Mälardalen University Press Licentiate Theses
No. 190

ON IMPROVING RESOURCE UTILIZATION IN DISTRIBUTED REAL-TIME EMBEDDED SYSTEMS

Meng Liu

2014

MÄLARDALEN UNIVERSITY
SWEDEN

School of Innovation, Design and Engineering

ISBN 978-91-7485-177-9
ISSN 1651-9256
Printed by Arkitektkopia, Västerås, Sweden

Populärvetenskaplig sammanfattning

I vårt vardagliga liv spelar inbyggda system en viktig roll. Ett inbyggt system är ett datorsystem inbyggt i en viss enhet, sak eller produkt, i syfte att uppnå enhetens funktion. För dessa enheter gäller först och främst att den funktionella korrektheten måste garanteras bortom allt tvivel. Ytterligare för flertalet av dessa inbyggda system gäller att även timingen på den funktionella korrektheten spelar en viktig roll för den övergripande korrekta funktionen. Tittar man till exempel på ett bromssystem i en bil så måste själva bromsfunktionen utföras inom en begränsad tid för att undvika att bromssituationen resulterar i en olycka. Sådana inbyggda system där timing är avgörande för korrekt funktion kallas för inbyggda realtidssystem.

Tittar man på vad som finns inuti dessa inbyggda realtidssystem så finns (kör) där ett eller oftast flertalet olika program (kallas tasks) som delar ett begränsat antal datorresurser (t.ex. processorer och minnen). Om systemet skulle köra programmen på ett icke förutsägbart sätt så skulle hela systemet och dess funktionalitet bli oförutsägbart. Man skulle då omöjligen kunna avgöra och verifiera att systemet uppfyller alla de krav som ställts på det för korrekt funktion, t.ex. rörande dess timing. Med andra ord skulle systemet inte kunna garanteras som säkert. Därför måste man konstruera förutsägbara algoritmer som styr över och kontrollerar hur programmen ska köras. Alla program i ett inbyggt system måste schemaläggas. Lösningen är att en schemaläggningsalgoritm schemalägger programmen och en matematisk analys kan givet en viss schemaläggningsalgoritm räkna ut timing hos programmen. På så sätt kan man avgöra om systemet kommer att bete sig korrekt eller inte.

När man använder sig av en viss schemaläggningsalgoritm och medföljande analys så händer det tyvärr ofta att man måste avdela för mycket av det inbyggda systemets resurser till programmen jämfört med vad som verkligen behövs av programmen när programmen körs. Denna överallokering är ett resultat av förenklingar gjorda i den matematiska analysen, förenklingar som är gjorda för att skapa en fungerande analys men på en kostnad av ett effektivt resursutnyttjande. Som ett svar på denna situation så vill vi i denna licentiatavhandling förbättra resursutnyttjandet i moderna distribuerade inbyggda realtidssystem. Vi försöker att lösa denna utmaning från två olika vinklar.

1. Utveckla nya tightare tidsanalyser. På grund av svårigheten i att utföra exakt matematisk schemaläggningsanalys så har de flesta befintliga analyserna en varierande grad av pessimism. Med andra ord kan systemet bete sig mycket bättre när man kör systemet jämfört med de pessimistiska prognoserna. Om vi kan minska pessimismen i schemaläggningsanalysen så kan vi köra de inbyggda systemen med högre last och bättre resursutnyttjande.

2. Utveckla nya schemaläggningsalgoritmer. Det är svårt att hitta en schemaläggningsalgoritm som är lämplig för alla situationer. Därför behöver vi olika schemaläggningsmekanismer för att hantera specifika systemegenskaper i syfte att förbättra resursutnyttjandet.

Abstract

In our modern life, embedded systems are playing an essential role. An embedded system is a computer system embedded into a certain device, in order to achieve computing functions. Beyond all doubt, as a validated system, the functional correctness must be guaranteed. However, for many embedded systems, timeliness also plays an important role in addition to the correctness of the functionalities. For example, in an automotive braking system, the braking function needs to be processed within a limited time duration in order to avoid accidents. Such systems are known as real-time embedded systems.

In these systems, there can be plenty of software programs (called tasks) sharing limited computing resources (e.g. processors, memories). If the system executes tasks in a random way, the whole system will become unpredictable. As a result, the system designers will not be able to verify if the system design can fulfill all the timing requirements or not. In other words, the system is not guaranteed to be safe. Therefore, system designers need to carefully implement algorithms to schedule all the tasks in a predictable manner. Regarding each scheduling algorithm, schedulability analyses are proposed which are used to check if the requirements can be satisfied.

Unfortunately, many real-time systems reserve too much computing resource for the sake of fulfilling timing requirements, without taking into account resource utilization. As a result, system resources cannot be efficiently utilized, which can cause significant resource waste in reality. Therefore, in this thesis, we aim to improve resource utilization in modern distributed real-time embedded systems. We try to tackle this problem from the following two aspects.

1. Investigating tighter timing analyses. Due to the difficulty in performing precise mathematical schedulability analyses, most of the existing analyses include varying degrees of pessimism. In other words, the ac-

tual performance of the system can be much better than the predictions. If we can reduce the pessimism in schedulability analyses, we can then admit more workload into the system.

2. Proposing new scheduling frameworks. It is difficult to find a scheduling algorithm which is suitable for all the situations. Therefore, we need different mechanisms to handle specific system characteristics in order to improve the resource utilization.

To my family

Acknowledgments

First of all, I would like to thank my supervisors Prof. Thomas Nolte and Dr. Moris Behnam. Without your inspirations, guidance and encouragement, I would not be able to produce this licentiate thesis. I also would like to express my appreciation to Prof. Luis Almeida and Prof. Shinpei Kato for the inspirations and cooperations.

Many thanks go to my office-mate Sara Afshar for the discussions and helps. I also would like to thank all former and present colleagues from my research group, Prof. Reinder J. Bril, Daniel Hallmans, Hamid Reza Faragardi, Hang Yin, Matthias Becker, Mikael Åsberg, Mohammad Ashjaei, Nima Khalilzad, Rafia Inam, for inspiring discussions and cooperations.

Further more, I would like to thank all the professors and lecturers at MDH, Emma Nehrenheim, Gordana Dodig-Crnkovic, Hans Hansson, Hongyu Pei-Breivold, Jan Gustafsson, Kristian Sandström, Margaret Obondo and Monica Odlare, from whom I have learned a lot during my graduate study.

Also, I would like to thank all the administrative stuff at IDT, especially Carola Ryttersson and Susanne Fronnå for your helps.

Besides the research work and study, I also had a lot fun during conference trips, badminton, barbecue, carting, cinema, etc. So many thanks also go to all the fun-mates: Abhilash, Adnan, Aida, Alessio, Andreas, Aneta, Batu, Cristina, Dag, Eduard, Elena, Francisco, Gabriel, Gregory, Guillermo, Irfan, Juraj, Kan, Kivanc, Leo, Luka, Mehrdad, Omar, Pablo, Predrag, Saad, Sara Ab., Sara x. Ab., Severine, Simin, Svetlana, Raluca, and others!

Finally, I would like to express my very great appreciation to my family for always encouraging and supporting.

Meng Liu
Västerås, November, 2014

This work has been supported by the Swedish Research Council (Vetenskapsrådet) under the project START.

List of publications

Papers included in the licentiate thesis[1]

Paper A *Schedulability Analysis of GMF-Modeled Messages over Controller Area Networks with Mixed-Queues*. Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 10th IEEE International Workshop on Factory Communication Systems (WFCS), 2014, May.

Paper B *Applying the Peak Over Thresholds Method on Worst-Case Response Time Analysis of Complex Real-Time Systems*, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2013, August.

Paper C *An EVT-based Worst-Case Response Time Analysis of Complex Real-Time Systems*, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 8th IEEE International Symposium on Industrial Embedded Systems (SIES), 2013, June.

Paper D *A Server-based Approach for Overrun Management in Multi-Core Real-Time Systems*, Meng Liu, Moris Behnam, Shinpei Kato, Thomas Nolte. In the Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2014, September.

Paper E *An Adaptive Server-Based Scheduling Framework with Capacity Reclaiming and Borrowing*, Meng Liu, Moris Behnam, Shinpei Kato, Thomas Nolte. In the Proceedings of the 20th IEEE International Conference on

[1]The included articles have been reformatted to comply with the licentiate thesis layout.

Embedded and Real-Time Computing Systems and Applications (RTCSA), 2014, August.

Additional papers, not included in the licentiate thesis

1. *Schedulability Analysis of Mixed-queued Controller Area Networks with Multi-Frame Messages*, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 18th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), WIP, 2013, September.

2. *Response time analysis for static priority based SpaceWire networks*, Meng Liu, Moris Behnam, Thomas Nolte, Luis Almeida. In the Proceedings of the 2nd International Workshop on Worst-Case Traversal Time (WCTT) located at 33rd IEEE Real-Time Systems Symposium (RTSS), 2012, December.

3. *Worst-Case Delay Analysis of Master-Slave Switched Ethernet Networks*, Mohammad Ashjaei, Meng Liu, Moris Behnam, Ahlem Mifdaoui, Luis Almeida, Thomas Nolte, In the Proceedings of the 2nd International Workshop on Worst-Case Traversal Time (WCTT) located at 33rd IEEE Real-Time Systems Symposium (RTSS), 2012, December.

4. *Probabilistic Application Interfaces for Hierarchical Scheduling*, Nima Moghaddami Khalilzad, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 34th IEEE Real-Time Systems Symposium (RTSS), WIP, 2013, December.

Contents

I

Thesis

Chapter 1

Introduction

Distributed real-time embedded systems are widely utilized in our life. Such systems can be found in vehicles, aircrafts, industrial robots, etc. For these systems, functional correctness is not sufficient to guarantee the overall correctness of the system. Additionally, timeliness should also be taken into account. A number of timing requirements are always specified for such a system, where violations of timing requirements can result in varying degrees of impact. In a hard real-time system, all the timing requirements must be strictly satisfied, since requirement violations can result in fatal problems. For example, in a brake-by-wire system, if the function cannot be achieved within a given time duration, serious accidents may happen. On the other hand, in a soft real-time system, some violations of timing requirements can be affordable, since a few violations (i.e. should be less than a certain amount) may only degrade system performance but are still acceptable. For example, in the Graphical User Interface (GUI) control system, a certain delay of displaying status information may not cause a fatal problem, unless the amount of delays exceeds a given bound. Therefore, while developing such systems, the designers need to take into account both the functional correctness and the timeliness. Moreover, the timing behaviors of such systems should be predictable, so that the designers are able to verify if the requirements can be fulfilled.

In most distributed real-time systems, resources are always shared by multiple applications (i.e. tasks or messages). Hence an appropriate scheduling framework is mandatary to enable the predictability of the system. By applying schedulability analyses, designers can predict if the system design is acceptable or if it needs modification regarding the given timing requirements.

However, many real-time systems reserve too much resource in order to guarantee the correct behavior (i.e. the fulfillment of the timing requirements), without taking into account resource efficiency. As a result, system resource cannot be efficiently utilized, which can cause significant resource waste in reality. One reason for waste of resource is due to the pessimism involved in the schedulability analyses. On another hand, the design of some scheduling frameworks only focus on the safety of the system without considering resource utilization as well. Therefore, we tried to tackle the resource wasting problem from the above two aspects.

A distributed real-time system can be considered as an integration of two subsystems: the subsystem inside a processor, and the subsystem of the communication network. In this thesis, we aim to take both subsystems into account. However, for each type of subsystems there are plenty of different implementations with different types of resources (e.g. CPU, cache, etc.). In order to tackle our research problems precisely and efficiently, we narrow down our vision on several specific fundamental system elements. Regarding in-processor systems, we focus on the resource of CPU. With respect to real-time communications, we focus on the resource of transmission media.

Generally, this thesis comprises three contributions targeting improvement of resource utilization:

1. Applying the Generalized Multi-Frame (GMF) task model on Controller Area Network (CAN) messages, and providing a corresponding response time analysis.

2. Applying statistical approaches (Extreme Value Theory) on Worst Case Response Time (WCRT) analysis, which can be utilized even if only partial system information is available.

3. Proposing new scheduling frameworks to handle tasks with large variations of execution times.

1.1 Research Methodology

The research methodology used in this thesis work is conformable to the steps proposed in [1]. The main steps of conducting the research are as follows:

1. Identify the research problems by studying the state of the art and define the research goal.

2. Divide the main research goal into smaller subgoals.
3. Formulate the research questions and challenges by refining the subgoals.
4. Propose proper solutions to the addressed research questions by studying the literature.
5. Provide formal proofs for developed solutions.
6. Analyze and evaluate the proposed solutions by performing simulation-based measurements and case studies.

Steps 3 - 6 are repeated until the desired results are achieved.

1.2 Goal of the Thesis

This thesis targets on improving resource utilization in distributed real-time embedded systems. First of all, we need to find the sources which can cause waste of system resource. Then, targeting each source, we propose corresponding solutions. However, there are many factors which can cause waste of system resource, leading to a giant and general research problem. Therefore, in this thesis, we try to narrow down the above problem, and focus on the following subgoals.

1.2.1 Subgoal I: Providing Tighter Time Analyses

Many system designers use time analyses to verify if the time requirements can be satisfied. As a result, the precision of an analysis can really affect the performance of the analyzed system.

If the analysis provides optimistic results, the designer may admit more workload than the amount that the system can actually afford. Such a design can result in runtime violations of time requirements, which accordingly violates correctness of the system. This is not acceptable for most real-time systems, especially for hard real-time applications.

Unfortunately, acquiring precise analysis results can be very difficult for many real-time applications due to the increasing complexity of their design and construction. Therefore, many existing analyses employ approximations with simplifying pessimistic assumptions, in order to provide guaranteed results. However, due to the included pessimism, the system resource may not be fully utilized, which can cause waste of resource and degrade the system performance.

Therefore, the first subgoal of this thesis is to develop time analyses which include less pessimism.

1.2.2 Subgoal II: Proposing New Scheduling Frameworks

For many real-time systems, the design relies on the estimations of the Worst-Case Execution Times (WCET) of tasks. However, the WCET of tasks can vary a lot during runtime. For example, path planning and perception tasks in mobility robots can have diversity in their execution times, due to the largely varied input information. For these applications, it is difficult to estimate precise WCETs. Alternatively, designers have to use overly estimated WCETs, in order to keep the system design safe. On another hand, for most applications, when we look at the probability distribution of execution times, the actual WCET only occurs with an extremely low probability. In other words, if the design is simply based on the estimated WCETs, the system may reserve a lot of resource which is rarely utilized. For soft real-time applications, such waste of system resource is really unnecessary.

Therefore, the second subgoal is to propose new scheduling frameworks, where we try to improve resource utilization by reducing the effect of pessimistic WCETs.

1.2.3 Research Questions

Following the research goals defined above, a list of research questions are proposed as follows:
Q1: What are the limitations of the existing analyses? How can we improve the accuracy and applicability of the analyses regarding specific limitations?

Q1.1: Traditional Response Time Analysis (RTA) assumes a periodic task/message model, which can simplify the analysis process. However, for many applications, the task arrival pattern does not exactly follow the periodic model. As a result, a transformation process is required, which can cause varying degrees of pessimism. Are there any more generalized models? How should the analyses be performed accordingly?

Q1.2: Most of the existing timing analyses require detailed information of the system. If the system under analysis includes many dynamic factors (i.e.

it is difficult to make precise predictions for these factors), we need to employ many approximations accordingly. The integration of these approximations can result in much pessimism due to the imprecision from each of them. Moreover, in a worse situation, the system may even include hidden factors (e.g. unpredictable behavior due to hardware integrations), which makes the traditional analysis infeasible. How can we provide a suitable schedulability test while only partial system information is available?

Q2: In many industrial applications, task execution times can vary a lot. To predict the WCETs of tasks is also becoming more and more challenging due to the increase of system complexity. As a result, the estimated WCET can be very pessimistic which can result in resource waste, or be optimistic which can degrade the system reliability. How can we deal with tasks with large variations of execution times from the scheduling framework perspective?

1.3 Thesis Overview

In Chapter 2, we briefly introduce the background knowledge of the works included in the thesis. In Chapter 3, we present the main technical contributions of this thesis with respect to the corresponding research goals. A summary of the included works is presented in Chapter 4, together with some prospects of the future work. In Chapter 5, we present an overview of all the included papers, and these papers are presented in Chapter 6-10.

Chapter 2

Background

2.1 Real-Time Systems

A real-time system is a computer system, where the correct behavior depends on both functional correctness and timeliness. In such systems, predictability is an important property, so that the system designers are able to analyze the timing behavior correctly. For an acceptable system design, all the timing requirements need to be fulfilled. Regarding different levels of requirement satisfaction, real-time systems can be categorized into two types: hard real-time systems and soft real-time systems.

In a hard real-time system, all the timing requirements must be strictly satisfied. A violation of the requirements can result in catastrophic consequences. For example, in an automotive system, the brake-by-wire subsystem and airbag subsystem are typical hard real-time applications. Because requirement violations in these systems can cause personal and property losses.

In a soft real-time system, some violations of timing requirements can be tolerated. In such systems, timing violations may degrade the system performance, but cannot cause fatal problems. For example, a multimedia entertainment system is a soft real-time application, since the requirement violation may only degrade the user satisfaction without causing any serious impact.

2.2 Scheduling Algorithms

In order to guarantee the predictability of a real-time system, the tasks[1] need to be executed following certain schedules instead of randomly occupying system resources. The traditional scheduling algorithms can be divided into two categories: static scheduling and dynamic scheduling.

Under static scheduling mechanisms, the schedule of tasks is predefined. The implementation of such a scheduler is simple, since the selection of the running task needs just a table look-up. Task executions also become very deterministic. Moreover, in the context of real-time communication, task executions can be easily adapted to time-slot based network protocols (e.g. TTCAN, TTP/C, TTEthernet, etc.). However, a significant disadvantage of static scheduling is the low flexibility, since the schedule cannot adapt itself regarding runtime changes.

Alternatively, under dynamic scheduling algorithms, the task schedule is determined by runtime information. Such scheduling algorithms can provide high flexibility and efficiency for different types of tasks. Priority-based scheduling is one of the most used scheduling mechanism, where the tasks are scheduled based on their specific priorities. Priority-based scheduling includes fixed-priority (e.g. Rate Monotonic (RM) [2], Deadline Monotonic (DM) [3]) and dynamic-priority (e.g. Earliest Deadline First (EDF) [4]) based scheduling algorithms. Unfortunately, it becomes more difficult to predict the runtime behavior under such algorithms comparing to static scheduling. Therefore, proper schedulability analyses are required.

The above scheduling mechanisms are originally designed for single-core systems. However, as the functional requirements increase, multi-core processors which can provide higher computation capability are becoming more and more popular. There are many different system features between single-core and multi-core systems which can affect the scheduling framework. For example, in a multi-core system, tasks on different cores can be executed in parallel; however, in a single-core system, all the tasks can only be executed sequentially. Therefore, different scheduling algorithms need to be designed regarding the features of multi-core systems. Most of the existing scheduling methods for multi-core systems can be categorized into two types: partitioned and global scheduling. In partitioned scheduling, tasks are statically allocated on different cores. Once a task is assigned to a certain core, it cannot migrate to any other cores. Due to this feature, the tasks on the same core can always be scheduled under single-core scheduling algorithms. Under global

[1] A task is defined as an executable software program.

scheduling (e.g. [5, 6, 7]), tasks are usually controlled in a common queue shared among multiple cores, and a task can migrate from one core to another. Another type of approach which is known as semi-partitioned scheduling has also been developed (e.g. [8, 9]). This type of scheduling is a combination of the partitioned and global scheduling, where some tasks are partitioned on different processors and the other tasks are globally scheduled.

2.3 Response Time Analysis

In order to check whether the given timing requirements of certain systems can be fulfilled, many schedulability analysis methods have been proposed. One of the most widely used approaches for schedulability analysis is Response Time Analysis (RTA) [10]. Using RTA, system designers can approximately calculate the response time (i.e. the time duration since the task instance is released until the end of its actual execution) for each task without running the task with a real implementation. Targeting different timing constraints (i.e. hard or soft), response time analyses can be categorized into two types: deterministic RTA and probabilistic RTA.

Using deterministic RTA (e.g. [10, 11, 12, 13, 14]), we can compute a single upper-bound of response time for each task. A typical timing requirement of a real-time system is that the response time of each task should not exceed a certain bound called deadline. Therefore, by comparing the estimated Worst Case Response Time (WCRT) with the deadline, the schedulability can be determined. Such analysis methods are more suitable for hard real-time applications, since the requirements should be strictly respected.

On the other hand, using probabilistic RTA (e.g. [15, 16]), system designers can calculate a probability distribution of task response times instead of a single upper-bound. Such analyses are more informative and more suitable for soft real-time applications. For these applications, the pessimism included in deterministic RTA is not necessary any more, since requirement violations are tolerable. By checking the distribution provided by probabilistic RTA, we can select different estimations of response times regarding expected probabilities. Therefore, the effects of the pessimism included in WCRT can be reduced. However, most of the probabilistic RTA have a large calculation overhead, which can limit the applicability.

Besides the response time analyses presented above, a number of measurement based timing analyses have also been proposed. Using these analysis methods, system designers first need to collect samples (e.g. actual response times of task instances) from a real system or approved simulator (i.e. the simulator is able to capture the main behaviors of the real system). By analyzing the collected samples with certain statistical tools, the analyses can provide different types of estimations or predictions as results. The most significant benefit of these approaches is that the designers do not need to know all the details of the system (e.g the effects between different system elements). The system under analysis is simply considered as a black box. Unfortunately, these analyses require real system implementations which may not be available during the initial phase of system design.

2.4 Controller Area Network

A distributed real-time embedded system can be considered as an integration of two subsystems: the in-processor subsystem and the real-time communication subsystem. For example, in an automotive system, the brake-by-wire subsystem consists of a number of Electronic Control Units (ECUs) connected by several buses/networks. The software programs inside each ECU form a subsystem. However, many of the programs need to communicate (e.g. exchanging data) with other programs located on different ECUs. Such communications are achieved by the subsystem of real-time networks using different protocols.

Among all the real-time communication protocols, Controller Area Network (CAN) [17] is one of the most widely used. CAN is a broadcast digital bus, which is used in many industrial applications including automotive systems, factory controls, medical devices and also avionics systems [18]. CAN has the features of low cost, deterministic contentions, and built-in error detection and retransmission mechanisms, that makes it attractive for embedded real-time applications.

A number of works regarding schedulability analysis of CAN have been proposed. The first response-time based schedulability analysis of CAN messages is presented by Tindell et al. in [19], which is based on the traditional RTA for task scheduling [10]. In [20], Davis et al. presented a revised schedulability analysis, where the potential optimism in [19] is removed. Later on, many extensions (e.g. [21, 22, 23, 24]) of these seminal works are presented regarding different system factors.

2.5 Task Model

The first task model with efficient schedulability test is the periodic task model proposed by Liu and Layland [2]. This model is observed from the fact that many real-time applications have periodic behaviors. As shown in Figure 2.1.a, a task τ_i under such model can be characterized by three main parameters: the period T_i, the relative deadline D_i and the WCET C_i. This model is quite simple leading to low complexity of schedulability analyses. However, the expressiveness of this model is also low, which limits the applicability in many complex real-time systems.

Later on, the traditional periodic task model is extended to model sporadic tasks [25]. For a sporadic task, the 'period' represents the minimum inter-arrival time between successive task instances (jobs). In [26], the authors propose the Multi-Frame (MF) model, where the repetition pattern of a task is based on several successive jobs instead of one single job. As shown in Figure 2.1.b, there can be multiple jobs within one period (large cycle), while the inter-arrival time between each two successive jobs is fixed. In [27], the authors present the Generalized Multi-Frame model, which is a further generalization of the MF model. As depicted in Figure 2.1.c, in the GMF model, successive jobs within one large cycle can have varied inter-arrival times and relative deadlines.

In order to increase the expressiveness, a number of other generalized task models have also been proposed (e.g. the Recurring Branching task model [28], the Recurring Real-Time (RRT) task model [29], the non-cyclic GMF model [30], etc.). However, as the expressiveness of a task model increases, the difficulty of the corresponding schedulability analysis also goes up [31] (i.e. the efficiency of the analysis decreases).

2.6 Extreme Value Theory

Extreme Value Theory (EVT) is a statistical tool widely utilized in many different disciplines, which is usually used to deal with extreme cases of an event. Instead of analyzing the whole probability distribution of a data set, EVT focuses more on the tails (i.e. extreme deviations from the median of a distribution). EVT has been applied on predicting the size of freak waves, mutational events during evolution, the amounts of large insurance losses, and so on. Similar to the above examples, in a real-time system, the worst-case scenario can also

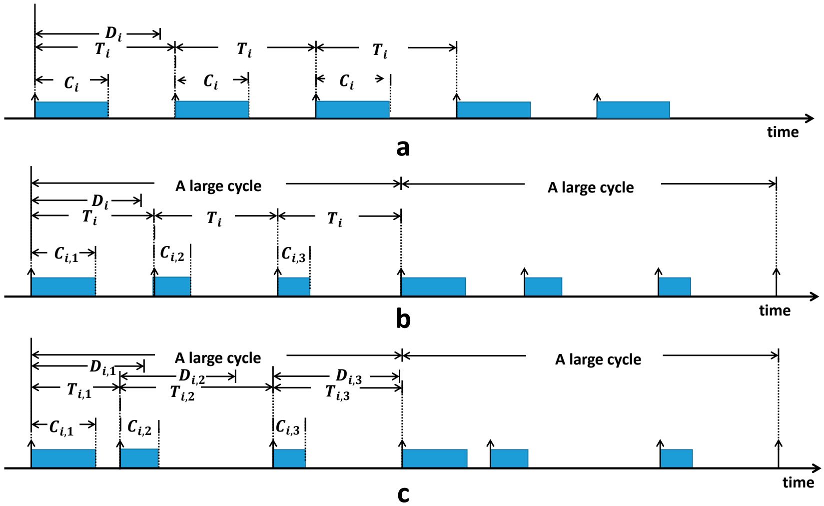

Figure 2.1: Traditional periodic model, MF model and GMF model. Note that T_i in figure a denotes the period of the task; in figure b&c, T_i and $T_{i,x}$ denote the inter-arrival times between successive jobs.

be considered as a rarely occurring event. Therefore, EVT can be an appropriate tool to analyze extreme timing behaviors (i.e. WCETs and WCRTs) of complex real-time systems. The EVT consists of two main approaches: block maxima (minima) and peak over thresholds.

The basic idea of block maxima (minima) is to analyze the sub data set comprised of the maximum (minimum) sample of each data block, where each block contains the same number of samples from the original population. Block maxima refers to the upper tail of the analyzed parent probability distribution, which matches the concept of the worst-case timing behaviors. On the other hand, block minima targets the lower tail of the parent distribution, which relates to the best-case timing behaviors. The basic procedure of the Block Maxima (BM) approach is illustrated in Figure 2.2.

a. Assume that the original data set is $X = \{x_1, x_2, ..., x_N\}$, which contains N independent random samples following the same probability distribution.

b. Divide the data set X into s blocks, where each block X^i contains the same number (i.e. n) of samples.

c. Create a new data set M which consists of the maximum sample m_i of each block X^i.

d. Apply the estimation process on the sub data set M.

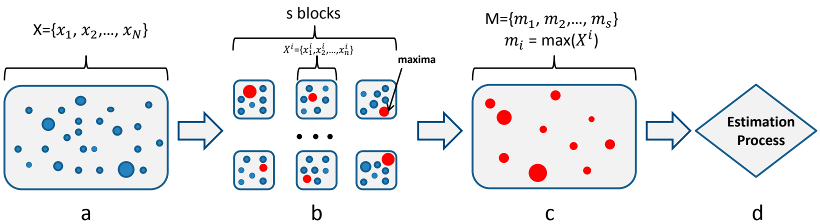

Figure 2.2: The basic procedure of the block maxima method.

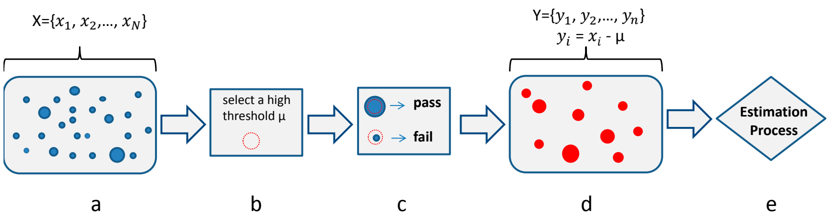

Figure 2.3: The basic procedure of the peak over thresholds method.

The Peak Over Threshold (POT) method relies on a sub data set generated from an original data set, where the samples in this sub data set must exceed a certain threshold or fall below a selected threshold. Figure 2.3 depicts the basic procedure of the POT method.

a. Assume that the original data set is $X = \{x_1, x_2, \ldots, x_N\}$, which contains N independent random samples following the same probability distribution.

b. Select a high threshold μ.

c. Filter the original data set based on the threshold μ. The samples below the threshold μ will be discarded. The remaining n samples form a sub data set X'.

d. Create a new data set Y which is comprised of the exceedance y_i of each sample in X' regarding the threshold μ.

e. Apply the estimation process on the sub data set Y.

2.7 Reservation-Based Scheduling

In many real-time systems, besides periodic tasks which are mostly time triggered and execute control activities, there are also aperiodic tasks which are mostly event-triggered. For example, in an automotive system, the task used to monitor the current speed can be periodic in order to keep up-to-date information. However, the task used to generate the braking signal is aperiodic, since it should only be triggered by a braking event (e.g. when the driver press the pedal). In order to schedule different types of tasks in the same system, Reservation-Based Scheduling (RBS) mechanisms are proposed. The resource reservation is achieved through the use of *servers*. From the scheduler point of view, servers can be considered as a special type of task which is used to execute aperiodic (event-triggered) tasks. A number of server schemes have been proposed in literature. These approaches can be generally divided into two categories: fixed-priority server (e.g. polling server, deferrable server, sporadic server, etc.) and dynamic priority server (e.g. dynamic sporadic server, total bandwidth server, constant bandwidth server, etc.).

A polling server (PS) [32] is similar to a periodic task, which can be characterized by a period T_s and a server capacity C_s. At the beginning of each period, the server capacity is recharged to the maximum. If there is no pending workload, C_s will be discharged directly to 0. If a job arrives just after the beginning of the period (i.e. C_s is just discharged), this job has to wait for the next server period. As the server executes, the capacity C_s is consumed.

A deferrable server (DS) [33] has similar characteristics to a PS, which also has a period T_s and a capacity C_s. The server capacity is also recharged at the beginning of each server period. However, if there is no pending requests at that time, the capacity of the DS can be preserved and available until the end of the period. Therefore, a job arrived after the beginning of the period, does not need to wait for the next period.

Similar to a DS, a Sporadic Server (SS) [32] can also preserve server capacity until it is consumed by requests. However, an SS can recharge its capacity only after it has been consumed, without any periodic replenishment. In [34], the authors propose an extension of SS called Dynamic Sporadic Server (DSS), which is designed for systems using the EDF algorithm. A DSS has almost the same features as an SS, except the priority assignment. The priority for an SS is fixed and decided by certain priority assignment algorithms. However, the priority of a DSS is dynamic and assigned through a suitable deadline.

The Constant Bandwidth Server (CBS) [35] algorithm is another well-known dynamic priority RBS approach. Under the CBS approach, each server is assigned a bandwidth U_s which is a fraction of processor time. The algorithm guarantees that each server cannot contribute more than U_s to the total system utilization. When a job requesting a CBS is released, a suitable deadline (to keep the reserved bandwidth) is assigned to it. If the job wants to execute more than expected, its deadline will be postponed. In other words, the priority of this job will be decreased, and the interference to other tasks can also be reduced.

Chapter 3

Technical Contributions

Targeting the research questions outlined in Chapter 1.2.3, we have three main contributions included in this thesis.

1. Applying the GMF task model on CAN messages, and providing a corresponding RTA.

    - Introduction: Most of the existing analyses for CAN are based on a traditional periodic message model. However, in some applications, the transmission of a message may follow a certain pattern instead of repeating the same transmission period by period. For these applications, a more general message model can improve the precision of the analysis, since the effects of model transformation can be reduced. A number of generalizations of the traditional periodic task model have been proposed (e.g. the Multi-Frame (MF) model [26] and the GMF model [27], the non-cyclic GMF model [30], and the recurring real-time (RRT) task model [29]). A more expressive model has larger complexity, which makes the corresponding schedulability analysis more difficult [31]. Therefore, we tackle the above problem step-by-step by starting with the MF model and the GMF model which have relatively lower analysis complexity. We propose the first work that applies the GMF model on CAN messages, together with the corresponding response time analysis.
    - These contributions are presented in Paper A, targeting the research question Q1.1.

2. Applying statistical approaches on WCRT analyses, which can be utilized even if only partial system information is available.

   - Introduction: Most of the existing analyses rely on static and detailed system information. However, sometimes only partial information of a system can be available; or it may require too much effort on obtaining those details, making those analysis methods much less feasible. Moreover, most of those methods only focus on some specific system models with unrealistic assumptions. Consequently, applying those methods on a complex industrial real-time system may result in overly pessimistic results. In order to tackle this problem, we propose two statistical methods to compute WCRTs of tasks/messages in complex real-time systems regarding soft timing constraints. The proposed methods can provide a higher general applicability with less required system information. These methods are based on the Extreme Value Theory (EVT). The EVT is a statistical tool widely utilized in many different disciplines, which is usually used to analyze rarely occurred events. In a real-time system, the occurrence of WCRTs can also be considered as extreme scenarios. Therefore, we select the EVT as the fundamental approach for our analyses.
   - These contributions are presented in Paper B and C, targeting the research question Q1.2.

3. Proposing new scheduling frameworks to handle tasks with large variations of execution times.

   - Introduction: Reservation-based scheduling frameworks are appreciated for many real-time applications. In these frameworks, task executions can be isolated by introducing the concept of servers, which can integrate multiple types of tasks into one system. However, due to the variation of task execution times, the predefined server capacities may not be appropriate (i.e. optimistic or overly pessimistic) during runtime. An overrun can occur when the server capacity is smaller than the real execution time. In Paper D, we proposed a scheduling framework for multi-core systems to handle task overruns. On another hand, due to the imprecision of the estimated WCET, it is very difficult to guarantee the reliability of the defined server capacities. Therefore, in Paper E, we present an adaptive scheduling framework, where each server can adapt itself according to runtime statistics.

- These contributions are presented in Paper D and E, targeting the research questions Q2 while partially covering Q1.1 and Q1.2 as well.

My Contribution

I was the main driver of all the works included in this thesis. The co-authors contributed by discussions and reviewing the papers.

Chapter 4

Conclusions

4.1 Summary and Conclusions

In this thesis, we try to improve the utilization of real-time embedded systems from two perspectives: (1) developing tighter time analyses; (2) proposing new scheduling frameworks which can decrease the effects of pessimistic analyses.

In paper A, we apply the GMF model on CAN messages and propose a corresponding response time analysis. The evaluation results show that the proposed analysis performs better (i.e. with less pessimism) than the traditional RTA regarding GMF-modeled messages. The main reason is that the pessimism caused by transformation of message models can be reduced while applying more generalized models.

In Paper B and C, we apply the extreme value theory on the time analysis of CAN transmissions. The two main approaches of EVT, Block Maxima and Peak Over Threshold, are both utilized. The proposed analyses can be utilized even when only partial system information is available. According to the evaluation results in Paper C, the BM-based analysis can provide tighter results (i.e. with less pessimism) comparing to the traditional deterministic RTA. In Paper B, we compare the performance between the POT-based method and the BM-based approach. As observed from the evaluation results, given the same exceedance probability of a random sample from the parent data set, the POT-based approach usually provides tighter results. The main reason is that when the sample size is small, the estimation uncertainties for the BM-based approach can be very large which affects the accuracy of the results. However, the POT-based approach may include more overhead than the BM-based approach, due to the threshold selection process. Therefore, a trade-off between accuracy and the overhead of the analysis needs to be considered. In general, the traditional RTA is more sensitive to uncertainties of different system factors, because the precision of the results relies on the accuracy of the approximations during the analysis. However, using an EVT-based approach, the analysis is based on measurement samples, which can avoid the effects of the above approximations. Moreover, the system under analysis is simply considered as a black box. Therefore, such analysis methods can be easily applied on different system models. Unfortunately, these measurement-based approaches require real system implementations.

In Paper D and E, we proposed two new scheduling frameworks in order to handle tasks with large variations of execution times. The framework presented in Paper D focuses on multi-core systems. The execution of each task is divided into two parts: a regular execution and an overrun. We use partitioned scheduling algorithms for regular task executions, while the overruns are globally scheduled. The framework uses deferrable servers to execute task overruns, which can isolate overruns from regular task executions. The evaluation results show that this framework can provide higher schedulability comparing to the frameworks using only local servers to handle overruns or without any overrun management. However, in this framework, we still assume that the size of the overrun is bounded and known. As we discussed in the previous sections, when a task has large variations of execution times, predicting the WCET can be very difficult. As a result, if the predictions of WCETs are not correct, the system performance can be significantly impacted. Regarding this concern, we propose another scheduling framework in Paper E, where no WCET predictions is required. This framework uses a reservation-based scheduling policy, which has three main features: capacity adaptation, capacity reclaiming and capacity borrowing. According to the evaluation results, our framework can provide better performance comparing to the related works regarding both prediction accuracy and schedulability.

4.2 Future Work

There are a number of directions that can be considered for future works.

- Besides the GMF model, several other generalized task models have also been proposed (e.g. non-cyclic GMF model [30], recurring real-time (RRT) task model [29], etc.). A more generalized (i.e. more expressive) model can decrease the pessimism included in the time analyses during the model transformation process. As far as we know, applying these generalized models on the analyses of real-time communications has not been well-addressed so far. We would like to investigate other generalized task models in the context of different real-time network protocols.

- Instead of using the above task models, applying a probabilistic model can also benefit many real-time applications with soft time constraints. However, as discussed in [36], developing appropriate probabilistic analysis can be quite challenging. Most of the existing probabilistic analyses rely on a set of unrealistic assumptions, which can significantly restrict the applicability. Therefore, we would like to develop stochastic real-time analyses taking into account different effective system factors (including both software and hardware) step-by-step.

- The proposed WCRT analyses using EVT are measurement-based analyses. One significant limitation of this approach is the overhead caused by sampling. Generally, the more samples we collect, the more precise results we can acquire. However, more collected samples will result in more analysis overhead (i.e. taking more time and storage space). Therefore, we need to further investigate how to balance the trade-off between the precision and the overhead, in order to improve the applicability of the analyses.

- In the evaluations of the proposed scheduling frameworks, we did not consider any practical factors (e.g. migration overhead, cache miss, etc.). The next step is to implement these scheduling frameworks on a real platform, and examine the effects of different practical factors. Further more, we can modify the proposed frameworks according to the observations, in order to achieve better performance.

Chapter 5

Overview of the Papers

5.1 Paper A

Schedulability Analysis of GMF-Modeled Messages over Controller Area Networks with Mixed-Queues, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 10th IEEE International Workshop on Factory Communication Systems (WFCS), May, 2014.

In this paper we apply the Generalized Multi-Frame (GMF) task model on CAN messages, where both priority-based and FIFO-based message queues are taken into account. We also present a corresponding sufficient schedulability analysis. According to the experimental evaluations, our analysis can provide tighter results compared to the existing CAN message response time analysis in the context of GMF-modeled messages. This paper addresses Contribution 1.

5.2 Paper B

Applying the Peak Over Thresholds Method on Worst-Case Response Time Analysis of Complex Real-Time Systems, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 19th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), August, 2013.

In this paper, we propose a statistical method to compute Worst-Case Response Times (WCRTs) of complex real-time systems regarding soft timing constraints, which can provide a higher general applicability with less required system information. Our approach employs a Peak Over Threshold (POT) method, which is a branch of the extreme value theory. For the evaluation, we have applied this approach on the analysis of message transmission latencies over CAN. This paper addresses Contribution 2.

5.3 Paper C

An EVT-based Worst-Case Response Time Analysis of Complex Real-Time Systems, Meng Liu, Moris Behnam, Thomas Nolte. In the Proceedings of the 8th IEEE International Symposium on Industrial Embedded Systems (SIES), June, 2013.

In this paper, we present a Block Maxima (BM) based method to compute worst-case response times targeting complex real-time systems. In the evaluation phase, we have applied this method on the calculation of worst-case transmission delays of messages over CAN, and some comparisons with static RTA are also provided. According to the experimental results, as the system complexity increases, our approach performs much more stable with less pessimism. This paper addresses Contribution 2.

5.4 Paper D

A Server-based Approach for Overrun Management in Multi-Core Real-Time Systems, Meng Liu, Moris Behnam, Shinpei Kato, Thomas Nolte. In the Proceedings of the 18th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September, 2014.

In this paper, we present a server-based scheduling framework for task overrun management in multi-core real-time systems. Instead of assuming a single upper-bound of the WCET for each task, we employ a probabilistic task model. The execution of each task is divided into two segments: regular execution and overrun. The regular executions are scheduled using partitioned scheduling algorithms, while the overruns are globally scheduled on synchronized deferrable servers. We also provide a deterministic WCRT analysis focusing on hard timing constraints, along with a probabilistic analysis of Deadline Miss Ratio (DMR) for soft real-time applications. This paper addresses Contribution 3.

5.5 Paper E

An Adaptive Server-Based Scheduling Framework with Capacity Reclaiming and Borrowing, Meng Liu, Moris Behnam, Shinpei Kato, Thomas Nolte. In the Proceedings of the 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), August, 2014.

In this paper, we present a new reservation based scheduling framework for soft real-time systems using the EDF algorithm (called CARB-EDF). This framework has the features of Capacity Adaptation, Reclaiming and Borrowing. We also present a Chebyshev's inequality based predictor to estimate task execution times. Due to the capacity adaptation mechanism, system designers do not need to provide any estimations of task execution times in order to define proper selections of server capacities. On the other hand, the imprecise runtime predictions of task execution times can be compensated by the capacity reclaiming and borrowing mechanism. The evaluation results show that our framework can provide better performance than the compared related works. This paper addresses Contribution 3.

Bibliography

[1] M. Shaw. The coming-of-age of software architecture research. In *23rd international conference on Software engineering (ICSE)*, 2001.

[2] C. L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1), 1973.

[3] J. Leung and J. Whitehead. On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance Evaluation*, 2(4), 1982.

[4] Massachusetts Institute of Technology. Project MAC. Engineering Robotics Group and ML Dertouzos. *Control robotics: The procedural control of physical processes*. 1973.

[5] B. Andersson, S.K. Baruah, and J. Jonsson. Static-priority scheduling on multiprocessors. In *22nd IEEE Real-Time Systems Symposium (RTSS)*, 2001.

[6] P. Holman and J.H. Anderson. Adapting pfair scheduling for symmetric multiprocessors. *Journal of Embedded Computing*, 1(4), 2005.

[7] J.H. Anderson, V. Bud, and U.C. Devi. An edf-based scheduling algorithm for multiprocessor soft real-time systems. In *17th Euromicro Conference on Real-Time Systems (ECRTS)*, 2005.

[8] J.M. Calandrino, J.H. Anderson, and D.P. Baumberger. A hybrid real-time scheduling approach for large-scale multicore platforms. In *19th Euromicro Conference on Real-Time Systems (ECRTS)*, 2007.

[9] S. Kato and N. Yamasaki. Semi-partitioned fixed-priority scheduling on multiprocessors. In *15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009.

[10] M. Joseph and P. Pandya. Finding response times in a real-time system. In *Computer Journal 29(5)*, 1986.

[11] N. Audsley, M. Burns, A.and Richardson, K. Tindell, and A.J. Wellings. Applying new scheduling theory to static priority pre-emptive scheduling. *Software Engineering Journal*, 8(5), 1993.

[12] L. Sha, T. Abdelzaher, K.E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A.K. Mok. Real time scheduling theory: A historical perspective. *Real-Time Systems*, 28(2-3), 2004.

[13] M. Bertogna and M. Cirinei. Response-time analysis for globally scheduled symmetric multiprocessor platforms. In *28th IEEE International Real-Time Systems Symposium (RTSS)*, 2007.

[14] H. Zhu, S. Goddard, and M.B. Dwyer. Response time analysis of hierarchical scheduling: The synchronized deferrable servers approach. In *32nd IEEE Real-Time Systems Symposium (RTSS)*, 2011.

[15] J.L. Díaz, D.F. García, K. Kim, C.G. Lee, L. Lo Bello, J.M. López, S.L. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. In *23rd IEEE Real-Time Systems Symposium (RTSS)*, 2002.

[16] J.M. López, J.L. Díaz, J. Entrialgo, and D. García. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Real-Time Systems*, 40(2), 2008.

[17] ISO 11898-1. Road vehicles – controller area network (can) – part 1: Data link layer and physical signalling. In *ISO 11898*, 2003.

[18] M. Di Natale. Understanding and using the controller area network. *inst. eecs. berkeley. edu/~ ee249/fa08/Lectures/handout_canbus2. pdf*, 2008.

[19] K.W. Tindell, H. Hansson, and A.J. Wellings. Analysing real-time communications: controller area network (can). In *Real-Time Systems Symposium (RTSS)*, 1994.

[20] R. I. Davis, A. Burns, B. Reinder, and L. Johan. Controller area network (can) schedulability analysis: Refuted, revisited and revised. 35, 2007.

[21] P. M. Yomsi, D. Bertrand, N. Navet, and R.I. Davis. Controller area network (can): Response time analysis with offsets. In *9th IEEE International Workshop onFactory Communication Systems (WFCS)*, 2012.

[22] R.I. Davis, S. Kollmann, V. Pollex, and F. Slomka. Controller area network (can) schedulability analysis with fifo queues. In *23rd Euromicro Conference on Real-Time Systems (ECRTS)*, 2011.

[23] R.I. Davis and N. Navet. Controller area network (can) schedulability analysis for messages with arbitrary deadlines in fifo and work-conserving queues. In *9th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2012.

[24] D.A. Khan, R.J. Bril, and N. Navet. Integrating hardware limitations in can schedulability analysis. In *8th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2010.

[25] A.K. Mok. Fundamental design problems of distributed systems for the hard real-time environment. *PhD Thesis, MIT, Cambridge, Massachusetts*, 2139, 1983.

[26] A.K. Mok and D. Chen. A multiframe model for real-time tasks. In *Real-Time Systems Symposium (RTSS)*, 1996.

[27] S.K. Baruah, D. Chen, S. Gorinsky, and A.K. Mok. Generalized multiframe tasks. *Real-Time Systems*, 17(1), 1999.

[28] S.K. Baruah. Feasibility analysis of recurring branching tasks. In *10th Euromicro Workshop on Real-Time Systems*, 1998.

[29] S.K. Baruah. Dynamic-and static-priority scheduling of recurring real-time tasks. *Real-Time Systems*, 24(1), 2003.

[30] N.T. Moyo, E. Nicollet, F. Lafaye, and C. Moy. On schedulability analysis of non-cyclic generalized multiframe tasks. In *22nd Euromicro Conference on Real-Time Systems (ECRTS)*, 2010.

[31] M. Stigge. Real-time workload models: Expressiveness vs. analysis efficiency. *Ph.D Thesis*, 2014.

[32] B. Sprunt, L. Sha, and J.P. Lehoczky. Aperiodic task scheduling for hard-real-time systems. *Real-Time Systems*, 1(1), 1989.

[33] J.K. Strosnider, J.P. Lehoczky, and L. Sha. The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments. *IEEE Transactions on Computers,*, 44(1), 1995.

[34] M. Spuri and G.C. Buttazzo. Efficient aperiodic service under earliest deadline scheduling. In *15th IEEE Real-Time Systems Symposium (RTSS)*, 1994.

[35] L. Abeni and G.C. Buttazzo. Integrating multimedia applications in hard real-time systems. In *19th IEEE Real-Time Systems Symposium (RTSS)*, 1998.

[36] T. Nolte, M. Liu, and B. Lisper. Challenges with probabilities in real-time analysis. In *5th Real-Time Scheduling Open Problems Seminar (RT-SOPS)*, 2014.