The Journal of Systems and Software 138 (2018) 100–107

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A survey of schedulability analysis techniques for rate-dependent tasks

Timo Feld[a,*], Alessandro Biondi[b], Robert I. Davis[c,d], 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*
[d] *Inria, France*

ARTICLE INFO

Article history:
Received 15 August 2017
Revised 19 December 2017
Accepted 20 December 2017
Available online 22 December 2017

Keywords:
Real-time analysis
Schedulability test
Automotive
Engine control unit
Rate-dependent
Adaptive variable-Rate

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.

1. Introduction

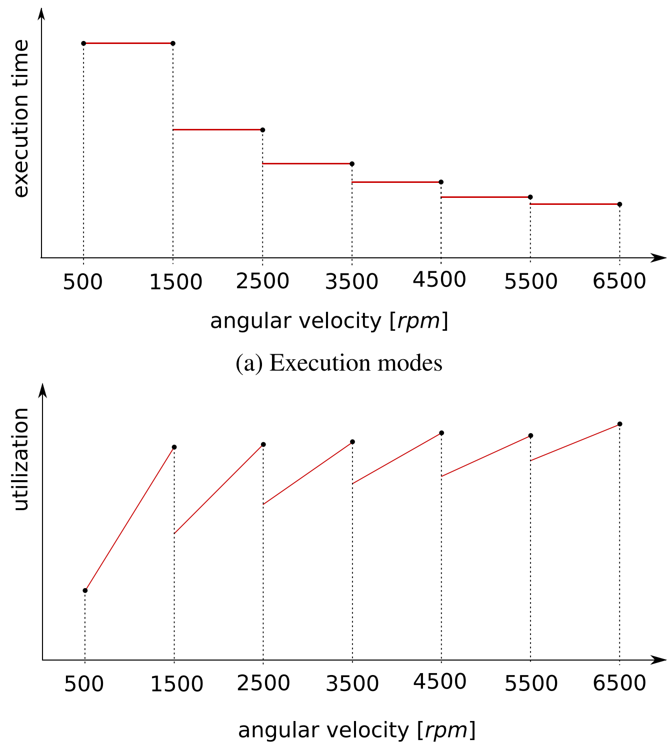
Real-time systems are characterised by the need for both functional and timing correctness. The system must produce the correct responses to input stimuli within specified time constraints or deadlines. Real-time functionality is typically decomposed into a set of tasks that are either activated periodically 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 Pollex et al. (2013a), 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. Fig. 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 Pollex et al. (2013a). 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.

* Corresponding author.
E-mail addresses: 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).

https://doi.org/10.1016/j.jss.2017.12.033
0164-1212/



(a) Execution modes

(b) Utilization

Fig. 1. Characterization of a rate-dependent task.

In his keynote talk Buttle (2012) 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: Section 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 (1973). 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* ($D_i \le T_i$) or *arbitrary* deadlines. Exact schedulability tests for the sporadic task model have been developed for fixed priority scheduling based on response time analysis Audsley et al. (1993) and for EDF scheduling based on the processor demand criterion Baruah et al. (1990). Other extensions to the model include generalised multiframe (GMF) tasks Baruah et al. (1999), 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 Moyo et al. (2010) allows a non-cyclic order of job types. The most general model is the Digraph Real-Time (DRT) task model Stigge et al. (2011), 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 Davis et al. (2014a). 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 Diagraph can capture the transitions between job types, simple instances of the DRT task model are more suitable for modeling rate-dependent tasks Guo and Baruah (2015); 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 Biondi et al. (2015a); 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