas at 5000 rpm, the inter-arrival time is just 6 ms and the relative deadline is 4 ms.

The variation in the period and deadline of rate-dependent tasks, when viewed in the time domain, implies that the time interval available for computation is greatly reduced at high engine speeds. This means that while at lower engine speeds, typical of normal driving, there is time available to execute complex functionality aimed at optimizing fuel consumption and minimizing emissions, at higher engine speeds simpler functionality is required, otherwise the processor would be overloaded and the system would be unschedulable. In practice, different control algorithms are adopted for different ranges of engine speed, leading to tasks characterized by a set of execution modes. Figure 1 shows the worst-case execution time and the utilization of a rate-dependent task with six execution modes as a function of the angular speed in rpm. We note that the highest angular speed of each mode corresponds to the highest processor utilization for that mode.

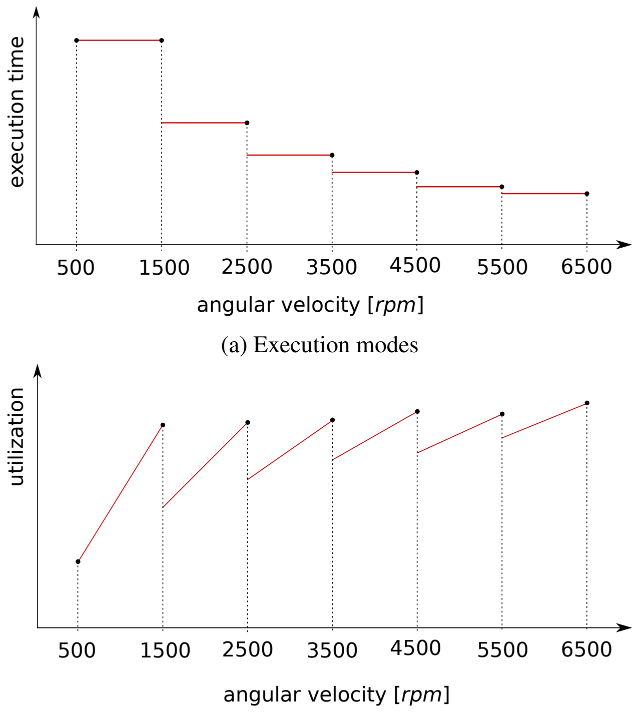
Rate-dependent tasks represent software components of a cyber-physical system. There are constraints on how the engine speed can evolve over time that derive from the physical properties of the system [26]. For example, the rate of angular acceleration is limited; a production car engine cannot go from 1000 rpm to 6000 rpm in a single revolution, thus the transitions between modes are constrained in time.

In his keynote talk [14] at the ECRTS conference in 2012, Buttle highlighted the problem of analysing tasks with inter-arrival times, deadlines and execution times that depend on engine speed. This prompted the real-time research community to look closely at this problem leading to a number of publications. In this paper, we survey the work on schedulability analysis for rate-dependent tasks triggered from rotational sources. We cover all such analyses published by July 2017. All are applicable to uniprocessor systems. We note that such tasks have appeared under a variety of names, including *Tasks with Variable Rate-dependent Behaviour (VRB)*, *Adaptive Variable-Rate (AVR)-Tasks*, *Rhythmic Tasks*, and *Engine-Triggered Tasks*. In this paper, we use the generic term *rate-dependent task*.

The remainder of the paper is organized as follows: Sec-

[*] corresponding author

[1] E-mail adresses: timo.feld@uni-ulm.de (T. Feld), alessandro.biondi@sssup.it (A. Biondi), rob.davis@cs.york.ac.uk (R. I. Davis), giorgio@sssup.it (G. Buttazzo), frank.slomka@uni-ulm.de (F. Slomka)



(a) Execution modes

(b) Utilization

Figure 1: Characterization of a rate-dependent task

tion 2 recaps other task models that can be used to approximate rate-dependent tasks. Section 3 discusses the proposed models for rate-dependent tasks. The following sections survey the literature on rate-dependent tasks. Section 4 covers the analysis for fixed priority preemptive scheduling; the standard approach for automotive systems using AUTOSAR or OSEK operating systems. Analysis for Earliest Deadline First (EDF) scheduling is covered in Section 5. Section 6 concludes the paper with a summary and perspectives on future research.

2. Previous Task Models and Terminology

Schedulability analysis has been developed for a variety of different task models. The first was the periodic task model introduced by Liu and Layland [22]. In this model, task activations are strictly periodic in time. Each task τ_i has a period T_i, a worst-case execution time C_i, and a relative deadline D_i that is *implicit* (i.e., equal to its period). This model was subsequently extended to allow sporadic arrivals with a minimum inter-arrival time of T_i and permit *constrained* ($D_i \leq T_i$) or *arbitrary* deadlines. Exact schedulability tests for the sporadic task model have been developed for fixed priority scheduling based on response time analysis [2] and for EDF scheduling based on the processor demand criterion [5]. Other extensions to the model include generalised multiframe (GMF) tasks [4], where a task can execute jobs of different types in a fixed sequence, with each job characterized by execution time, minimum inter-arrival time, and deadline parameters specific to its type. A further extension to the *non-cyclic GMF* task model [24] allows a non-cyclic order of job types. The most general model is the Digraph Real-Time (DRT) task model [28], where each task is described by a directed graph, with each vertex representing a type of job (execution time and deadline) and each edge the minimum inter-arrival time to a subsequent job of the type specified by the connected vertex.

The sporadic, non-cyclic GMF and the DRT task models can all be used to represent and analyse rate-dependent tasks; however, the approximations needed come at the expense of pessimism in the analysis. For example, using the sporadic task model, a rate-dependent task would be assumed to require its maximum execution time in its shortest period; however, this leads to substantial pessimism, as shown in [15]. The non-cyclic GMF task model can be used to represent a rate-dependent task by assigning different job types to sections of the speed range. However, since the non-cycle GMF task model allows *any* order of job types it is pessimistic. Since the Digraph can capture the transitions between job types, simple instances of the DRT task model are more suitable for modeling rate-dependent tasks [19]; however, there is still room for improvement, since physical constraints limit the possible sequences of job types. An exact characterization of rate-dependent tasks can be achieved by means of complex instances of the DRT task model [9]; however, this approach may become intractable in terms of the analysis runtime.

We note that the problem of analysing rate-dependent tasks has some similarities with the classic mode change problem; however, there are also substantial differences. For example, different rate-dependent tasks may change their execution modes according to different thresholds, and they can be driven from different independent rotational sources. Further changes in execution mode may take place over consecutive jobs. This differs from the traditional concept of operational mode changes.

3. Models for Rate-Dependent Tasks

This section presents a general model for rate-dependent tasks and then discusses some variations and restrictions that have been proposed in the literature.

A system may contain multiple **rate-dependent tasks**, as well as periodic/sporadic tasks. Different rate-dependent tasks may be triggered from either the same rotational source (e.g., the crankshaft) or *independent* rotational sources. Multiple rate-dependent tasks triggered from the same rotational source may share a common release in terms of angular position, or have *angular offsets* between their releases (similar to offset release times in the case of classical periodic tasks).

A rate-dependent task is characterized by an *angular period* at which jobs of the task are released, and a relative *angular deadline* by which computation must be completed. If the angular deadline divided by the angular period is 1, then the deadline is referred to as *implicit*, if the ratio is ≤ 1 then it is *constrained*. The behaviour of each job depends on a set of M execution modes for the task, each corresponding to a predetermined range of angular speeds. Each mode m is characterized by a WCET C^m that is valid for the speed range $(\omega^{m+1}, \omega^m]$ where ω^{M+1} and ω^1 correspond to the minimum ω^- and maximum ω^+ angular speeds allowed by the system, respectively.

When the system runs, the trajectory taken by the angular speed is limited by physical constraints on the maximum α^+ and minimum α^- *angular acceleration*, as well as the minimum and maximum permitted angular speeds.

Much of the existing literature has made simplifying assumptions, restricting the general model described above. These include (i) single vs. multiple rotational sources, (ii) single vs. multiple angular periods for the set of rate-dependent tasks, (iii) synchronous release vs. arbitrary angular offsets, (iv) constant acceleration between two jobs vs. arbitrary patterns of acceleration, (v) execution mode selection based on instantaneous angular speed vs. considering speed estimation and lag.

Table 1 (at the end of the paper) provides a taxonomy of the different analysis methods classifying them according to the assumptions made and the precision of the analysis (i.e., sufficient or exact) *with respect to* their assumptions.

A key aspect of the analysis for rate-dependent tasks relates to the acceleration model used. Simple analysis may ignore limits on the maximum rate of acceleration, assuming infinite acceleration, and hence arbitrary transitions between execution modes. This leads to a sufficient analysis. However, such an analysis considers combinations of execution modes which cannot occur in a short time interval in a system with limited acceleration. At the other extreme, if zero acceleration is assumed, then the analysis is only valid for steady-state operation. Such an analysis ignores combinations of execution modes even though they can occur under acceleration. While considering an arbitrary pattern of acceleration within limits is required for a precise and sound analysis, one simplifying assumption is to assume that acceleration is limited, but constant between jobs. While not entirely valid, this reduces the complexity of the analysis and can be corrected for.

A further important aspect is how the value of the angular speed used by the application software to select the execution mode is determined. A simplifying assumption here is to assume that the *instantaneous* angular speed at the time of the job release can be used. In practice, however, the instantaneous speed cannot be measured directly, rather it must be estimated over some angular interval, for example the previous angular period. We note that differences in the angular speed used can lead to a different sequence of execution modes for the same trajectory or evolution of angular speeds.

4. Fixed Priority Preemptive Scheduling

This section reviews the schedulability analysis for rate-dependent tasks under fixed priority preemptive scheduling. Figure 3 gives an overview of the different schedulability analyses for rate-dependent tasks under fixed priority scheduling.

4.1. Preliminary Work with Restrictive Assumptions

In 2012, Negrean et al. [25] presented a case study based on an automotive system and identified the challenge of analyzing rate-dependent tasks. They proposed the application of standard techniques for mode changes under fixed priority scheduling to each pair of modes.

The first analysis specifically designed for rate-dependent tasks was introduced by Kim et al. [21] in 2012. They derived a schedulability test assuming a restrictive scenario with a single rate-dependent task with the highest priority and an inter-arrival time that is *always* smaller than the periods of the other tasks.

In 2013, Pollex et al. [27] introduced an analysis for rate-dependent tasks under the simplifying assumption of constant angular speed, thus ignoring the effects of mode changes. In a later work [26], they accounted for arbitrary but bounded accelerations. They further relaxed the assumption of specific execution modes, allowing the relation between task execution time and angular speed to be modeled via a continuous curve. Pollex et al. [26] derived a sufficient schedulability test for task sets containing multiple rate-dependent tasks by extending classic response time analysis. They verify whether a task is schedulable by comparing the maximum response time with the deadline for each angular speed ω at which the engine can operate, hence requiring a quantization of the given speed range. The response time for a given angular speed ω is calculated by maximizing the number of jobs and the WCETs separately, which leads to high pessimism in the analysis.

In 2015, Feld and Slomka [17] introduced a sufficient schedulability test for task sets containing only rate-dependent tasks, where these tasks may have arbitrary angular offsets. They quantized the speed range and determined the maximum response times for each specific angular position.

4.2. Sufficient Tests

In 2014, Davis et al. [15] introduced a number of sufficient tests for rate-dependent tasks with arbitrary angular periods and constrained deadlines, that may be driven from multiple independent rotational sources. The task sets may also contain periodic/sporadic tasks. Further, software resources may be shared between any of the tasks according to the Stack Resource Policy [3].

The initial analysis in Section III of [15] made the simplifying assumption that acceleration may be unbounded, thus any sequence of execution mode transitions is valid. We note that this makes the tests valid, but somewhat pessimistic for the case where acceleration is bounded by the physical constraints of the system. Davis et al. [15] used an Integer Linear Program (ILP) formulation to determine the combination of periods for different execution modes of each rate-dependent task that results in the maximum total amount of execution time or *interference* $I(t)$ in an interval of length t. This interference calculation is used within response time analysis to compute the maximum response time for each execution mode of each task, and thus to check if all deadlines can be met. Huang and Chen [20] provided a schedulability test for the same problem formulation which runs in polynomial time, thus improving the runtime. Davis et al. [15] also introduced two simple linear-time approximations for the interference term. These expressions are similarly integrated into response time analysis. Evaluation shows that the simple sufficient tests provide substantially better performance than the default option of approximating rate-dependent tasks as sporadic tasks.

In Section IV of the same paper, Davis et al. [15] extended their methods to consider constraints due to bounds on the maximum and minimum rate of acceleration (approx. 10,000 rpm/sec for a production car engine). Further, they assumed that the angular speed used to select the execution mode is the average over the previous angular period; as can be obtained by recording the time at which an interrupt is raised releasing each job of the task. They also accounted for the effect that the lag between the estimated angular speed and instantaneous angular speed can have on the time available until the angular deadline. Their analysis derives constraints on the maximum total number of job releases in any mode in a time interval t starting at speed ω, and also constraints that capture the fact that if there are job releases in modes m and $m + 2$ it may be a necessary consequence that there are also job releases in the intervening mode $m + 1$. The constraints are used in an ILP formulation which determines the maximum interference $I(\omega, t)$ in an interval of length t, starting from speed ω. This interference function is then integrated into response time analysis, to determine if all deadlines are met starting from angular speed ω. To cover all possible angular speeds, quantized values of ω are used representing small speed ranges (e.g. 100 rpm) and the constraints are lifted to speed ranges using the maximum and minimum speeds from each range. This ensures that the analysis remains sufficient, see the appendix of the technical report [16] for a detailed discussion. The use of quantized speed ranges means that the analysis is sufficient, but not exact. Precision can, however, be improved by choosing a suitably small quantization. This quantization also has another effect, it means that the analysis can only cover systems with rate-dependent tasks that are driven from a single rotational source or from synchronized sources with a fixed relationship between their angular speeds (e.g. crankshaft and camshaft rotation).

4.3. Exact Tests

In 2014, Biondi et al. [10] derived an exact characterization of the interference produced by rate-dependent tasks, under the simplifying assumption that acceleration is constant between one job and the next, and that the execution mode depends on the instantaneous angular speed at job release. The interference computation is approached as a search problem in the speed domain (as illustrated by Figure 2), by relying on the fact that the instantaneous engine speeds at the activation of consecutive jobs (such as ω^0, ω^1 and ω^2 in the diagram) are constrained in corresponding ranges that are determined by the acceleration bounds α^- and α^+ of the engine. This reasoning leads to the definition of a search tree: however, since the speed domain is a continuum, such a tree is infinite. To cope with this issue, Biondi et al. [10] identified a limited set of *dominant speeds*, which are particular angular speeds that are proved to capture the worst-case behavior of rate-dependent tasks for the purpose of schedulability analysis. The use of dominant speeds drastically restricts the number of scenarios that have to be considered to characterize the exact worst-case interference, while enabling the exploration of the search tree without the need for quantization.

Feld and Slomka [18] introduced another method for the exact characterization of the worst-case interference, which allows reducing the complexity with respect to the approach proposed in [10], with a corresponding improvement of the runtime for the analysis. Similarly to Biondi et al. [10], their approach is based on the exploration of a search tree in the speed domain, but relying on a notably smaller set of initial speeds and at most three scenarios for defining the subsequent search patterns. In contrast to [10], they assumed that the angular speed used to select the execution mode is the average over the previous angular period, which corresponds to the speed estimation that can be obtained by recording the time between the last two activations of the rate-dependent task. Furthermore, the analysis in [18] also assumes arbitrary accelerations within bounds, but with the restriction that the maximum deceleration and acceleration have the same absolute value (i.e., $|\alpha^-| = |\alpha^+|$). Hence, their analysis is not exact when $|\alpha^-| \neq |\alpha^+|$. In the same paper, the authors combined their characterization of the interference with the response time analysis proposed by Biondi et al. in [11].

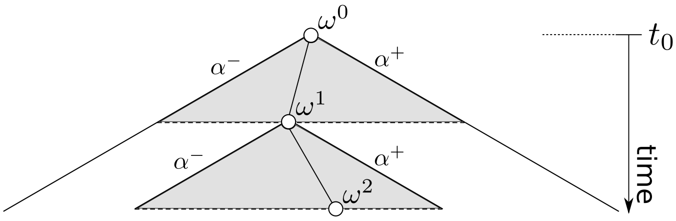


Figure 2: Search tree representing possible job sequences

In 2015, Biondi et al. [11] built on their results [10] to derive an exact schedulability test for fixed priority preemptive scheduling. They introduced an algorithm to compute the maximum response times for a mixed set of rate-dependent and periodic tasks. Since a stand-alone computation of the interference generated by rate-dependent tasks may introduce pessimism in the analysis, the algorithm leverages a dynamic pruning of the search space (defined with dominant speeds) at the stage of response-time analysis. The analysis in [11] caters only for rate-dependent tasks with the same angular period and cannot be used to produce an exact result for systems with angular offsets.

In 2016, Biondi and Butazzo [8] showed how to account for different speed estimators (measured over fixed angular distances or fixed time intervals) when using analysis originally developed considering instantaneous speeds [8]. These modifications make the analysis sufficient for the more sophisticated model.

The simplifying assumption of constant acceleration between two jobs, assumed in both [10] and [11], was relaxed by Biondi in 2017 [6], where the same analysis technique was proven to work with arbitrary but bounded acceleration patterns.

4.4. Fixed Priority Mode-level Scheduling

In fixed priority scheduling, each task is assigned a unique fixed priority, then at runtime every job of the task inherits the priority of the task. By contrast, in *fixed priority mode-level (FPM)* scheduling, each *mode* of a task is assigned a unique fixed priority, then at runtime, jobs execute with a priority which is determined by their execution mode. Huang and Chen [20] proposed an analysis for FPM-scheduling of rate-dependent tasks. They showed that a utilization bound of $2 - \sqrt{2} \approx 0.5857$ can be guaranteed in implicit deadline systems if mode priorities are assigned in *rate-monotonic priority order*.

4.5. Experiments comparing the precision of the analysis

Davis et al. [15] compared several schedulability tests via a metric referred to as the success ratio; the proportion of randomly generated task sets that are schedulable. They compared some tests based on linear upper bounds, another test obtained by reducing each rate-dependent task to the sporadic task model and their proposed test using an ILP formulation. As their experiments show, the ILP-method significantly improves upon the other tests.

Biondi et al. [11] compared the success ratio of their exact method with the ILP-method and showed that further improvement in precision can be obtained using an exact approach. Feld and Slomka [18] ran further experiments and showed some improvements in precision of their method over [11]. Since both methods from [11] and [18] are exact with respect to their assumptions, the small difference in precision between them is due to the different model used: Biondi et al. [11] assume precise knowledge of the instantaneous angular speed, while Feld and Slomka [18] assume that the angular speed is obtained by measuring the time between two activations of the task.

5. Earliest Deadline First Scheduling

This section reviews the schedulability analysis for rate-dependent tasks under Earliest Deadline First (EDF) scheduling. Figure 3 gives an overview of the different schedulability analyses for rate-dependent tasks under EDF.

5.1. Sufficient Tests

In 2014, Buttazzo et al. [13] and in 2015, Guo and Baruah [19] derived schedulability tests based on the utilization of rate-dependent tasks. In addition, Guo and Baruah [19] provided the *speedup factor* for their schedulability test, as compared to a hypothetical optimum algorithm. In this case, the speedup factor is dependent on both the acceleration bounds and the maximum angular speed for the system. For typical parameters for a production car with a petrol engine, the speedup factor is ≤ 1.14 showing that the test gives away at most around 13% utilization with respect to an optimal algorithm and exact test.

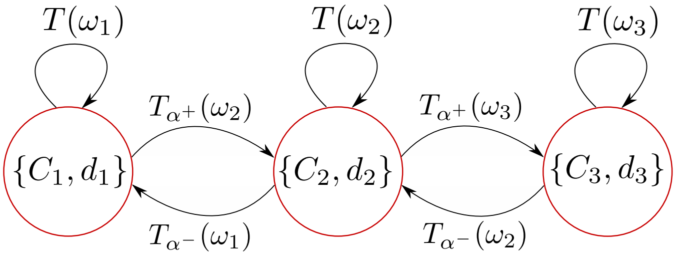


Figure 4: A rate-dependent task represented by a Digraph

Guo and Baruah [19] also proposed transforming rate-dependent tasks to the Digraph Real-Time (DRT) task model. The diagram in figure 4 illustrates a digraph representing one task with three modes. Each vertex is labeled with the execution time C and deadline d of that mode. The edges are labeled with the minimum time T between releases of the execution modes which they connect. These differ for each direction, assuming different bounds on deceleration and acceleration (and are denoted by T_{α^-} and T_{α^+}). Note that the speed ω in brackets denotes the following mode (with its index). Once all rate-dependent tasks are transformed into a DRT-model, schedulability analysis can be performed according to the methods derived by Stigge et al. [28]. Note that each vertex represents a range of angular speeds.

Mohaqeqi et al. [23] refined the representation of a rate-dependent task with the DRT task model. In particular, given a partitioning of the speed range into sub-ranges, they provided a method for constructing a safe DRT representation of a rate-dependent task where each DRT vertex is associated with a speed sub-range. Furthermore, they also showed how to derive a speed partitioning that leads to an exact analysis by means of the transformation. The proposed partitioning is quite similar to the one proposed in [9]. In contrast to [9, 11], and analogously to [6], they assumed an arbitrary but bounded acceleration. Also, rate-dependent tasks are assumed to be independent.

Buttazzo et al. [13] proposed an analysis for implicit-deadline, rate-dependent tasks driven by different independent rotational sources, under EDF scheduling.

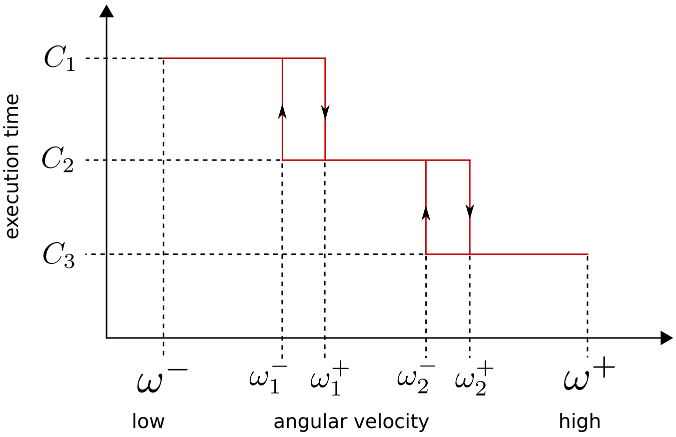


Figure 5: Hysteresis

In practical implementations, the mode change of rate-dependent tasks includes a hysteresis to allow mode switches to occur at different angular speeds under acceleration and deceleration. This is done to prevent frequent mode changes when the angular speed oscillates around the switching threshold. An example of modes with hysteresis is shown in Figure 5. While accelerating from mode 1 to mode 2, the execution mode

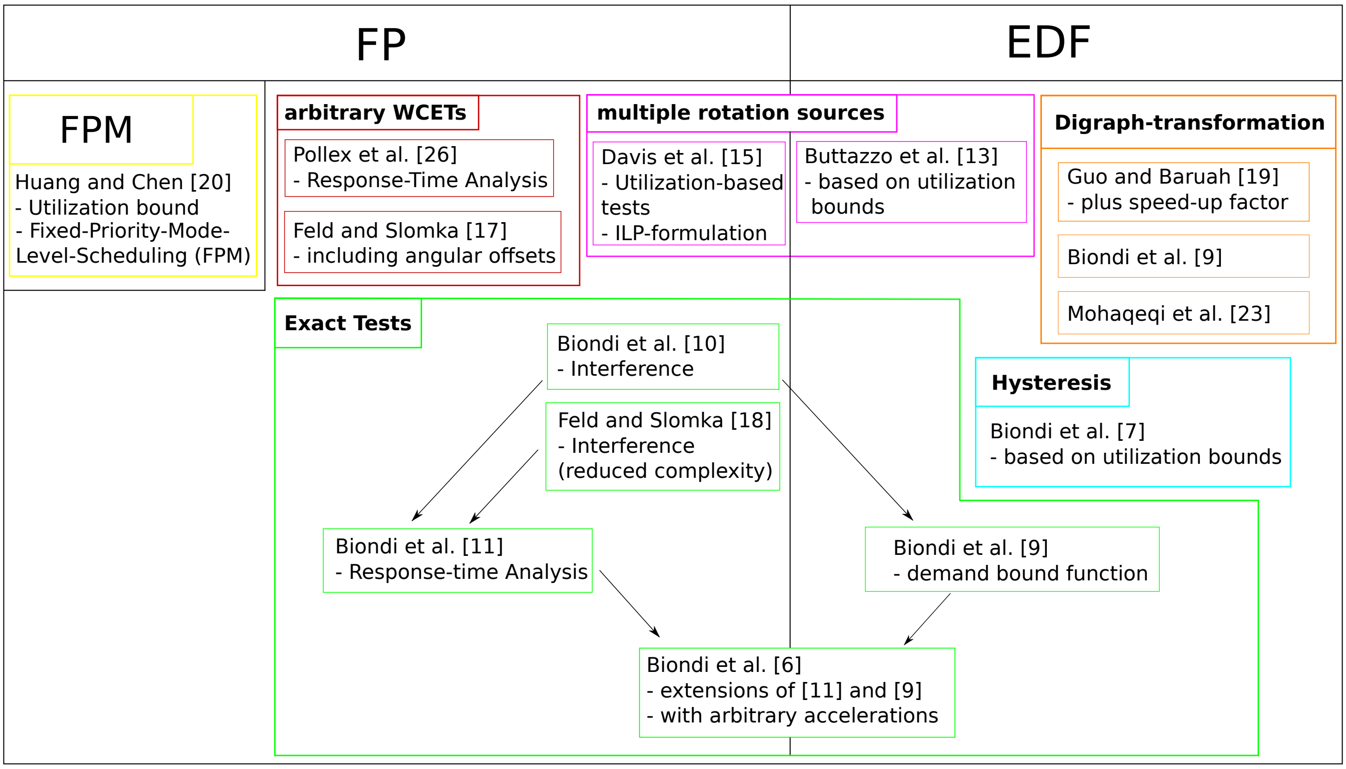


Figure 3: Overview of analyses for rate-dependent tasks

changes above ω_1^+, whereas when decelerating from mode 2 to mode 1, the mode changes at speed ω_1^-. In 2015, Biondi and Buttazzo [7] described a utilization-based schedulability test for EDF that takes hysteresis into account. The test works only under the assumption of constant acceleration between two consecutive jobs. In the presence of a set of rate-dependent tasks activated by the same rotational source, the analysis proposed in [7] allows improving the precision of the one proposed in [13]. Furthermore, if no hysteresis is considered, the schedulability test of [7] is safe even under arbitrary (but bounded) acceleration patterns.

5.2. *Exact tests*

Biondi et al. [9] derived an exact test under EDF-scheduling for constrained-deadline rate-dependent tasks, under the simplifying assumption that acceleration is constant between one job and the next, and that the execution mode depends on the instantaneous angular speed at job release. They determined the *demand bound function* [5] based on the idea of identifying the critical job sequences in the speed domain [10] (based on dominant speeds, as discussed in Section 4.3). The analysis applies only to rate-dependent tasks with the same angular period and does not produce an exact result for systems with angular offsets. In [9], it is also shown how to transform a rate-dependent task into a DRT task in order to achieve an exact analysis under the same assumptions. Further, Biondi et al. [9] as well as Guo and Baruah [19] ran experiments in which they show that EDF outperforms FP-scheduling in terms of precision.

In 2016, Biondi and Buttazzo [8] showed how to account for different speed estimators when using analysis originally developed considering instantaneous speeds. These modifications make the analysis sufficient for the more sophisticated model. The simplifying assumption of constant acceleration between two jobs was relaxed by Biondi in 2017 [6], where the same analysis technique of [9] was proven to work with arbitrary but bounded acceleration patterns.

We note that all exact methods [9, 7, 11, 18], the ILP-method [15] and all methods based on transformations to the DRT task-model [23, 19] have an analysis runtime which has exponential complexity. Other tests [26, 13, 17] require a linear number of the standard RTA or Processor Demand Tests (which themselves have pseudo-polynomial complexity).

6. Perspectives and Future work

This survey has reviewed research into schedulability analysis for rate-dependent tasks under fixed priority or EDF scheduling, triggered via sources of angular rotation, such as the crankshaft or camshafts of a 4-stroke petrol engine. Initially, sufficient analyses were developed with a number of simplifying assumptions applied. More recently, these have been supplanted by more sophisticated analyses that make use of constraints provided by the physical system (e.g., limits on the maximum rate of acceleration). Exact analyses have been developed for simplified models (assuming that all the rate-dependent tasks have the same angular period, and assuming that the instantaneous angular speed is available), which have been extended by relaxing some assumptions at the cost of a lower precision. Table 1 provides a taxonomy of the different

Table 1: Taxonomy

Paper	**Scheduler**	**Precision**	**Rotation source**	**Multiple arbitrary angular periods/ offsets**	**Acceleration model**	**WCET model**	**Notes**
Kim et al. [21]	RM	sufficient	single	no / no	arbitrary	execution modes	single rate-dependent task at the highest priority
Pollex et al. [27]	FP	sufficient	single	yes / no	none (steady-state)	execution modes	ignores mode changes and accelerations
Pollex et al. [26]	FP	sufficient	single	yes / no	arbitrary	arbitrary	pessimism due to separate maximization
Feld and Slomka [17]	FP	sufficient	single	yes / yes	arbitrary	arbitrary	only task sets without periodic tasks
Davis et al. [15]	FP	sufficient	multiple independent	yes / no	n.a.	execution modes	ignores physical constraints
Davis et al. [15]	FP	sufficient	single	yes / no	arbitrary	execution modes	requires quantization to setup an ILP formulation
Biondi et al. [10]	FP	sufficient	single	no / no	constant between two jobs	execution modes	characterization of the exact interference
Feld and Slomka [18]	FP	sufficient	single	no / no	arbitrary	execution modes	characterization of the exact interference (reduced complexity)
Biondi et al. [11]	FP	exact	single	no / no	constant between two jobs	execution modes	based on [10]
Guo and Baruah [19]	EDF	sufficient	single	yes / no	arbitrary	execution modes	model transformation to DRT tasks, provides speed-up factor
Mohaqeqi et al. [23]	EDF	exact	single	no / no	arbitrary	execution modes	exact model transformation to DRT tasks
Biondi and Buttazzo [7]	EDF	sufficient	single	yes / no	constant between two jobs	execution modes	based on utilization bounds, accounts for hysteresis
Huang and Chen [20]	FP	sufficient	n.a.	n.a.	n.a.	execution modes	ignores physical constraints
Buttazzo et al. [13]	EDF	sufficient	multiple independent	yes / no	arbitrary	execution modes	based on utilization bounds
Biondi et al. [9]	EDF	exact	single	no / no	constant between two jobs	execution modes	same analysis method of [11]
Biondi [6]	FP and EDF	exact	single	no / no	arbitrary	execution modes	extended versions of [11] and [9]

analysis methods classifying them according to the assumptions made and the precision of the analysis (i.e., sufficient or exact) *with respect to* their assumptions.

Possible future work includes a precise analysis for a more comprehensive rate-dependent task model with arbitrary but bounded acceleration, angular speed derived via a practical method, and hysteresis in the execution mode changes. Further enhancements include covering systems with tasks triggered from multiple independent rotational sources, and providing analysis for systems with angular offsets. Another area for improvement would be to reduce the runtime complexity of precise analysis. Further work could also extend schedulability tests for rate-dependent tasks to multi-core systems.

Other areas for future research include design methods for optimizing the performance of the control functions implemented by systems with rate-dependent tasks. For instance, the design problem of determining the switching speeds for rate-dependent tasks that optimize the engine performance has been addressed by Biondi et al. in [12]. In the context of rate-dependent tasks it is clear that EDF scheduling has considerable advantages over fixed priority scheduling. Recent work has therefore sought to provide RTOS support (similar to the OSEK/AUTOSAR standards) for rate-dependent tasks under EDF [1]. A final area for further research is to note that the control systems implemented via rate-dependent tasks can often tolerate a small number of deadline misses, provided they are not too frequent. Future work may therefore consider relaxed temporal constraints, allowing for some transient overloads.

Acknowledgements

This work was partially funded by the UK EPSRC MCCps project (EP/P003664/1) and by the INRIA International Chair program.

References

[1] Apuzzo, V., Biondi, A., Buttazzo, G., 2016. OSEK-like kernel support for engine control applications under EDF scheduling. In: Real-Time and Embedded Technology and Applications Symposium (RTAS).

[2] Audsley, N. C., Burns, A., Richardson, M., Wellings, A. J., September 1993. Applying new scheduling theory to static priority pre-emptive scheduling. Software Engineering Journal 8, 284 – 292.

[3] Baker, T. P., 1990. A stack-based resource allocation policy for realtime processes. In: Real-Time Systems Symposium (RTSS).

[4] Baruah, S., Chen, D., Gorinsky, S., Mok, A., July 1999. Generalized multiframe tasks. Real-Time Systems 17, 5–22.

[5] Baruah, S., Howell, R., Rosier, L., November 1990. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. Real-Time Systems 2, 301 – 324.

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

[7] Biondi, A., Buttazzo, G., 2015. Engine control: task modeling and analysis. In: Design Automation and Test in Europe (DATE).

[8] Biondi, A., Buttazzo, G., 2016. Real-time analysis of engine control applications with speed estimation. In: Design Automation and Test in Europe (DATE).

[9] Biondi, A., Buttazzo, G., Simoncelli, S., 2015. Feasibility analysis of engine control tasks under EDF scheduling. In: Euromicro Conference on Real-Time Systems (ECRTS).

[10] Biondi, A., Melani, A., Marinoni, M., Natale, M. D., Buttazzo, G., 2014. Exact interference of adaptive variable-rate tasks under fixed-priority scheduling. In: Euromicro Conference on Real-Time Systems (ECRTS).

[11] Biondi, A., Natale, M. D., Buttazzo, G., 2015. Response-time analysis for real-time tasks in engine control applications. In: Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems (ICCPS). pp. 120–129.

[12] Biondi, A., Natale, M. D., Buttazzo, G., 2016. Performance-driven design of engine control tasks. In: Proceedings of the 7th International Conference on Cyber-Physical Systems (ICCPS). No. 45.

[13] Buttazzo, G., Bini, E., Buttle, D., 2014. Rate-adaptive tasks: Model, analysis and design-issues. In: Design Automation and Test in Europe (DATE).

[14] Buttle, D., 2012. Real-time in the prime time. Keynote talk at the Euromicro Conference on Real-Time Systems (ECRTS).

[15] Davis, R., Feld, T., Pollex, V., Slomka, F., 2014. Schedulability tests for tasks with variable rate-dependent behaviour under fixed priority scheduling. In: Real-Time and Embedded Technology and Applications Symposium (RTAS).

[16] Davis, R. I., Feld, T., Pollex, V., Slomka, F., 2014. Schedulability tests for tasks with variable rate-dependent behaviour under fixed priority scheduling. Tech. rep., University of York, Department of Computer Science Technical Report.

[17] Feld, T., Slomka, F., 2015. Sufficient response time analysis considering dependencies between rate-dependent tasks. In: Design Automation and Test in Europe (DATE).

[18] Feld, T., Slomka, F., 2017. Exact interference of tasks with variable rate-dependent behaviour. accepted at Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)Digital Object Identifier: 10.1109/TCAD.2017.2729459.

[19] Guo, Z., Baruah, S., 2015. Uniprocessor EDF scheduling of AVR task systems. In: Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems (ICCPS). pp. 159–168.

[20] Huang, W.-H., Chen, J.-J., 2015. Techniques for schedulability analysis in mode change systems under fixed-priority scheduling. In: International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA).

[21] Kim, J., Lakshmanan, K., Rajkumar, R., 2012. Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems. In: Proceedings of the IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS). pp. 55–64.

[22] Liu, C. L., Layland, J. W., January 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM 20, 46–61.

[23] Mohaqeqi, M., Abdullah, J., Ekberg, P., Yi, W., 2017. Refinement of workload models for engine control by state space partitioning. In: Euromicro Conference on Real-Time Systems (ECRTS).

[24] Moyo, N. T., Nicollet, E., Lafaye, F., Moy, C., 2010. On schedulability analysis of non-cyclic generalized multiframe tasks. In: Euromicro Conference on Real-Time Systems (ECRTS).

[25] Negrean, M., Ernst, R., Schliecker, S., 2012. Mastering timing challenges for the design of multi-mode applications on multi-core real-time embedded systems. In: Embedded Real Time Software and Systems (ERTS).

[26] Pollex, V., Feld, T., Slomka, F., Margull, U., Mader, R., Wirrer, G., 2013. Sufficient real-time analysis for an engine control unit. In: Proceedings of the 21st International conference on Real-Time Networks and Systems (RTNS). pp. 247–254.

[27] Pollex, V., Feld, T., Slomka, F., Margull, U., Mader, R., Wirrer, G., 2013. Sufficient real-time analysis for an engine control unit with constant angular velocities. In: Design Automation and Test in Europe (DATE).

[28] Stigge, M., Ekberg, P., Guan, N., Yi, W., 2011. The digraph real-time task model. In: Real-Time and Embedded Technology and Applications Symposium (RTAS).