Resource-aware Automotive Control Systems Design: A Cyber-Physical Systems Approach

Wanli Chang
Singapore Institute of Technology, TUM CREATE Singapore
wanli.chang@singaporetech.edu.sg

Samarjit Chakraborty
TU Munich
samarjit@tum.de

now
the essence of knowledge
Boston — Delft

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

W. Chang and S. Chakraborty. *Resource-aware Automotive Control Systems Design: A Cyber-Physical Systems Approach.* Foundations and Trends® in Electronic Design Automation, vol. 10, no. 4, pp. 249–369, 2016.

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

ISBN: 978-1-68083-239-6

Full text available at: http://dx.doi.org/10.1561/1000000045

Foundations and Trends® in Electronic Design Automation

Volume 10, Issue 4, 2016

Editorial Board

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, 2016, Volume 10, 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. 10, No. 4 (2016) 249–369

DOI: 10.1561/1000000045

Resource-aware Automotive Control Systems Design: A Cyber-Physical Systems Approach

Wanli Chang
Singapore Institute of Technology, TUM CREATE Singapore
wanli.chang@singaporetech.edu.sg

Samarjit Chakraborty
TU Munich
samarjit@tum.de

Contents

Abstract

As the automotive industry is entering the smart era through advances in sensing, computation, storage, communication, and actuation technologies, a larger number of more complex control applications with better performances are expected to be on board. This requires an implementation platform with abundant resources, which is undesired in the cost-sensitive automotive domain. The implementation platform, often embedded in an Electronic Control Unit (ECU) and shared by multiple applications to save cost, is mainly comprised of a processor for computation, memory for storing instructions and data, and bus for internal and external communication. Conventionally, automotive control systems are designed using model-based approaches, where the details of the implementation platform are ignored. Techniques that integrate the characteristics of implementation resources into control algorithms design are largely missing. Such a separate design paradigm is too conservative in resources dimensioning and utilization for modern vehicles. This article presents recently developed approaches in automotive control systems design that take implementation resources into consideration, aiming to improve the control performances for a given amount of resources, or equivalently, realize the required control performances with fewer resources. While communication resources have been extensively explored in the literature of networked embedded control systems, we will focus on memory and computation resources, which have started to receive attention from the academic community and industry just recently. As Electric Vehicles (EVs) have become a new trend in the automotive industry, energy resources of EVs, i.e., the batteries, are also investigated. A number of real-world applications validate the resource-aware automotive systems design techniques presented in this article.

W. Chang and S. Chakraborty. *Resource-aware Automotive Control Systems Design: A Cyber-Physical Systems Approach*. Foundations and Trends® in Electronic Design Automation, vol. 10, no. 4, pp. 249–369, 2016.
DOI: 10.1561/1000000045.

1

Introduction

Performance and reliability of automobiles are influenced by feedback control applications implemented on board. At the inception of the first horseless carriage, some form of control was already applied to motor vehicles. Engine idle speed control, which can be found in every modern vehicle powered by an internal combustion engine (ICE), traces back to the Watt's governor in 1769. This device marking the origin of both feedback control and the industrial revolution can be viewed as a mechanical idle speed feedback controller for a steam engine Brennan et al. [2007].

Over the last century, control has been applied to almost every aspect of vehicle operation, from engine to drivetrain, from steering to braking. Applications including anti-lock braking system (ABS), traction control, electronic stability control (ESC), and active safety systems have decreased the number and severity of accidents. With advances in sensing, computation, storage, communication, and actuation technologies, more complex automotive control applications targeting better performances have emerged. In the combustion engine control, homogeneous charge compression ignition has been developed to reduce NO_x emission Chiang et al. [2007], Ortner and del Re [2007]. In the powertrain control, a design for Six Sigma (DFSS) analysis approach is used to determine automatic transmission gear content, aiming at fuel consumption minimization for various powertrain systems Robinette [2014]. A unified chassis control strategy integrating active front steering (AFS) and ESC is proposed in Cho et al. [2012] to improve agility, maneuverability, and vehicle lateral stability.

Along the direction of autonomous driving, which can be classified into five levels[1], new control applications have debuted in the modern premium cars. Nowadays, most vehicles with autonomous features in the market fall into Level 1 or 2. For instance, adaptive cruise control (ACC) enables the driver to cede limited authority over a primary control. Lane keeping automatically assumes limited authority over a primary control. Dynamic brake support provides added control to aid the driver in emergencies. Other emerging control applications include automated parking Müller et al. [2007], path tracking Snider [2009], obstacle avoidance Villagra et al. [2007], and vehicle control at friction limits Kritayakirana [2012].

One key future research direction for autonomous vehicles is to develop safe software. A guarantee on the vehicle behavior is highly desirable. One way is to ensure that all the executables will generate correct results within a certain period of time. The other way is, when some programs fail, the controller is still able to maintain safe operations of the vehicle. This requires researchers and engineers to consider the interaction between control algorithms and the embedded implementation platform.

1.1 Motivation

In the automotive Electrical/Electronic (E/E) architecture as shown in Figure 1.1, control applications are implemented on a platform embedded in an electronic control unit (ECU). ECUs are connected to physical plants under control (e.g., the engine, the motor, brakes, and

[1] Level 0: no automation; Level 1: function-specific automation; Level 2: combined-function automation; Level 3: limited self-driving automation; Level 4: full self-driving automation National Highway Traffic Safety Administration [2013]

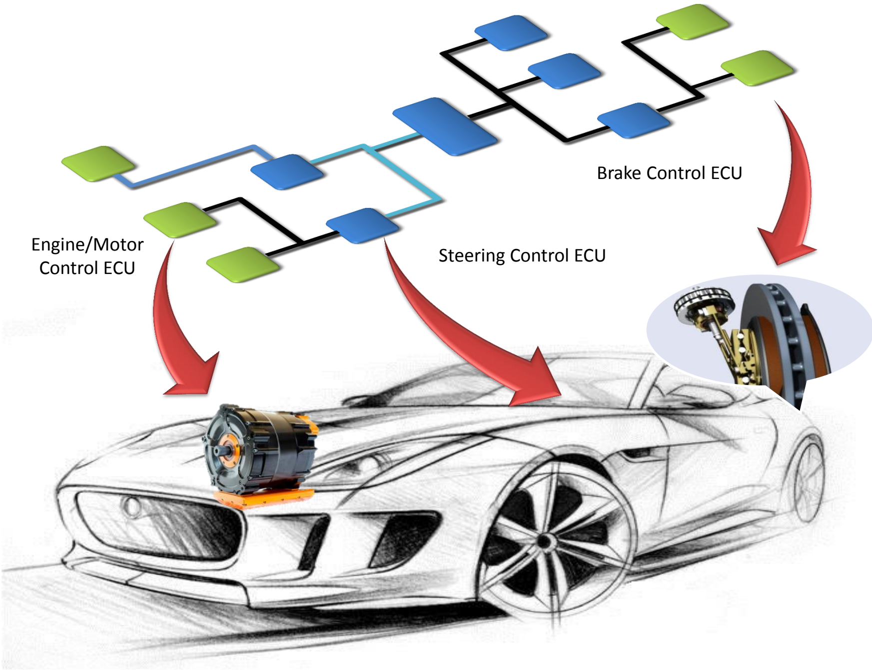

Figure 1.1: Backbone of automotive E/E architecture

the steering wheel) via the communication bus. A feedback control loop has three operations:

- **Measurement:** Sensors measure the states of the physical plants. This is also called sampling.
- **Computation:** Taking the data from sensors, control programs are executed and compute the control input.
- **Actuation:** The control input is sent to actuators, aiming to achieve certain desired behavior of the plants.

A typical embedded implementation platform for automotive control applications is shown in Figure 1.2. There are often programs of multiple control applications executed on one processor, which necessitates an operating system (OS) for coordination. The flash memory stores all

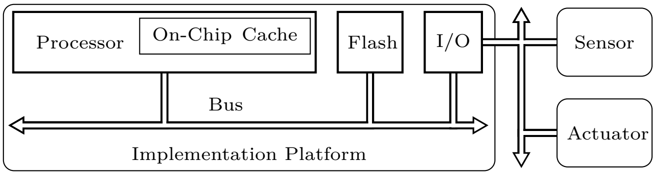

Figure 1.2: A typical embedded implementation platform for automotive control applications. The processor executes control programs. Instructions and data are stored in the flash memory. On-chip cache accelerates the memory access. Programmable I/O peripherals are used for communication with sensors and actuators.

instructions and data. The on-chip cache accelerates the memory access. Programmable input/output (I/O) peripherals are used for communication with sensors and actuators over the bus.

The implementation platform considerably impacts the control performances via, e.g., sampling periods and sensor-to-actuator delays. The sampling period is defined as the time duration between two consecutive measurements (or samplings) of the plant states under control. The sensor-to-actuator delay is defined as the time duration between the measurement and the actuation of one feedback control loop. If the processor or the memory access or the communication is not fast enough, the execution time of the control program and the transmission of messages might be too long to meet the desired sampling periods and sensor-to-actuator delays. Therefore, a larger number of more complex control applications calls for an implementation platform with abundant resources, which contradicts the cost-sensitive nature of the automotive industry.

The algorithms development for control applications from a control-theoretical perspective is well-established. The controller design methods can be drawn from a large pool of research and practical experience that have been accumulated in the control community. However, little attention has been paid to the embedded implementation platform. Control theorists and embedded system engineers make model-based assumptions of the other side. Since most control applications are safety-critical, such assumptions in this separate design paradigm are inevitably conservative to guarantee the required control performances. As a result, the resources on the embedded implementation platform, such as communication, computation and memory, are inefficiently utilized. This article presents recently developed techniques in automotive control systems design that take implementation resources into consideration, aiming to improve the control performances for a given amount of resources, or equivalently, realize the required control performances with fewer resources.

Motivated by the increasing worldwide efforts to reduce greenhouse gas (GHG) emissions, automotive manufacturers have been struggling in upgrading their ICEs. It is challenging to reduce emissions while keeping the engine performance. An alternative solution is an electric vehicle (EV). Another major advantage of an EV is its independence of fossil fuels[2]. Besides, the torque and noise performances of an electric motor are generally better than an ICE of the similar size at low speeds. Most major car manufacturers have presented their mass-produced EVs, including Nissan Leaf, BMWi3, Volkswagen e-Golf, Chevy Volt, and Tesla Model S.

One major issue that impedes the market acceptance of EVs is the range anxiety. The energy resource is the major factor determining the driving range of an EV. Given a fixed battery pack, it is desired to maximize the battery usage (i.e., maximize the effective battery capacity and minimize the energy consumption of a control task instance), which is directly related to the driving range. Different control strategies result in different discharging current profiles and the battery usage depends on the discharging current profile. This article discusses the control systems design in an EV taking the energy resource into consideration. The influence of processor aging in the implementation platform on both the control performance and the battery usage is discussed, and countermeasures are presented.

The resources on the embedded implementation platform for control applications (e.g., memory and computation resources) are a little different from the energy resources (including the influence from processor aging). The common issue is that these resources have not be consid-

[2]It is noted that both GHG emissions reduction and fossil fuels independence also depend on the way of electricity generation.

ered in designing control algorithms, despite that they are important in the context of cost-sensitive domains like automotive systems. This article highlights the design techniques to account for these resources in designing automotive control systems.

1.2 Resources

In this section, communication, memory, computation, and energy resources are described, along with how controller design strategies might take these resources into account. There is a considerable amount of work on communication-aware control systems design, particularly in the networked control systems literature. In contrast, other forms of resources – like computation, memory or energy – have not been sufficiently studied so far. Hence, in this article we will focus on controller design strategies for these types of resources and not elaborate on communication-aware controller design. Instead, the this topic is briefly surveyed below, and in particular we describe a controller design strategy for hybrid time- and event-triggered automotive communication protocols such as FlexRay.

1.2.1 Communication Resources

The number of bits that can be transmitted per unit of time over a communication network is limited by its bandwidth. Precise characterization of the communication resources for automotive control systems is protocol-specific. The communication protocols are broadly classified into two groups — event-triggered (ET) and time-triggered (TT) networks. For instance, Controller Area Network (CAN) is ET and has been widely used since its first official release in 1986 Bosch [1991]. FlexRay FlexRay Consortium [2005], which was designed about a decade ago to be faster and more reliable than CAN, can be found in most premium cars. Media access control in FlexRay is based on communication cycles of equal and predefined length in time. Each communication cycle is divided into a TT static and an ET dynamic segment as shown in Figure 1.3. Messages can be sent with FlexRay over either the TT or ET segment using a bandwidth of 10 Mbit/s.

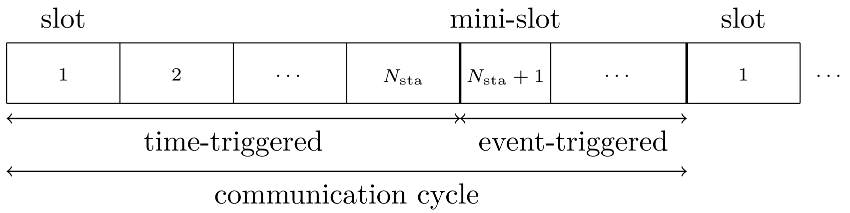

Figure 1.3: FlexRay bus with both time-triggered static and event-triggered dynamic segments

The TT static segment follows a Timing Division Multiple Access (TDMA) policy for media access control, where the entire segment is divided into multiple slots with the same predefined length in time. In every segment, slots are statically indexed starting from 1 to N_{sta}, which is the total number of TT slots in the static segment. Each application involved in the TT communication is assigned a dedicated index number and only uses the TT slot of this index to transmit messages. This allows a predictable temporal behavior, since in every communication cycle, an application is able to access the TT segment once, and the interval between two consecutive allowed transmissions is fixed. Deterministic timing and short delay can be transformed to good control performance with appropriate controller design. If no messages from an application need to be sent on its given slot, then the network is idle for this period of time, resulting in an inefficient utilization of bandwidth.

In the ET dynamic segment, media access control is priority-based and the entire segment is divided into mini-slots. Every application involved in the ET communication is associated with an index and in effect, a priority. In a segment, each mini-slot is dynamically assigned an index. The starting index is $N_{\text{sta}} + 1$. The application matching the mini-slot index is allowed to transmit a message. A message can be transmitted over multiple mini-slots and the mini-slots transmitting the same message have the same index. After the transmission ends, the mini-slot index is incremented. If the message is not ready when its mini-slot starts, the mini-slot goes idle and the index is incremented. Since a mini-slot is typically much shorter than a TT slot, the ET

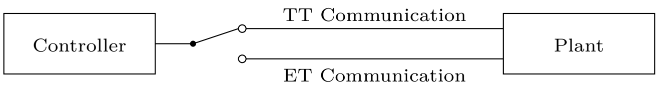

Figure 1.4: The switching between ET and TT communication when transmitting a control message from the controller to the plant

segment offers more efficient utilization of bandwidth compared to the TT segment. The ET segment generally does not provide temporal guarantee due to its priority-based nature of arbitration. The timing of a message over ET communication depends on the presence of messages with higher priorities. This degrades the control performance.

A number of recent efforts have been made to address the communication-aware embedded control systems design. An aperiodic strategy for dynamic allocation of bandwidth according to the current state of the plants and available resources is proposed in Anta and Tabuada [2009]. Control loops closed over CAN are discussed and illustrated on a train car. In Samii et al. [2009], communication delay and jitter resulting from complex timing behavior are considered. A method integrating controller design and message scheduling is developed to optimize the overall control performance. A predictive compensator co-located with the actuator is proposed in Henriksson et al. [2008] to deal with communication outages. When a new control command is not received, a replacement one based on the history of past control commands is suggested.

There are several challenges in the communication-control co-design. First, the design space can be too large to be tractable. There are many parameters to determine in the design of the controller and the communication network. A combined design space can be difficult to handle. This is aggravated by the increase of system size. Second, the trade-off between the control performance and the communication resource utilization, which enables more design freedom, has not been explored. Some first efforts have been made in Roy et al. [2016] to address these two issues.

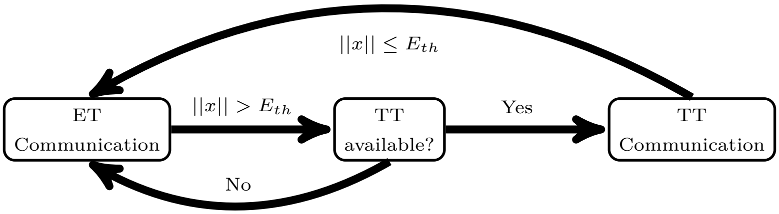

Figure 1.5: The hybrid communication protocol

Another challenge is to minimize the use of TT communication for each application, while still achieving satisfactory control performances. This will result in more efficient utilization of the communication resources. One method to address this challenge is to implement a hybrid communication scheme. As illustrated in Figure 1.4, when transmitting a control message from a controller to a plant, the hybrid communication scheme allows switching between TT and ET communication. This enables the trade-off between required resources and control performance. In order to achieve satisfactory control performance that is close to pure TT communication while using fewer TT slots, the protocol illustrated in Figure 1.5 is implemented. If a plant is in the steady-state, i.e., the norm of the state vector $||x||$ is less than or equal to the threshold E_{th}, ET communication is used. When a disturbance forces the plant out of the steady-state, it is checked whether the TT slot is available. If the TT slot is available, then the control message switches to TT communication. The TT slot is used until the plant is brought back to the steady-state. Then the control message switches back to ET communication. If the TT slot is occupied by another application and thus unavailable, the ET communication continues to be used and the TT availability keeps getting checked, until that the plant is back to the steady-state. Such a protocol allows multiple applications to share one TT slot, and thus reduces the usage of TT communication. A schedulability analysis is needed to ensure that desired control performances are satisfied.

Different ways to conduct such schedulability analysis have been reported:

- Generally, the schedulability analysis for real-time systems mainly determines the response time of a real-time task Bini and Buttazzo [2004], Jayachandran and Abdelzaher [2009], Yao et al. [2015].

- There are specific schedulability analysis techniques for TT Hu et al. [2015], ET Broster et al. [2002], and the hybrid communication Phan et al. [2009]. The focus is to compute the upper bounds on communication delays.

- There are schedulability analysis techniques particularly considering control systems and aiming to satisfy control performance requirements Tabuada [2007], Majumdar et al. [2011], Han et al. [2013].

1.2.2 Memory Resources

In the two-level memory architecture shown in Figure 1.2 (such as the XC23xxB Series microcontroller Infineon [2009] from Infineon that is popular in automotive systems), the flash memory has a large size and can thus store all the application programs and data, but experiences high read/write latencies (hundreds of processor cycles). The cache is faster with low read/write latencies (several processor cycles), but usually limited in size due to its high cost. It is assumed that the access times of cache and flash memory are t_c and t_m, respectively, where $t_c \ll t_m$. In this article, the focus is on instruction memory, since control applications are typically not data-intensive.

When a processor executes an instruction, it checks the cache first. If this instruction is located in the cache, it is a cache hit and the access time is t_c. If this instruction is not in the cache, the memory block containing it is fetched from the flash memory and then written into cache. This is a cache miss and the access time is t_m. Afterwards, when the same instruction is called again by the processor, the access time is t_c if it is still in the cache without being replaced. This is a cache reuse.

A program usually has different execution paths resulting in different execution times. The worst-case execution time (WCET) is defined to be the maximum length of time a program takes to be executed. The WCET constrains the sampling period of a control application. There are two general methods to reduce the WCET of a program — increasing the cache size and/or cache reuse. In resource-aware automotive control systems design, it is desirable to minimize the cache size while satisfying the performance requirement, or equivalently, improve the performance for the given memory resources. Therefore, the cache reuse should be maximized.

Given a collection of control applications (e.g., $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$), it is conventional to run the control loops of them in a round-robin fashion ($\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\cdots$). Since the codes for different control applications are different, the on-chip cache is frequently refreshed in this process. This results in poor cache reuse and long WCET. In order to address this issue, a new sampling order for the control applications has been proposed Chang et al. [2017], using which cache reuse is improved and the WCET of each application is reduced. In particular, a non-uniform sampling scheme has been studied, where the control loop of each application is consecutively run multiple times — in order to increase cache reuse, before moving on to the next application.

An example memory-aware sampling order ($\mathcal{C}_1(1)$, $\mathcal{C}_1(2)$, $\mathcal{C}_1(3)$, $\mathcal{C}_2(1)$, $\mathcal{C}_2(2)$, $\mathcal{C}_2(3)$, $\mathcal{C}_3(1)$, $\mathcal{C}_3(2)$, $\mathcal{C}_3(3)$, $\cdots$) is illustrated in Figure 1.6, where $\mathcal{C}_i(j)$ denotes the jth execution of the control application $\mathcal{C}_i$. Before the first execution $\mathcal{C}_i(1)$, the cache is either empty (i.e., cold cache) or filled with instructions from other applications, that are not used by $\mathcal{C}_i$ (equivalent to cold cache). The WCET of $\mathcal{C}_i(1)$ can be computed by a number of existing standard techniques Wilhelm and et al. [2008], Andalam et al. [2013], Wilhelm et al. [2009]. Before the second execution $\mathcal{C}_i(2)$, the instructions in the cache are from the same application $\mathcal{C}_i$ and thus can be reused. This results in more cache hits and hence shorter WCET. Depending on which execution path the program takes, the amount of WCET reduction varies. Therefore, a technique is required to compute the guaranteed WCET reduction of $\mathcal{C}_i(2)$ and $\mathcal{C}_i(3)$ relative to $\mathcal{C}_i(1)$, independent of the path taken.

Control parameters of the applications, such as sampling periods and sensor-to-actuator delays, can be derived from the WCET results.

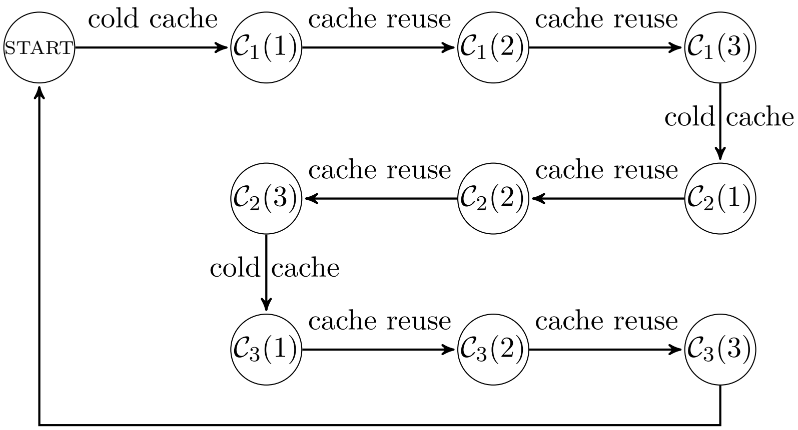

Figure 1.6: An example memory-aware sampling order with three control applications. Each application is consecutively executed three times. After the first execution $\mathcal{C}_i(1)$, some instructions in the cache can be reused and thus the WCETs of the following two executions are shortened.

A controller must be tailored for the memory-aware non-uniform sampling orders, so that the control performance can be improved. In summary, two main techniques are required — (i) cache analysis to compute the guaranteed WCET reduction between two consecutive executions of one program; (ii) controller design for the non-uniform sampling with sensor-to-actuator delays shorter than or equal to the sampling periods. Details of the memory-aware automotive control systems design will be discussed in Chapter 3.

1.2.3 Computation Resources

For a given processor with a certain operating frequency, computation resources usually mean the available execution time. When multiple applications share one processor, in general, the performance of an application can be improved if it is allowed to access the processor for a longer period of time. Computation-aware automotive control systems design aims to reduce the execution time of a control application, while still satisfying its performance requirement. In this way, more applications can be mapped to the processor, thereby saving the cost. This is the recent trend of ECU consolidation in the automotive industry.

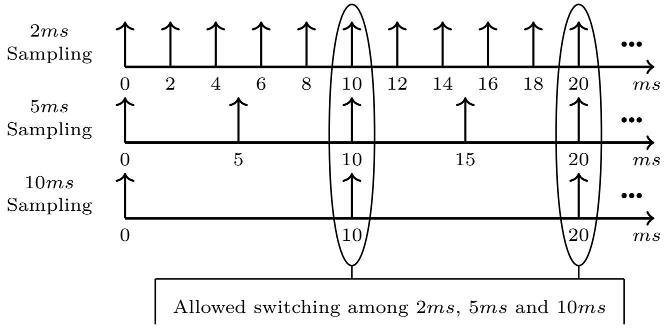

Figure 1.7: Allowed switching instants among multiple sampling periods

Generally, a shorter sampling period allows the controller to respond to its plant more frequently, and is thus potentially able to achieve better control performance with an appropriately designed controller. The obvious downside is a higher processor load, which is defined to be the WCET of an application divided by its sampling period. This prevents more functions and applications from being integrated onto the processor. Therefore, the controller should use the largest possible sampling period that is able to fulfill the control performance requirement and satisfy the system constraints.

Due to the safety-critical nature of the automotive domain, TT OS usually runs on the processor. For instance, OSEK/VDX (Open Systems and Their Corresponding Interfaces for Automotive Electronics/Vehicle Distributed Executive) OS Consortium [2005], Feiler [2003] is widely used in automobiles and considered in this article. OSEK/VDX OS only offers a limited set of predefined periods, which implies that the sampling periods of control applications have to be taken from this set. In most cases, the optimal sampling period is not directly realizable on the OS. The conventional way to handle it is to use the largest sampling period offered by the OS that is smaller than the optimal one. This is a straightforward method, yet leads to a waste of computational resources. It is desirable to minimize the processor load of an application, while still satisfying the performance requirement.

Towards this goal, a multirate controller that switches between available sampling periods offered by OSEK/VDX OS has been proposed Chang et al. [2016]. A typical example with sampling periods of $2ms$, $5ms$ and $10ms$ on OSEK/VDX OS is illustrated in Figure 1.7. Switching between two sampling periods can only occur at the common multiplier of them. For instance, switching between $2ms$ and $5ms$ is possible at the time instant of $10ms$, $20ms$, and so on. Therefore, possible sequences of sampling periods are $\{2ms, 2ms, 2ms, 2ms, 2ms, 5ms, 5ms, repeat\}$, $\{5ms, 5ms, 10ms, repeat\}$, and so on. The main challenge lies in the performance-oriented multirate controller design under the non-uniform sampling scheme with negligible sensor-to-actuator delays, aiming to reduce the processor load. Details of the computation-aware automotive control systems design will be discussed in Chapter 4.

1.2.4 Energy Resources

For all practical purposes, a longer driving range is desired in EVs to increase their usability. A battery pack with large capacity is needed to offer a long driving range. However, with larger capacity, the battery weight also increases leading to higher energy consumption. Moreover, the capacity is restricted by the space that can be allocated to the battery pack in EVs. One potential solution to the above problem is to design the controller in such a way that the energy consumption of a control task instance can be minimized.

All off-the-shelf battery packs are labeled with a nominal capacity. However, due to the rate capacity effect, the effective capacity or full-charge capacity (FCC) of a battery pack, which is defined to be the amount of electric charges that can be delivered from the battery after it is fully charged, actually varies with different discharging current profiles Doerffel and Sharkh [2006], Kim and Qiao [2011]. Generally speaking, larger discharging current tends to reduce the effective capacity. For most common lithium-ion batteries in the market, the capacity could potentially get significantly compromised if the rate capacity effect is not properly considered in the control systems design.

In this article, an optimization framework considering the control performance as one design objective and battery usage as the other is presented Chang et al. [2014]. The trade-off between these two design objectives is explored by generating a Pareto front. The battery usage is quantified by the number of times the control system can reach a steady state after a disturbance occurs powered up by a fully charged battery pack. In order to maximize the battery usage, the energy consumption of a control task instance, i.e., the disturbance rejection, should be small and the battery effective capacity should be increased by generating a battery-friendly discharging current profile.

In this context, the other important design aspect is processor aging. As a processor ages, the switching time of its transistors increases, resulting in longer path delays. On-chip monitors could be used to measure the delay of the critical path. It always has to be guaranteed that the signal transmission can be complete along any path within one clock cycle Lorenz et al. [2010]. Therefore, the processor operating frequency is reduced based on the new critical path delay.

As discussed above, a shorter sampling period can potentially provide a better control performance. Therefore, with a smaller processor operating frequency, the sampling period increases and the control performance gets deteriorated, which is dangerous and thus highly unwanted for safety-critical applications in EVs, such as electric motor control. To deal with the above situation, the same optimization framework can be slightly modified to re-optimize the controller with the longer sampling period, which results from processor aging, aiming to ensure that the control performance is kept with an inconsiderable compromise on battery usage. Details of the battery- and aging-aware automotive control systems design will be discussed in Chapter 5.

A literature review on resource-aware embedded control systems design is summarized in Table 1.1. Communication-aware embedded control systems design has been extensively explored, our contributions are mainly on memory, computation, and energy resources. While there have been a number of works discussing computation-aware embedded control systems design, we focus on the automotive OS. No works other than ours have investigated memory and energy resources in embedded control systems design. We do benefit from the literature on memory and battery modelling.

1.3 Organization

This article comprises six sections. Section 1 is the introduction. The background, including the necessary mathematical basics of control theory and optimization techniques, and the real-life automotive control applications used in the experiments, is presented in Chapter 2. Feedback control applications are first described, following which are the linear state-feedback control law and the non-linear MPC. The relationship between the control performance and the sampling period is shown based on an electronic wedge brake (EWB) developed by Siemens Fox et al. [2007]. The presented optimization techniques include particle swarm optimization (PSO), gradient-based sequential quadratic programming (SQP), and genetic algorithms for non-convex single- and multi-objective problems.

Chapter 3 discusses memory-aware automotive control systems design. The memory analysis technique that computes the guaranteed WCET reduction due to consecutive executions of one control program is first given. A motivational example is used for the illustration purpose. The control parameters, such as sampling periods and sensor-to-actuator delays, are then derived based on the WCET results. The controller design techniques for both the conventional memory-oblivious uniform sampling scheme and the memory-aware non-uniform sampling scheme are elaborated. Experimental results are reported at the end of the chapter.

Chapter 4 discusses computation-aware automotive control systems design. The OS used in automobiles is described and the restriction on the choice of sampling periods is addressed. The multirate controller design technique is presented to reduce the processor load while satisfying the control performance requirement and system constraints.

Chapter 5 discusses battery- and aging-aware automotive control systems design. The design objective of battery usage is introduced with battery characteristics. The processor aging and its influence on

Table 1.1: A literature summary on resource-aware embedded control systems design

Resource Type	Paper
Communication	On the Benefits of Relaxing the Periodicity Assumption for Networked Control Systems over CAN Anta and Tabuada [2009]
	Integrated Scheduling and Synthesis of Control Applications on Distributed Embedded Systems Samii et al. [2009]
	Predictive Compensation for Communication Outages in Networked Control Systems Henriksson et al. [2008]
	Multi-Objective Co-Optimization of FlexRay-based Distributed Control Systems Roy et al. [2016]
Memory	Memory-Aware Feedback Scheduling of Control Tasks Robertz et al. [2006]
	Dynamic Round-Robin Task Scheduling to Reduce Cache Misses for Embedded Systems Batcher and Walker [2008]
	Memory Hierarchies, Pipelines, and Buses for Future Architectures in Time-Critical Embedded Systems Wilhelm et al. [2009]
	Accurate Estimation of Cache-Related Preemption Delay Negi et al. [2003]
	Cache-Aware Timing Analysis of Streaming Applications Chakraborty et al. [2009]
	A Synergetic Approach to Accurate Analysis of Cache-Related Preemption Delay Kleinsorge et al. [2011]
Computation	Resource Management for Control Tasks based on the Transient Dynamics of Closed-Loop Systems Castane et al. [2006]
	Design and Stability Analysis for Anytime Control via Stochastic Scheduling Greco et al. [2011]
	Dynamic Scheduling and Control-Quality Optimization of Self-Triggered Control Applications Samii et al. [2010]
Energy	Control Oriented 1D Electrochemical Model of Lithium Ion Battery Smith et al. [2007]
	Simple PSpice Models Let You Simulate Common Battery Types Hageman [1993]
	An Analytical High-Level Battery Model for Use in Energy Management of Portable Electronic Systems Rakhmatov and Vrudhula [2001]

the control system is analyzed. Then the optimization framework and flow are shown. Experimental results can be found at the end of the chapter. The conclusion of this article is given in Chapter 6 and possible future work is discussed.

References

J. Ackermann and V. Utkin. Sliding Mode Control Design based on Ackermann's Formula. *IEEE Transactions on Automatic Control*, 43(2):234–237, 1998.

S. Andalam, A. Girault, R. Sinha, P. Roop, and J. Reineke. Precise Timing Analysis for Direct-Mapped Caches. In *Proceedings of the 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, May 2013.

A. Anta and P. Tabuada. On the Benefits of Relaxing the Periodicity Assumption for Networked Control Systems over CAN. In *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, pages 3–12, Dec. 2009.

K. Astrom and R. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2009.

K. Batcher and R. Walker. Dynamic Round-Robin Task Scheduling to Reduce Cache Misses for Embedded Systems. In *Proceedings of the 2008 Design, Automation & Test in Europe Conference & Exhibition (DATE 2008)*, pages 260–263, Mar. 2008.

A. Bhave and B. Krogh. Performance Bounds on State-Feedback Controller with Network Delay. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, Dec. 2008.

E. Bini and G. Buttazzo. Schedulability analysis of periodic fixed priority systems. *IEEE Transactions on Computers*, 53(11):1462–1473, 2004.

E. Bini and G. Buttazzo. The Optimal Sampling Pattern for Linear Control Systems. *IEEE Transactions on Automatic Control*, 59(1):78–90, 2014.

E. Bini and A. Cervin. Delay-Aware Period Assignment in Control Systems. In *Proceedings of the 29th IEEE Real-Time Systems Symposium (RTSS)*, pages 291–300, Nov. 2008.

Bosch. CAN Specification Version 2.0, 1991.

K. Bowman, J. Tschanz, C. Wilkerson, S. Lu, T. Karnik, V. De, and S. Borkar. Circuit Techniques for Dynamic Variation Tolerance. In *Proceedings of the 46th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 4–7, Jul. 2009.

S. Brennan, J. Buckland, U. Christen, I. Haskara, and I. Kolmanovsky. Editorial: Special Issue on Control Applications in Automotive Engineering. *IEEE Transactions on Control Systems Technology*, 15(3):403–405, 2007.

I. Broster, A. Burns, and G. Rodriguez-Navas. Probabilistic analysis of CAN with faults. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2002.

R. Castane, P. Marti, M. Velasco, A. Cervin, and D. Henriksson. Resource Management for Control Tasks based on the Transient Dynamics of Closed-Loop Systems. In *Proceedings of the 18th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 171–182, Jul. 2006.

A. Cervin, M. Belasco, P. Marti, and A. Camacho. Optimal Online Sampling Period Assignment: Theory and Experiments. *IEEE Transactions on Control Systems Technology*, 19(4):902–910, 2011.

S. Chakraborty, T. Mitra, A. Roychoudhury, and L. Thiele. Cache-Aware Timing Analysis of Streaming Applications. *Real-Time Systems*, 41(1): 52–85, 2009.

W. Chang, A. Pröbstl, D. Goswami, M. Zamani, and S. Chakraborty. Battery- and Aging-Aware Embedded Control Systems for Electric Vehicles. In *Proceedings of the 35th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2014.

W. Chang, D. Gowami, S. Chakraborty, and A. Hamann. OS-Aware Automotive Control Systems Design. *ACM Transactions on Cyber-Physical Systems*, 2016. submitted.

W. Chang, D. Goswami, S. Chakraborty, J. Xue, L. Ju, and S. Andalam. Memory-Aware Embedded Control Systems Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017. forthcoming.

C. Chiang, A. Stefanopoulou, and M. Jankovic. Nonlinear Observer-based Control of Load Transitions in Homogeneous Charge Compression Ignition Engines. *IEEE Transactions on Control Systems Technology*, 15(3):438–448, 2007.

W. Cho, J. Choi, C. Kim, S. Choi, and K. Yi. Unified Chassis Control for the Improvement of Agility, Maneuverability, and Lateral Stability. *IEEE Transactions on Vehicular Technology*, 61(3):1008–1020, 2012.

OSEK/VDX Consortium. OSEK/VDX Operating System Specification Version 2.2.3, 2005.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

D. Doerffel and S. Sharkh. A Critical Review of Using the Peukert Equation for Determining the Remaining Capacity of Lead-Acid and Lithium-Ion Batteries. *Journal of Power Sources*, 165(2):395–400, 2006.

Harmonic Drive. Produktbeschreibung PMA.

P. Feiler. Real-Time Application Development with OSEK: A Review of the OSEK Standards. Technical report, Carnegie Mellon University, 2003.

FlexRay Consortium. The FlexRay Communications System Specifications Version 2.1, 2005.

J. Fox, R. Roberts, C. Baier-Welt, L. Ho, L. Lacraru, and B. Gombert. Modeling and Control of a Single Motor Electronic Wedge Brake. Technical report, SAE, 2007.

L. Greco, D. Fontanelli, and A. Bicchi. Design and Stability Analysis for Anytime Control via Stochastic Scheduling. *IEEE Transactions on Automatic Control*, 56(3):571–585, 2011.

S. Hageman. Simple PSpice Models Let You Simulate Common Battery Types. *Electronic Design News*, 38:117–129, 1993.

S. Han, K. Lam, J. Wang, and K. Ramamritham. On co-scheduling of update and control transactions in real-time sensing and control systems: Algorithms, analysis, and performance. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2325–2342, 2013.

E. Henriksson, H. Sandberg, and K. Johansson. Predictive Compensation for Communication Outages in Networked Control Systems. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, pages 2063–2068, Dec. 2008.

M. Hu, J. Luo, W. Yang, and B. Veeravalli. Scheduling periodic task graphs for safety-critical time-triggered avionic systems. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3):2294–2304, 2015.

Infineon. XC2300B-Series 16/32-bit μC for Automotive Safety, 2009.

P. Jayachandran and T. Abdelzaher. Flow-based mode changes: towards virtual uniprocessor models for efficient reduction-based schedulability analysis of distributed systems. In *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, pages 281–290, Dec. 2009.

A. Jordehi and J. Jasni. Parameter Selection in Particle Swarm Optimization: A Survey. *Journal of Experimental and Theoretical Artificial Intelligence*, 25(4):527–542, 2013.

T. Kim and W. Qiao. A Hybrid Battery Model Capable of Capturing Dynamic Circuit Characteristics and Nonlinear Capacity Effects. *IEEE Transactions on Energy Conversion*, 26(4):1172–1180, 2011.

J. Kleinsorge, H. Falk, and P. Marwedel. A Synergetic Approach to Accurate Analysis of Cache-Related Preemption Delay. In *Proceedings of the 2011 International Conference on Embedded Software (EMSOFT 2011)*, Oct. 2011.

K. Kritayakirana. *Autonomous Vehicle Control at the Limits of Handling*. PhD thesis, Stanford University, 2012.

A. Kumar, D. Sharma, and K. Deb. A Hybrid Multi-Objective Optimization Procedure Using PCX based NSGA-II and Sequential Quadratic Programming. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, pages 3011–3018, Sep. 2007.

E. Lavretsky and K. Wise. *Robust and Adaptive Control*. Springer London, 2013.

C. Lefurgy, A. Drake, M. Floyd, M. Alle, B. Brock, J. Tierno, and J. Carter. Active Management of Timing Guardband to Save Energy in POWER7. In *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1–11, Dec. 2011.

H. Lin and P. Antsaklis. Stability and Stabilizability of Switched Linear Systems: A Survey of Recent Results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.

O. Ljungkrantz, H. Lonn, H. Blom, C. Ekelin, and D. Karlsson. Modelling of Safety-Related Timing Constraints for Automotive Embedded Systems. In *Proceedings of the 2012 International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2012)*, pages 190–201, Sep. 2012.

D. Lorenz, M. Barke, and U. Schlichtmann. Aging Analysis at Gate and Macro Cell Level. In *Proceedings of the 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2010)*, Nov. 2010.

R. Majumdar, I. Saha, and M. Zamani. Performance-aware scheduler synthesis for control systems. In *Proceedings of the 2011 International Conference on Embedded Software (EMSOFT 2011)*, Oct. 2011.

A. Masrur, P. Kindt, M. Becker, S. Chakraborty, V. Kleeberger, M. Barke, and U. Schlichtmann. Schedulability Analysis for Processors with Aging-Aware Autonomic Frequency Scaling. In *Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2012)*, pages 11–20, Aug. 2012.

R. Mishra, D. Ioannou, S. Mitra, and R. Gauthier. Effect of Floating-Body and Stress Bias on NBTI and HCI on 65-nm SOI pMOSFETs. *IEEE Electron Device Letters*, 29(3):262–264, 2008.

Maxon Motor. Product Specifications, 2011.

B. Müller, J. Deutscher, and S. Grodde. Continuous Curvature Trajectory Design and Feedforward Control for Parking a Car. *IEEE Transactions on Control Systems Technology*, 15(3):541–553, 2007.

U.S. Department of Transportation National Highway Traffic Safety Administration. Preliminary Statement of Policy Concerning Automated Vehicles, 2013.

H. Negi, T. Mitra, and A. Roychoudhury. Accurate Estimation of Cache-Related Preemption Delay. In *Proceedings of the 1st IEEE/ACM/IFIP/ International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 201–206, Oct. 2003.

A. Nickabadi, M. Ebadzadeh, and R. Safabakhsh. A Novel Particle Swarm Optimization Algorithm with Adaptive Inertia Weight. *Applied Soft Computing*, 11(4):3658–3670, 2011.

P. Ortner and L. del Re. Predictive Control of a Diesel Engine Air Path. *IEEE Transactions on Control Systems Technology*, 15(3):449–456, 2007.

J. Park and J. Abraham. A Fast, Accurate and Simple Critical Path Monitor for Improving Energy-Delay Product in DVS Systems. In *Proceedings of the 2011 International Symposium on Low Power Electronics and Design (ISLPED 2011)*, pages 391–396, Aug. 2012.

M. Pedersen. Good Parameters for Particle Swarm Optimization. Technical report, Hvass Laboratories, 2010.

M. Pedram, N. Chang, Y. Kim, and Y. Wang. Hybrid Electrical Energy Storage Systems. In *Proceedings of the 2010 International Symposium on Low Power Electronics and Design (ISLPED 2010)*, pages 363–368, Aug. 2010.

L. Phan, S. Chakraborty, and I. Lee. Timing analysis of mixed time/event-triggered multi-mode systems. In *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, pages 271–280, Dec. 2009.

D. Rakhmatov and S. Vrudhula. An Analytical High-Level Battery Model for Use in Energy Management of Portable Electronic Systems. In *Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2001)*, pages 488–493, Nov. 2001.

S. Robertz, D. Henriksson, and A. Cervin. Memory-Aware Feedback Scheduling of Control Tasks. In *Proceedings of the 2006 IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2006)*, Sep. 2006.

D. Robinette. A DFSS Approach to Determine Automatic Transmission Gearing Content for Powertrain-Vehicle System Integration. *SAE International Journal of Passenger Cars — Mechanical Systems*, 7(3):1138–1154, 2014.

D. Roy, L. Zhang, W. Chang, D. Goswami, and S. Chakraborty. Multi-Objective Co-Optimization of FlexRay-based Distributed Control Systems. In *Proceedings of the 22nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr. 2016.

M. Ruderman, J. Krettek, F. Hoffmann, and T. Bertram. Optimal State Space Control of DC Motor. In *Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC)*, pages 5796–5801, Jul. 2008.

S. Samii, A. Cervin, P. Eles, and Z. Peng. Integrated Scheduling and Synthesis of Control Applications on Distributed Embedded Systems. In *Proceedings of the 2009 Design, Automation & Test in Europe Conference & Exhibition (DATE 2009)*, pages 57–62, Apr. 2009.

S. Samii, P. Eles, Z. Peng, P. Tabuada, and A. Cervin. Dynamic Scheduling and Control-Quality Optimization of Self-Triggered Control Applications. In *Proceedings of the 31th IEEE Real-Time Systems Symposium (RTSS)*, Nov. 2010.

D. Schroder and J. Babcock. Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing. *Journal of Applied Physics*, 94(1):1–18, 2003.

D. Sedighizadeh and E. Masehian. Particle Swarm Optimization Methods, Taxonomy and Applications. *International Journal of Computer Theory and Engineering*, 1(4):486–502, 2009.

D. Shin, Y. Kim, J. Seo, N. Chang, Y. Wang, and M. Pedram. Battery-Supercapacitor Hybrid System for High-Rate Pulse Load Applications. In *Proceedings of the 2011 Design, Automation & Test in Europe Conference & Exhibition (DATE 2011)*, Mar. 2011.

K. Smith, C. Rahn, and C. Wang. Control Oriented 1D Electrochemical Model of Lithium Ion Battery. *Energy Conversion and Management*, 48 (9):2565–2578, 2007.

J. Snider. Automatic Steering Methods for Autonomous Automobile Path Tracking. Technical report, Robotics Institute, Carnegie Mellon University, 2009.

Sony. Lithium Ion Rechargeable batteries Technical Handbook, 2016. URL `https://cdn.sparkfun.com/datasheets/Prototyping/Lithium%20Ion%20Battery%20MSDS.pdf`.

N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.

P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

J. Villagra, B. d'Andrea Novel, H. Mounier, and M. Pengov. Flatness-based Vehicle Steering Control Strategy with SDRE Feedback Gains Tuned via a Sensitivity Approach. *IEEE Transactions on Control Systems Technology*, 15(3):554–565, 2007.

R. Wilhelm and et al. The Worst-Case Execution-Time Problem — Overview of Methods and Survey of Tools. *ACM Transactions on Embedded Computing Systems*, 7(3):1–53, 2008.

R. Wilhelm, D. Grund, J. Reineke, M. Schlickling, M. Pister, and C. Ferdinand. Memory Hierarchies, Pipelines, and Buses for Future Architectures in Time-Critical Embedded Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(7):966–978, 2009.

C. Yang and D. Simon. A New Particle Swarm Optimization Technique. In *Proceedings of the 18th International Conference on Systems Engineering (ICSENG)*, pages 164–169, Aug. 2005.

G. Yao, H. Yun, Z. Wu, R. Pellizzoni, M. Caccamo, and L. Sha. Schedulability analysis for memory bandwidth regulated multicore real-time systems. *IEEE Transactions on Computers*, 65(2):601–614, 2015.

P. Yih. *Steer-by-Wire: Implications for Veicle Handling and Safety*. PhD thesis, Stanford University, 2005.

F. Zhang, K. Szwaykowska, W. Wolf, and V. Mooney. Task Scheduling for Control Oriented Requirements for Cyber-Physical Systems. In *Proceedings of the 29th IEEE Real-Time Systems Symposium (RTSS)*, pages 47–56, Nov. 2008.