Utilization Control and Optimization of Real-Time Embedded Systems

---

Xue Liu
McGill University
xueliu@cs.mcgill.ca

Xi Chen
McGill University
xi.chen11@mail.mcgill.ca

Fanxin Kong
McGill University
fanxin.kong@mail.mcgill.ca

now
the essence of knowledge
Boston — Delft

Full text available at: http://dx.doi.org/10.1561/1000000042

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

X. Liu, X. Chen, and F. Kong. *Utilization Control and Optimization of Real-Time Embedded Systems*. Foundations and Trends® in Electronic Design Automation, vol. 9, no. 3, pp. 211–307, 2015.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-063-7
© 2015 X. Liu, X. Chen, and F. Kong

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The 'services' for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Foundations and Trends® in Electronic Design Automation

Volume 9, Issue 3, 2015

Editorial Board

Editor-in-Chief

Editors

Editorial Scope

Topics

Foundations and Trends® in Electronic Design Automation publishes survey and tutorial articles in the following topics:

- System level design
- Behavioral synthesis
- Logic design
- Verification
- Test
- Physical design
- Circuit level design
- Reconfigurable systems
- Analog design
- Embedded software and parallel programming
- Multicore, GPU, FPGA, and heterogeneous systems
- Distributed, networked embedded systems
- Real-time and cyberphysical systems

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2015, Volume 9, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Foundations and Trends® in Electronic Design
Automation
Vol. 9, No. 3 (2015) 211–307

DOI: 10.1561/1000000042

Utilization Control and Optimization of Real-Time Embedded Systems

Xue Liu
McGill University
xueliu@cs.mcgill.ca

Xi Chen
McGill University
xi.chen11@mail.mcgill.ca

Fanxin Kong
McGill University
fanxin.kong@mail.mcgill.ca

Contents

Abstract

Real-time embedded systems have been widely deployed in mission-critical applications, such as avionics mission computing, highway traffic control, remote patient monitoring, wireless communications, navigation, etc. These applications always require their real-time and embedded components to work in open and unpredictable environments, where workload is volatile and unknown. In order to guarantee the temporal correctness and avoid severe underutilization or overload, it is of vital significance to measure, control, and optimize the processor utilization adaptively.

A key challenge in this mission is to meet real-time requirements even when the workload cannot be accurately characterized *a priori*. Traditional approaches of worst-case analysis may cause underutilization of resources, while Model Predictive Control (MPC) based approaches may suffer severe performance deterioration when large estimation errors exist. To address this challenging problem and provide better system performance, we have developed several important online adaptive optimal control approaches based on advanced control techniques. Our approaches adopt Recursive Least Square (RLS) based model identification and Linear Quadratic (LQ) optimal controllers to guarantee that the systems are neither overloaded, nor underloaded. These proposed approaches, as well as the associated tools, can quickly adapt to volatile workload changes to provide stable system performance. To minimize the impact of modeling errors, we adopt the Adaptive Critic Design (ACD) technique and develop an improved solution that requires little information of the system model. To deal with the discrete task rates, we further propose to utilize the frequency scaling technique to assist the utilization control and optimization.

The computational overhead of centralized approaches explodes as the scale of systems increases. To ensure system scalability and global stability, decentralized control and optimization approaches are desired. We leverage an efficient decoupling technique and derive several distributed approaches. These approaches adopt one feedback loop to adjust the task rate, and apply another feedback loop to control the CPU frequency asynchronously. As these two manipulated variables (i.e., the

CPU frequency and task rate) contribute to the system performance together with a strong coupling, asynchronous control approaches may not be able to achieve the optimal performance. To handle this coupling, we further develop a synchronous rate and frequency control and optimization approach. This approach jointly and synchronously adjusts rate and frequency settings, and achieves enhanced system performance.

All the aforementioned approaches are based on certain mathematical models. However, it is sometimes hard to develop an exact model to characterize a real-time embedded system. In order to deal with this issue, we further develop a model-free utilization control and optimization solution by applying the fuzzy logic control theory. The application of this theory allows us to achieve the desired performance in a nonlinear dynamic system without a specific system model. The proposed fuzzy utilization control approaches are stable and fast-converging, and achieve smaller tracking errors than model-based approaches.

X. Liu, X. Chen, and F. Kong. *Utilization Control and Optimization of Real-Time Embedded Systems*. Foundations and Trends® in Electronic Design Automation, vol. 9, no. 3, pp. 211–307, 2015.
DOI: 10.1561/1000000042.

1

Introduction

1.1 Real-Time Embedded Systems

A **real-time** system is required to accomplish jobs and deliver services on a timely basis. To function correctly, these systems must provide the correct values of computation within precise time constraints. Examples of real-time systems include Multiprocessor System-On-Chip (MPSoC), media streaming systems, video games, automotive electronics, aircraft control systems, telecommunications, robotics, sensor networks and etc. In real-time systems, a unit of work scheduled or executed by a system is called a job. A real-time task is then defined as a set of related jobs together providing a certain function. One key parameter, the deadline, distinguishes real-time jobs from non-realtime ones. A real-time task is required to be completed before the deadline. Depending on types of systems, missing deadlines will cause performance losses or even complete failures of the real-time systems. Another important concept is the release time, which is the time when a real-time job becomes available for execution. The release time and the deadline together can specify a timing constraint of a real-time job.

The timing constraints are generally divided into two types: hard and soft. There are several different definitions of hard and soft timing constraints. In this article, we adopt a well-known and widely used definition as follows [45].

- A real-time constraint is hard, if violating this constraint is considered as a fatal fault and may cause serious consequences.
- A real-time constraint is soft, if meeting this constraint is desirable, but missing this constraint does not seriously damage the system behavior.

In this article, we mainly focus on the soft real-time systems.

Many real-time systems are embedded as part of a complete device to deliver specified real-time services with limited resources. We call these systems real-time embedded systems. Formally, an **embedded system** is a computer system that is designed to perform a few dedicated functions [42]. Compared with general-purpose computer systems, embedded systems have several distinct features including low energy consumption, small size, low per-unit price, and limited processing power. The real-time embedded systems preserve the features of both real-time systems and embedded systems. They are enabling components of a large number of real-life applications, including automotive control, car navigation, robot control, patient monitoring, wireless communications, sensor networks, gaming electronics and etc.

As the key components of these applications, real-time embedded systems often function in open and unpredictable environments. In such environments, the sensing, computing and control workload is volatile and unknown. Meanwhile, real-time embedded systems are required to accomplish tasks with limited resources. Therefore, it is of crucial importance to to measure, control and optimize the utilization of resources adaptively. Otherwise, the temporal correctness of the systems will be violated, and the resources will be severely underutilized or overloaded.

1.2 Improving Reliability and Performance of Real-Time Embedded Systems

In this section, we discuss several kinds of approaches that are widely adopted to improve the reliability and performance of real-time embedded systems. Note that all the following approaches can be integrated with utilization control and optimization to achieve enhanced reliability and improved performance.

1.2.1 Monitoring Real-Time Embedded Systems

A variety of real-time monitoring schemes assist this goal of improving reliability and performance by collecting measurement results of critical system metrics. The functions or metrics being monitored include, but are not limited to, Quality of Service (QoS), resource utilization, power, energy and temperature. For example, Tedesco et al. in [65] proposed a QoS-aware monitoring scheme, which monitors congestion events among subsystems and applies adaptive routing decisions for the on-chip traffic. Cherkasova et al. in [14] developed a scheme to measure the processor utilization and performance loss due to visualization for parallel I/O. Isci et al. in [30] designed an approached to monitor the run-time power consumption of processors. Merkel et al. [51] proposed a metric named task activity vector to enhance temperature monitoring and temperature-aware scheduling.

More detailed discussions of real-time monitoring schemes can be found in [37]. Based on the real-time monitoring results, different kinds of advanced control and optimization approaches can be applied to the software and hardware of real-time embedded systems. Among these approaches, task scheduling, Dynamic Voltage and Frequency Scaling (DVFS), and utilization control and optimization are widely adopted schemes.

1.2.2 Scheduling Real-Time Embedded Tasks

Scheduling approaches for different platforms and tasks have been extensively studied. For general real-time systems, a number of classic real-time scheduling algorithms have been developed for better system performance. For example, Stankovic et al. in [64] leveraged feedback control to develop a scheduling algorithm, which meets the performance requirements without accurate knowledge of task execution parameters. Goel et al. in [24] utilized a feedback technique to automatically infer system requirements, which makes it possible to design real-time scheduling mechanisms for general-purpose systems that have unknown real-time constraints. Lin et al. in [44] developed a feedback-based scheduling algorithm that maintains a desired deadline miss ratio while keeping the processor utilization as high as possible. Lu et al. in [47] proposed a feedback control real-time scheduling (FCS) framework, which establishes dynamic models and performance analyses of FCS algorithms. It can guarantee robust system performance even when the task execution time varies as much as 100% from initial estimation. Anderson et al. in [1] designed an EDF scheduling algorithm to ensure bounded deadline tardiness in soft real-time systems.

In order to support real-time embedded systems, significant improvements over classic scheduling approaches have been made, by considering distinct features of these systems. These features include, but are not limited to, close coordination among multiple processing units, parallel processing, limited energy, and vulnerability to high temperature. For example, Lakshmanan et al. in [40] addressed the challenge brought by parallel programming models to MPSoCs, and proposed a partitioned preemptive fixed-priority scheduling algorithm that greatly improves the processing speed. Goossens et al. in [26] developed an efficient Earliest Deadline First (EDF) scheduling algorithm for periodic tasks in multiprocessor systems.

As for energy consumption, Aydin et al. in [3] proposed an online power-aware scheduling algorithm for periodic hard real-time tasks, which saves 50% of energy over static algorithms and meets all the deadlines at the same time. Xu et al. in [73] minimized the energy consumption while satisfying the throughput and response time requirement with an energy-aware scheduling algorithm in the context of content streaming. Bruns et al. in [5] considered the proportional fairness in power-aware scheduling, and developed an approach to guarantee the availability of sufficient energy for all real-time tasks. To avoid reliability degradation due to thermal hot spots and high temperature gradients on MPSoCs, Coskun et al. in [15] proposed a temperature-aware scheduling algorithm.. To address the aging issue, Huang et al. in [29] developed a lifetime- and reliability-aware task allocation and scheduling scheme.

More comprehensive overviews of recent advances on scheduling in real-time embedded systems can be found in [16, 6, 32].

1.2.3 Dynamic Voltage and Frequency Scaling in Real-Time Embedded Systems

DVFS is another powerful tool in improving the performance of real-time embedded systems, especially the energy efficiency. For instance, Pillai et al. in [54] considered deadlines and periodicity of real-time tasks, and developed a class of real-time dynamic voltage scaling (RT-DVS) schemes for low-power embedded systems. The RT-DVS schemes maintain real-time guarantees while reducing energy consumption by 20% to 40%. Saewong et al. in [61] minimized the energy consumption while guaranteeing the schedulability of the real-time tasks in CMOS circuits with discrete operating frequencies. By adopting event-driven control updates, Durand et al. in [19] designed a DVFS approach for MPSoCs with distributed and asynchronous clocks. Kahng et al. in [34] investigated the observation that energy overheads of energy-constrained systems is still relatively high in low-power modes. They leveraged the intrinsic characteristics of the hardware design, and developed a context-aware DVFS design flow to further improve the efficiency of DVFS.

Recently, DVFS has been utilized to control the temperature of processing units, since the power consumption, performance and lifetime of embedded systems are all highly related to temperature. Hanumaiah et al. in [27] presented a method that satisfies both hard real-time and temperature constraints on multi-core processors. Durand et al. in [18] analyzed the nonlinearity between power and temperature, and implemented a chopped scheme to limit the temperature increase. Kim et al. in [35] studied the performance loss caused by DVFS-based dynamic management, and proposed a temperature-aware DVFS for mobile devices that saves 12.7% energy consumption.

DVFS is sometimes combined with scheduling to achieve better system performance. For example, Zhu et al. in [79] integrated an EDF scheduler with a DVFS controller to prolong battery life of embedded systems. Gerards et al. in [23] studied the relation between globally synchronous DVFS and scheduling in multi-core processors. and further increases the energy efficiency over local DVFS schemes. Islam et al. in [31] adopted a reinforcement learning-based approach to adaptively select the optimal DVFS algorithm from a set of algorithms, so as to achieve optimal performance under different conditions.

Interested readers can refer to surveys [52, 36] for more extensive discussions on recent advances in DVFS.

1.2.4 Other Control and Optimization Approaches in Real-Time Embedded Systems

Besides system monitoring, task scheduling and DVFS, other approaches have also been actively studied. For example, Dubach et al. in [17] a predictive model for micro architectural adaptivity control, which adjusts the hardware architecture at runtime to fulfill the needs of different applications. Sharifi et al. in [62] applied a Proportional Integral Derivative (PID) controller to increase communication throughput across a network-on-chip (NoC) based multi-core system. Rafiliu et al. in [58] studied the issue of large workload variations, and derived the stability conditions of real-time resource management.

It is worth noticing that all the above approaches (including monitoring, scheduling, DVFS and other approaches) can be combined with utilization control and optimization to further boost the reliability and performance of real-time embedded systems. Furthermore, compared with scheduling and DVFS, utilization control and optimization achieves more benefits, including better resource utilization and more stable performance in the presence of unpredictable disturbances. An overview of utilization control and optimization is to be presented in the next section.

1.3 Utilization Control and Optimization in Real-Time Embedded Systems

Conventional real-time systems functioning in a closed environment mainly rely on worst-case analysis. This tool provides offline timeliness guarantees by scheduling real-time tasks based on conservative estimations of their execution time. However, this is not suitable for real-time embedded systems in open and unpredictable environments, which bring workload uncertainties and estimation errors on execution time. As a result, the traditional approach of worst-case analysis leads to severe under-utilization and wasted resources.

To enforce the desired utilization, task rate based utilization control has been leveraged in the design. Lu et al. in [48] developed an end-to-end utilization control (EUCON) algorithm based on Model Predictive Control (MPC) technique. EUCON applies a Multiple-Input-Multiple-Output (MIMO)s model to describe the utilization control as a multi-variable constrained optimization problem. Experimental results demonstrated that EUCON is more effective and efficient than existing solutions based on worst-case analysis. Wang et al. in [71] focused on distributed real-time systems, and proposed the decentralized end-to-end utilization control (DEUCON) algorithm. To control and optimize the distributed systems, DEUCON requires only localized coordination among neighbor processors. It is scalable and robust even in large-scale distributed systems with varying task execution times. Wang et al. in [67] developed a real-time utilization control middleware, which adopts task rate adaptation to handle variations in application workload and system resources. Fu et al. in [22] focused on cluster-based soft real-time applications, and developed a utilization control algorithm DUC-LB for large-scale server clusters. Koutsoukos et al. in [38] considered a scenario where control variables are discrete. The proposed hybrid supervisory utilization control (HySUCON) algorithm enforces utilization bounds by adaptively switching between the discrete configurations. Chen et al. in [13] studied the computation overhead of discrete task rate adaption, and proposed a multi-parametric rate adaptation (MPRA) algorithm that conducts the online computation in polynomial time. Luiz et al. in [49] developed a model based on approximate system state identification, so as to reduce the negative impact of prediction errors in utilization control.

The above work on utilization control and optimization mainly applies the tool of Model Predictive Control (MPC). MPC-based approaches manage to establish analytical models of the real-time tasks and the underlying systems. These models are then adjusted online based on runtime system variations, and are applied to control and optimize resource utilization. For example, based on a model of task execution time, we can keep processor utilization to be under the schedulable utilization bound while still providing real-time guarantees. However, MPC-based approaches fail to handle large estimation errors that are commonly found in open environments. For instance, if the estimated execution time is much smaller than the actual one, the performance of the system may degrade greatly, or even the whole system may become unstable. On the other hand, overestimating the execution time may result in under-utilization of the processor and a slow convergence of the system.

To handle the highly dynamic workload in an open environment, this article presents several important online adaptive optimal control approaches based on advanced control techniques. In order to guarantee that the systems are neither overloaded nor underloaded, our approaches adopt Recursive Least Square (RLS) based model identification and Linear Quadratic (LQ) optimal controller. To provide a good system performance, these proposed approaches quickly adapt to workload changes. We further improve these approaches with the Adaptive Critic Design (ACD) and the frequency scaling techniques, so as to handle the issues of modeling errors and discrete task rates. We also design distributed approaches to ensure system scalability and global stability and reduce computational overhead. In order to minimize the impact of modeling errors, we further apply the tool of fuzzy logic to develop a model-free solution.

Besides the challenge of unpredictable workload, the use of the technology of MPSoCs in modern real-time embedded system also increases the complexity of utilization control and optimization. Although the MPSoC technology significantly raises the computing power of real-time embedded systems, the utilizations of multiple distributed processors need to be controlled and optimized jointly. In addition, a real-time application usually consists of multiple tasks, each of which may be dispatched to a different processor. Therefore, the whole system is a Multiple-Input-Multiple-Output (MIMO) system. The problem of utilization control and optimization becomes a MIMO control and optimization problem. [1]

1.4 Road Map

The road map of this article is presented in Fig. 1.1.

In Chapter 2, we study the centralized utilization control and optimization in real-time embedded systems. We first present the control model used to characterize a MIMO system. We then discuss how to online adjust the model parameters according to real-time variations of the system. With this online model, we apply an optimal controller to embrace the self-tuning ability of adaptive control and optimization in the design. By optimizing a quadratic cost function, we are able to achieve a good tradeoff between input variations and system performance. We further tackles the issues of modeling errors and discrete task rates by applying the ACD technique and frequency scaling, respectively.

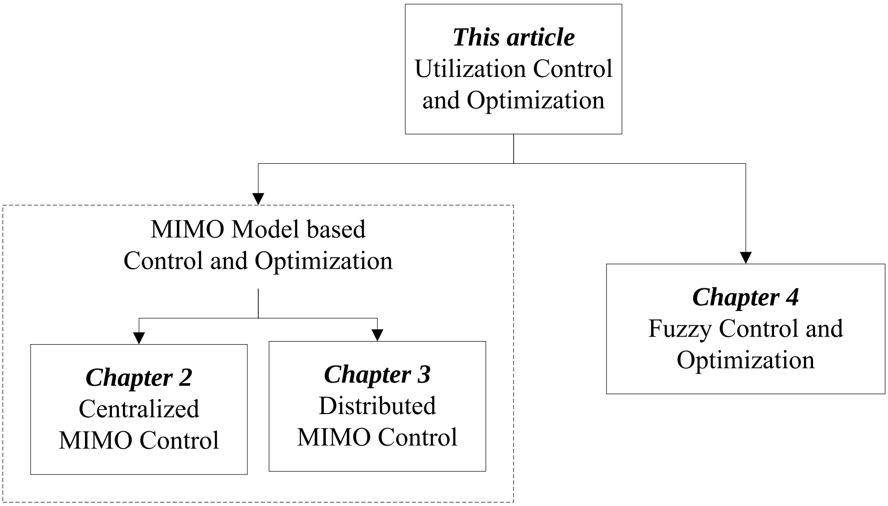

Figure 1.1: The road map of this article.

---

[1] Note that in this article, the term MIMO refers to the multiple control inputs and outputs of a control model or system. This is different from the concept of multiple antennas and multiple I/O data streams in wireless communications.

In Chapter 3, in order to ensure system scalability and global stability, we develop distributed approaches for utilization control and optimization. A communication technique is designed for distributed subsystems to share information. An efficient decoupling technique is leveraged to adjust the task rate and processor frequency asynchronously and in a decentralized manner. To achieve enhanced system performance and increased energy efficiency, we further develop an approach to jointly and synchronously adjust rate and frequency settings.

In Chapter 4, we apply fuzzy logic control theory to develop a model-free utilization control and optimization approach that requires no mathematical model. With this approach, we can minimize the errors of mathematical models in characterizing a real-time embedded system. The application of fuzzy logic control allows us to achieve the desired performance in a nonlinear dynamic system, and achieves smaller control and tracking errors than mathematical model based approaches. In Chapter 5, we summarize this article.

References

[1] J. H. Anderson, V. Bud, and U. C. Devi. An EDF restricted-migration scheduling algorithm for multiprocessor soft real-time systems. *Real-Time Systems*, 38(2):85–131, 2008.

[2] O. Ardakanian, C. Rosenberg, and S. Keshav. Distributed control of electric vehicle charging. In *Proceedings of the 4th International Conference on Future Energy Systems*, e-Energy 2013, pages 101–112, 2013.

[3] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez. Dynamic and aggressive scheduling techniques for power-aware real-time systems. In *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, RTSS 2001, pages 95–105, 2001.

[4] C. Basaran, M. H. Suzer, K.-D. Kang, and X. Liu. Robust fuzzy CPU utilization control for dynamic workloads. *Journal of Systems and Software*, 83(7):1192–1204, 2010.

[5] F. Bruns, S. Bruggemann, D. Kuschnerus, and A. Bilgic. Energy-driven proportional fair scheduling for industrial measurement devices. In *Proceedings of the 19th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS 2013, pages 195–204, Apr 2013.

[6] G. Buttazzo, M. Bertogna, and G. Yao. Limited preemptive scheduling for real-time systems. a survey. *IEEE Transactions on Industrial Informatics*, 9(1):3–15, Feb 2013.

[7] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1):427–438, Feb 2013.

[8] C. Cecati, F. Ciancetta, and P. Siano. A multilevel inverter for photovoltaic systems with fuzzy logic control. *IEEE Transactions on Industrial Electronics*, 57(12):4115–4125, Dec 2010.

[9] H. Chaoui and P. Sicard. Adaptive fuzzy logic control of permanent magnet synchronous machines with nonlinear friction. *IEEE Transactions on Industrial Electronics*, 59(2):1123–1133, Feb 2012.

[10] J. Chen, R. Tan, G. Xing, X. Wang, and X. Fu. Fidelity-aware utilization control for cyber-physical surveillance systems. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1739–1751, 2012.

[11] J.-J. Chen, C.-M. Hung, and T.-W. Kuo. On the minimization of the instantaneous temperature for periodic real-time tasks. In *Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium*, RTAS 2007, pages 236–248, 2007.

[12] X. Chen, X.-W. Chang, and X. Liu. SyRaFa: Synchronous rate and frequency adjustment for utilization control in distributed real-time embedded systems. *IEEE Transactions on Parallel and Distributed Systems*, 24(5):1052–1061, May 2013.

[13] Y. Chen, C. Lu, and X. Koutsoukos. Optimal discrete rate adaptation for distributed real-time systems. In *Proceedings of the 28th IEEE International Real-Time Systems Symposium*, RTSS 2007, pages 181–192, 2007.

[14] L. Cherkasova and R. Gardner. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In *Proceedings of the 2005 Annual Conference on USENIX Annual Technical Conference*, ATEC 2005, pages 387–390, 2005.

[15] A. Coskun, T. Rosing, K. Whisnant, and K. Gross. Static and dynamic temperature-aware scheduling for multiprocessor SoCs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 16(9):1127–1140, Sep 2008.

[16] R. I. Davis and A. Burns. A survey of hard real-time scheduling for multiprocessor systems. *ACM Computing Surveys*, 43(4):35:1–35:44, Oct 2011.

[17] C. Dubach, T. Jones, E. Bonilla, and M. O'Boyle. A predictive model for dynamic microarchitectural adaptivity control. In *Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 2010, pages 485–496, Dec 2010.

[18] S. Durand and S. Lesecq. Nonlinear and asymmetric thermal-aware DVFS control. In *Proceedings of the 2013 European Control Conference*, ECC 2013, pages 3240–3245, Jul 2013.

[19] S. Durand, S. Lesecq, E. Beigne, and D. Puschini. Event-based DVFS control in GALS-ANoC MPSoCs. In *Proceedings of the 2013 American Control Conference*, ACC 2013, pages 5074–5079, 2013.

[20] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid - the new and improved power grid: A survey. *IEEE Communications Surveys Tutorials*, 14(4):944–980, 2012.

[21] X. Fu, K. Kabir, and X. Wang. Cache-aware utilization control for energy efficiency in multi-core real-time systems. In *Proceedings of the 23rd Euromicro Conference on Real-Time Systems*, ECRTS 2011, pages 102–111, Jul 2011.

[22] Y. Fu, H. Wang, C. Lu, and R. Chandra. Distributed utilization control for real-time clusters with load balancing. In *Proceedings of the 27th IEEE International Real-Time Systems Symposium*, RTSS 2006, pages 137–146, 2006.

[23] M. Gerards, J. Hurink, and J. Kuper. On the interplay between global DVFS and scheduling tasks with precedence constraints. *IEEE Transactions on Computers*, 64(6):1742–1754, Jun 2015.

[24] A. Goel, J. Walpole, and M. Shor. Real-rate scheduling. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS 2004, pages 434–441, May 2004.

[25] G. Goodwin, S. Graebe, and M. Salgado. *Control System Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[26] J. Goossens, S. Funk, and S. Baruah. Priority-driven scheduling of periodic task systems on multiprocessors. *Real-Time Systems*, 25(2-3):187–205, 2003.

[27] V. Hanumaiah and S. Vrudhula. Temperature-aware DVFS for hard real-time applications on multicore processors. *IEEE Transactions on Computers*, 61(10):1484–1494, Oct 2012.

[28] J. Hu and G. Feng. Distributed tracking control of leader-follower multiagent systems under noisy measurement. *Automatica*, 46(8):1382–1387, 2010.

[29] L. Huang, F. Yuan, and Q. Xu. Lifetime reliability-aware task allocation and scheduling for MPSoC platforms. In *Proceedings of the 2009 Design, Automation Test in Europe Conference Exhibition*, DATE 2009, pages 51–56, Apr 2009.

[30] C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: methodology and empirical data. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 2003, pages 93–104, Dec 2003.

[31] F. Islam and M. Lin. Hybrid DVFS scheduling for real-time systems based on reinforcement learning. *IEEE Systems Journal*, PP(99):1–10, 2015.

[32] P. Ittershagen, P. Hartmann, K. Gruttner, and A. Rettberg. Hierarchical real-time scheduling in the multi-core era - an overview. In *Proceedings of the 16th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, ISORC 2013, pages 1–10, Jun 2013.

[33] C. Jalier, D. Lattard, G. Sassatelli, P. Benoit, and L. Torres. Flexible and distributed real-time control on a 4G telecom MPSoC. In *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, ISCAS 2010, pages 3961–3964, 2010.

[34] A. Kahng, S. Kang, R. Kumar, and J. Sartori. Enhancing the efficiency of energy-constrained DVFS designs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(10):1769–1782, Oct 2013.

[35] J. M. Kim, Y. G. Kim, and S. W. Chung. Stabilizing CPU frequency and voltage for temperature-aware DVFS in mobile devices. *IEEE Transactions on Computers*, 64(1):286–292, Jan 2015.

[36] J. Kong, S. W. Chung, and K. Skadron. Recent thermal management techniques for microprocessors. *ACM Computing Surveys*, 44(3):13:1–13:42, June 2012.

[37] G. Kornaros and D. Pnevmatikatos. A survey and taxonomy of on-chip monitoring of multicore systems-on-chip. *ACM Transactions on Design Automation of Electronic Systems*, 18(2):17:1–17:38, 2013.

[38] X. Koutsoukos, R. Tekumalla, B. Natarajan, and C. Lu. Hybrid supervisory utilization control of real-time systems. In *Proceedings of the 11th IEEE Real Time and Embedded Technology and Applications Symposium*, RTAS 2005, pages 12–21, 2005.

[39] R. Kulhavy. Restricted exponential forgetting in real-time identification. *Automatica*, 23(5):589–600, 1987.

[40] K. Lakshmanan, S. Kato, and R. Rajkumar. Scheduling parallel real-time tasks on multi-core processors. In *Proceedings of the 31st IEEE Real-Time Systems Symposium*, RTSS 2010, pages 259–268, Nov 2010.

[41] P. Lama and X. Zhou. Efficient server provisioning with control for end-to-end response time guarantee on multitier clusters. *IEEE Transactions on Parallel and Distributed Systems*, 23(1):78–86, Jan 2012.

[42] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach, Second Edition, http://LeeSeshia.org, ISBN 978-1-312-42740-2, 2015.*

[43] H. Li, H. Liu, H. Gao, and P. Shi. Reliable fuzzy control for active suspension systems with actuator delay and fault. *IEEE Transactions on Fuzzy Systems*, 20(2):342–357, Apr 2012.

[44] S. Lin and G. Manimaran. Double-loop feedback-based scheduling approach for distributed real-time systems. In *High Performance Computing (HiPC)*, volume 2913 of *Lecture Notes in Computer Science*, pages 268–278. 2003.

[45] J. W. Liu. *Real-time Systems*. Prentice Hall, Upper Saddle River, New Jersey, 2000.

[46] X. Liu and F. Kong. Datacenter power management in smart grids. *Foundations and Trends in Electronic Design Automation*, 9(1):1–98, 2015.

[47] C. Lu, J. A. Stankovic, S. H. Son, and G. Tao. Feedback control real-time scheduling: Framework, modeling, and algorithms. *Real-Time Systems*, 23(1/2):85–126, 2002.

[48] C. Lu, X. Wang, and X. Koutsoukos. Feedback utilization control in distributed real-time systems with end-to-end tasks. *IEEE Transactions on Parallel and Distributed Systems*, 16(6):550–561, 2005.

[49] S. Luiz, A. Perkusich, A. Lima, J. Silva, and H. Almeida. System identification and energy-aware processor utilization control. *IEEE Transactions on Consumer Electronics*, 58(1):32–37, Feb 2012.

[50] Y. Ma, W. Jiang, N. Sang, and X. Zhang. ARCSM: A distributed feedback control mechanism for security-critical real-time system. In *Proceedings of the 10th IEEE International Symposium on Parallel and Distributed Processing with Applications*, ISPA 2012, pages 379–386, 2012.

[51] A. Merkel and F. Bellosa. Task activity vectors: A new metric for temperature-aware scheduling. In *Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems*, Eurosys 2008, pages 1–12, 2008.

[52] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys*, 46(4):47:1–47:31, Mar 2014.

[53] K. M. Passino and S. Yurkovich. *Fuzzy Control*. Addison Wesley, Boston, MA, USA, 1998.

[54] P. Pillai and K. G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, SOSP 2001, pages 89–102, 2001.

[55] H. Pourshaghaghi and J. de Gyvez. Dynamic voltage scaling based on supply current tracking using fuzzy logic controller. In *Proceedings of the 16th IEEE International Conference on Electronics, Circuits, and Systems*, ICECS 2009, pages 779–782, Dec 2009.

[56] D. Prokhorov and D. Wunsch. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 8(5):997–1007, 1997.

[57] J. Quesada, F. Oterino-Echavarri, J. Gil-Garcia, J. Sainz, R. Sebastian, and M. Castro. A platform for real-time fault-tolerant distributed control of replica-determinate inverters. In *Proceedings of the 2015 IEEE International Conference on Industrial Technology*, ICIT 2015, pages 2042–2047, 2015.

[58] S. Rafiliu, P. Eles, and Z. Peng. Stability conditions of on-line resource managers for systems with execution time variations. In *Proceedings of the 23rd Euromicro Conference on Real-Time Systems*, ECRTS 2011, pages 151–161, Jul 2011.

[59] P. Rattanatamrong and J. Fortes. Fuzzy scheduling of real-time ensemble systems. In *Proceedings of the 2014 International Conference on High Performance Computing Simulation*, HPCS 2014, pages 146–153, Jul 2014.

[60] E. Rigas, S. Ramchurn, and N. Bassiliades. Managing electric vehicles in the smart grid using artificial intelligence: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):1619–1635, 2015.

[61] S. Saewong and R. Rajkumar. Practical voltage-scaling for fixed-priority RT-systems. In *Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS 2003, pages 106–114, 2003.

[62] A. Sharifi, H. Zhao, and M. Kandemir. Feedback control for providing QoS in NoC based multicores. In *Proceedings of the 2010 Design, Automation Test in Europe Conference Exhibition*, DATE 2010, pages 1384–1389, Mar 2010.

[63] A. Sobeih, M. Viswanathan, D. Marinov, and J. Hou. J-Sim: An integrated environment for simulation and model checking of network protocols. In *Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium*, IPDPS 2007, pages 1–6, 2007.

[64] J. A. Stankovic, T. He, T. Abdelzaher, M. Marley, G. Tao, S. Son, and C. Lu. Feedback control scheduling in distributed real-time systems. In *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, RTSS 2001, pages 59–70, 2001.

[65] L. Tedesco, F. Clermidy, and F. Moraes. A monitoring and adaptive routing mechanism for QoS traffic on mesh NoC architectures. In *Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS 2009, pages 109–118, 2009.

[66] G. Velkoski, M. Simjanoska, S. Ristov, and M. Gusev. CPU utilization in a multitenant cloud. In *Proceedings of the 2013 IEEE EUROCON*, pages 242–249, Jul 2013.

[67] X. Wang, Y. Chen, C. Lu, and X. Koutsoukos. FC-ORB: A robust distributed real-time embedded middleware with end-to-end utilization control. *Journal of Systems and Software*, 80(7):938–950, 2007.

[68] X. Wang, X. Fu, X. Liu, and Z. Gu. Power-aware CPU utilization control for distributed real-time systems. In *Proceedings of the 15th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS 2009, pages 233–242, Apr 2009.

[69] X. Wang, X. Fu, X. Liu, and Z. Gu. PAUC: Power-aware utilization control in distributed real-time systems. *IEEE Transactions on Industrial Informatics*, 6(3):302–315, Aug 2010.

[70] X. Wang, Y. Hong, J. Huang, and Z.-P. Jiang. A distributed control approach to a robust output regulation problem for multi-agent linear systems. *IEEE Transactions on Automatic Control*, 55(12):2891–2895, Dec 2010.

[71] X. Wang, D. Jia, C. Lu, and X. Koutsoukos. DEUCON: Decentralized end-to-end utilization control for distributed real-time systems. *IEEE Transactions on Parallel and Distributed Systems*, 18(7):996–1009, 2007.

[72] Z. Wang, D. Ma, and Y. Zhao. A web services container supporting QoS hierarchical control with multiple measurements for utilization. In *Proceedings of the 11th International Conference on Dependable, Autonomic and Secure Computing*, DASC 2013, pages 39–46, Dec 2013.

[73] R. Xu, R. Melhem, and D. Mosse. Energy-aware scheduling for streaming applications on chip multiprocessors. In *Proceedings of the 28th IEEE International Real-Time Systems Symposium*, RTSS 2007, pages 25–38, 2007.

[74] J. Yao, X. Liu, Z. Gu, X. Wang, and J. Li. Online adaptive utilization control for real-time embedded multiprocessor systems. *Journal of Systems Architecture*, 56(9):463–473, 2010.

[75] J. Yao, X. Liu, W. He, and A. Rahman. Dynamic control of electricity cost with power demand smoothing and peak shaving for distributed Internet data centers. In *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems*, ICDCS 2012, pages 416–424, 2012.

[76] J. Yao, X. Liu, M. Yuan, and Z. Gu. Online adaptive utilization control for real-time embedded multiprocessor systems. In *Proceedings of the 6th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS 2008, pages 85–90, 2008.

[77] P. Zhou, A. Paul, C. Kim, and S. Sapatnekar. Distributed on-chip switched-capacitor DC-DC converters supporting DVFS in multicore systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(9):1954–1967, 2014.

[78] D. Zhu and H. Aydin. Reliability-aware energy management for periodic real-time tasks. *IEEE Transactions on Computers*, 58(10):1382–1397, 2009.

[79] Y. Zhu and F. Mueller. Feedback EDF scheduling exploiting dynamic voltage scaling. In *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium*, RTAS 2004, pages 84–93, 2004.

[80] Y. Zhu and K. Tomsovic. Real-time control of distributed energy resources. In *Proceedings of the 2010 IEEE Power and Energy Society General Meeting*, pages 1–6, 2010.