Proceedings

3rd Workshop on Security and Dependability of Critical Embedded Real-Time Systems

In conjunction with IEEE/IFIP International Conference on
Dependable Systems and Networks, DSN 2018

June 25, 2018
Luxembourg

Contents

Keynote

Full Papers

Short Papers

Invited Papers

Message from the chairs

This is the third iteration of the workshop on security and dependability of critical embedded real-time systems (CERTS) and it is co-located with the IEEE/IFIP international conference on dependable systems and networks (DSN). This edition takes places in Luxembourg. The technical program includes multiple peer reviewed and invited papers and a keynote talk by Dr. Michael Paulitsch, Principal Engineer at Intel Labs, Germany. The aim of this workshop is to bring researchers and practitioners from a variety of domains, viz., real-time and embedded systems, security, dependability and cyber-physical systems to name just a few. The idea is to foster a community that looks at all of these topics and develops techniques, algorithms, policies and frameworks to improve the security and dependability of critical systems. We hope that the papers and the keynote will help foster such discussions and collaborations.

CERTS 2018 owes its success to a variety of people. We would like to thank the steering committee that consists of: Paulo Esteves-Verissimo, Marcus Volp, Antonio Casimiro and Rodolfo Pellizzoni. We would also like to thank the technical program committee members for taking the time to review and provide feedback for the papers. In addition, we also wish to thank the organizers of DSN 2018, in particular: Paulo Esteves-Verissimo (general chair of DSN); Gilles Muller and Marco Vieira (program co-chairs for DSN); Antonio Casimiro, Matti Hiltunen and Mohamed Kaaniche (workshop co-chairs for DSN) and Zbigniew Kalbarczyk and Karthik Pattabhiraman (DSN publication co-chairs). Finally, we would like to thank the authors and participants of the workshop without whom this event would not be successful.

We hope that you will enjoy the CERTS 2018 program and that it will foster many new research directions and collaborations.

Mikael Asplund
Linköping University

Sibin Mohan
University of Illinois at Urbana-Champaign

Workshop Organizers and Program Committee

Workshop Organizers:
Mikael Asplund, Linköpings Universitet
Sibin Mohan, University of Illinois at Urbana-Champaign

Technical Program Committee:
Antônio Augusto Fröhlich, Federal University of Santa Catarina
Rakesh Bobba, Oregon State University
Christian Esposito, University of Naples Federico II
Martin Gilje Jaatun, University of Stavanger
Karl Goeschka, Vienna University of Technology
Zbigniew Kalbarczyk, University of Illinois
Ravi Prakash, University of Dallas
Sasikumar Punnekat Maelardalen University
Hans Reiser, Universität Passau
Guillermo Rodriguez-Navas. Mälardalen University
José Rufino, Faculdade de Ciencias da Universidade de Lisboa
Elad Schiller, Chalmers University of Technology
Elena Troubitsyna, Åbo Akademi
Martin Törngren, KTH
Bryan Ward. MIT Lincoln Laboratory
Heechul Yun, Kansas University
Saman Zonouz, Rutgers University

CERTS 2018 Program

8:30 -- 9:00	Registration
9:00 -- 9:10	*Welcome and Opening Remarks* Sibin Mohan, University of Illinois at Urbana-Champaign Mikael Asplund, Linkoping University
9:10 -- 10:10	Keynote: "*Dependability and Security in Critical Transportation Industries*" Michael Paulitsch, Intel Labs Europe
10:10 -- 10:30	Q&A
10:30 -- 11:00	Coffee Break
11:00 -- 12:30	**Technical Session I: Full Papers**
11:00 -- 11:30	*A Systematic Way to Incorporate Security in Safety Analysis* Elena Lisova, Malardalen University Aida Causevic, Malardalen University Kaj Hanninen, Malardalen University Henrik Thane, Malardalen University Hans Hansson, Malardalen University
11:30 -- 12:00	*Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework* Henrique Mendes, University of Lisbon Ibéria Medeiros, University of Lisbon Nuno Neves, University of Lisbon
12:00 -- 12:30	*Design for Dependability through Error Propagation Space Exploration* Imre Kocsis, Budapest University of Technology and Sciences
12:30 -- 14:00	Lunch Break
14:00 -- 15:40	**Technical Session II : Short and Invited Papers**
14:00 -- 14:20	Short Paper: Real-Time Security through a TEE

	Roberto Duenez, University of Houston Albert Cheng, University of Houston
14:20 -- 14:40	Invited Paper: *Security-Aware Safety: Development and Assessment Perspectives* Elena Troubitsyna, Royal Institute of Technology (KTH)
14:40 -- 15:00	Invited Paper: *Cyber-Physical Control Systems: Vulnerabilities, Threats, and Mitigations* Luis Garcia, Rutgers University Saman Zonouz, Rutgers University
15:00 -- 15:20	Invited Paper: *Using Schedule-Abstraction Graphs for the Analysis of CAN Message Response Times* Mitra Nasri, Max Planck Institute for Software Systems Arpan Gujarati, Max Planck Institute for Software Systems Björn B. Brandenburg, Max Planck Institute for Software Systems
15:20 -- 15:40	Invited Paper: *Fault-Injection on a Haptic Rendering Algorithm in the Raven Surgical Robot* Keywhan Chung, University of Illinois at Urbana-Champaign Xiao Li, University of Illinois at Urbana-Champaign Zbigniew T. Kalbarczyk, University of Illinois at Urbana-Champaign Ravishankar K. Iyer, University of Illinois at Urbana-Champaign Thenkurussi Kesavadas, University of Illinois at Urbana-Champaign
15:40 -- 15:45	*Closing Remarks* Mikael Asplund, Linkoping University Sibin Mohan, University of Illinois at Urbana-Champaign

Keynote

Dependability and Security in Critical Transportation Industries

Michael Paulitsch

Intel Labs Europe

Ensuring safety and security is hard by itself. The future will demand integrated security and safety approaches due to cost and operational requirements and evolving system complexities based on additional functional needs. This talk will present examples of high-level system electronic architectures of aerospace and railway systems as examples of safety-critical transportation systems with focus on current and evolving safety and security perspectives. In this context it will discuss current and future research directions of transportation industries for open discussion at the workshop.

Michael Paulitsch is Dependability Systems Architect (Principal Engineer) at Intel as part of Intel Labs Europe since March 2018 and located in Munich, Germany,. His research interests are in dependable systems including security aspects affecting safety of all types of cyber-physical systems. From August 2014 to March 2018, He has been Head of Base Systems and Product Line Manager Vital Platform of Main Line Systems at Thales Austria GmbH (part of Thales Ground Transportation Systems) in Vienna, Austria. In these roles he has been responsible for the execution and strategy of as well as research on a safety-critical computing and communication platform with security requirements for railway and subway systems – called Thales TAS Platform. Before this, Michael has been Senior Expert of "Dependable Computing and Networks" as well as Scientific Director at Airbus Group Innovations in the "Electronic, Communication and Intelligent Systems" department based in Munich, Germany. There his work focused on dependable embedded and secure embedded computing and networks. From 2003 to 2008, he worked at Honeywell Aerospace in the U.S. on software and electronic platforms in the area of business, regional, air transport, and human space avionics and engine control electronics. Michael Paulitsch published 50+ scientific papers in his area of expertise, participates in internal scientific conference committees and holds 25+ patents. He holds a doctoral degree in technical sciences from the Vienna University of Technology, Vienna, Austria with emphasis on dependable embedded systems and a doctoral degree in economics and social sciences with emphasis on production management aspects.

Full Papers

Due to copyright reasons, the papers published in IEEE Xplore will appear here as abstracts only.

A Systematic Way to Incorporate Security in Safety Analysis

Elena Lisova, Aida Causevic, Kaj Hanninen, Henrik Thane, Hans Hansson

Mälardalen University

Safety and security engineering have for a long time been regarded as two separate disciplines, which has resulted in separate cultures, regulations, standards and practices. Today's systems are being built to connect to public or semi-public networks, are able to communicate with other systems, e.g., in the context of Internet-of-Things (IoT), involve multiple stakeholders, have dynamic system reconfigurations, and operate in increasingly unpredictable environments. In such complex systems, assuring safety and security in a continuous and joint effort is a major challenge, not the least due to the increasing number of attack surfaces arising from the increased connectivity.

In this paper we present an approach that aims to bridge the gap between safety and security engineering. The potential of the approach is illustrated on the example of E-gas system, discussing the cases when unintentional faults as well as malicious attacks are taken into consideration when assuring safety of the described system.

Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework

Henrique Mendes, Ibéria Medeiros, Nuno Neves

University of Lisbon

The electrical grid is a critical infrastructure for modern society. It has been evolving into a smart(er) grid, allowing infrastructure aware decisions based on data collected in real-time from smart meters and other devices. Smart meters and their uplinks have, however, limited physical security due to their location within customer premises. DLMS/COSEM is a standard protocol for remote interactions with smart meters, often being deployed above power-line communication links. The paper presents the ValiDLMS framework, the first open source solution for validation and security auditing of DLMS/COSEM implementations using this communication profile. The framework was developed as an extension to Wireshark and was used to analyse an industry partner's DLMS/COSEM implementation. The results show that ValiDLMS can effectively support the discovery of bugs and/or other non-conformance problems.

Design for Dependability through Error Propagation Space Exploration

Imre Kocsis

Budapest University of Technology and Sciences

With dependability-, and specifically safety-critical systems becoming more and more open, the importance of the ability to reason about error propagation in an exploratory style is becoming increasingly important. This paper proposes an initial theoretical framework for Error Propagation Space Exploration (EPSE) and outlines the activities it is able to support. The key difference from classic Error Propagation Analysis (EPA) is that all error propagation hypotheses are handled together as a hypothesis set relation. Key aspects of operationalizing the framework are also discussed.

Short Papers

Due to copyright reasons, the papers published in IEEE Xplore will appear here as abstracts only.

Real-Time Security through a TEE

Albert Cheng

University of Houston

The increasing complexity in embedded and cyber-physical systems has demanded a new dimension of security. Trusted Execution Environments (TEE) provide process isolation and dedicated hardware to prevent breach of sensitive information. This paper will focus on the use of TEE on TrustZone architectures for embedded systems security and an implementation with real-time constraints.

Invited Papers

Security-Aware Safety: Development and Assessement Perspectives

Elena Troubitsyna
School of Electrical Engineering and Computer Science
KTH – Royal Institute of Technology
Stockholm, Sweden
e-mail: elenatro@kth.se

***Abstract*—Increasing openness and reliance on networking in modern safety-critical control systems requires novel methodologies integrating security consideration in the system development and safety assessment. We discuss the steps to promote security-aware safety-driven development and propose a generic pattern from safety case that integrates both safety and security aspects.**

Keywords-safety-critical software; safety; secuirty; integrated analysis; safety assurance

I. Introduction

Modern control systems increasing rely on networking technologies in their functioning. While offering greater flexibility and possibility to provide richer functionality, the increased system openness also introduces security threats. To ensure safety, we have to integrate the mechanisms for coping with both accidental component failures and malicious failures – security attacks. However, traditionally safety and security have been considered as separate fields [1].

Both safety and security require the dedicated design efforts to detect a failure and perform error recovery. Currently there is a lack of the approaches that allow the designers to consider safety and security in an integrated way while designing software for high-assurance systems [1] [2] and assessing system safety.

Security analysis is typically data-centeric, i.e., it focuses on determining the impact of security attacks on the system data flow. In contrast, safety analysis is concerned with defining the impact of failures on function provisioning. We discuss an integrated approach to deriving safety and security requirements in a systematic way and integrating both safety and security considerations into safety argument. To derive both type of the requirements, we rely on the systems approach and apply safety analysis to the systems data flow. In its turn, the safety case reflects the integrated view and contains both safety and security goals and tactics.

II. Integrated Safety Cases

Safety case [3,4] is a structured argument allowing to justify system safety. The Goal Structuring Notation [3] allows the engineers to present the safety case in a graphical systematic way and define the evidences justifying system safety as a decomposition of the top level safety goal. Safety cases are widely adopted in industrial practice and recommended by various standards [3].

We demonstrate how to apply the systems-theoretic approach [4] to define the structure of the safety case that enables integrated analysis of safety and security. We show how to derive safety and security constraints in a systematic manner by the decomposition of the top-level safety goals.

Essentially a safety case constructed using GSN consists of the goals, strategies and solutions. The goals are propositions in an argument that can be said to be true or false (e.g., claims of requirements to be met by a system). The solutions contain the information extracted from an analysis, testing or simulation of a system showing that the goals have been met. Finally, the strategies are reasoning steps describing how the goals are decomposed and addressed by the sub-goals.

Thus, a safety case constructed in GSN presents a decomposition of the given safety goals into sub-goals until they can be supported by the direct evidence (a solution). It also explicitly defines the argument strategies, relied assumptions, the context in which the goals are declared, as well as justification for the use of a particular goal strategy.

III. Proposed Approach

To enables an integrated derivation of safety and security constraints, we propose a general pattern for structuring safety case. Our pattern takes an inspiration from the STAMP approach [4]. STAMP views safety as a control problem rather than a reliability problem.

STAMP is built on top of three basic constructs: safety constraints, hierarchical safety control structures and process models. In STAMP, the systems are viewed as the dynamic entities that are continually adapting to achieve their goals and to embrace their own changes and in the environment surrounding them. STAMP analysis consists of two steps. The first step focuses on is identifying the unsafe control actions that can lead to system unsafe behavior. The second step aims at discovering the potential scenarios leading to unsafe control, which results identifying the additional safety requirements.

By applying STAMP, we define three main groups of causes leading to unsafe control actions as follows:

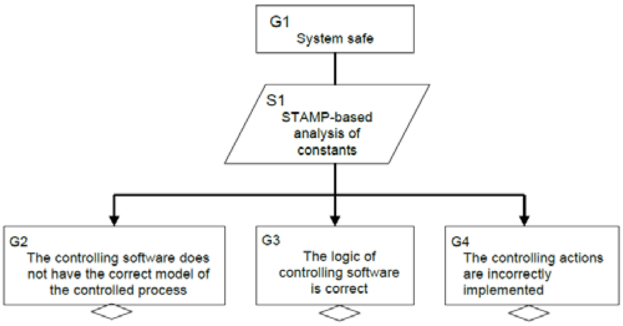
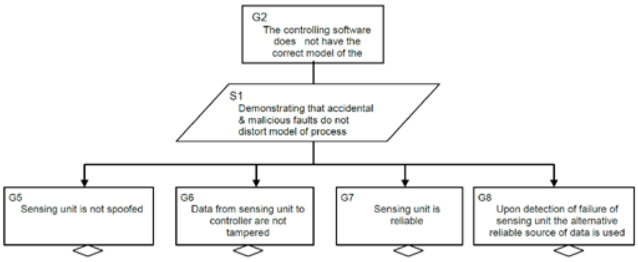

G1
System safe
S1
STAMP-based analysis of constants
G2
The controlling software does not have the correct model of the controlled process
G3
The logic of controlling software is correct
G4
The controlling actions are incorrectly implemented

Figure 1. General structure of a safety case using STAMP.

1. the controlling software does not have the correct model of the controlled process
2. the logic of controlling software is incorrect
3. the controlling actions are incorrectly implemented.

The top-level safety goal can be decomposed into three high-level sub-goals aiming at precluding occurrence of each of these classes of causes. The fragment of the safety case in the Goal Structuring Notation showing such a decomposition is given in Fig.1. Each class of constraints defines the corresponding goal that needs to be decomposed further.

The defined goals serve as the basis for the analysis of system architecture and deriving the safety and security constraints required to provide the evidences for achieving the corresponding goal. For instance, the goal **G2** – can be further decomposed as shown in Fig. 2.


G2
The controlling software does not have the correct model of the
S1
Demonstrating that accidental & malicious faults do not distort model of process
G5
Sensing unit is not spoofed
G6
Data from sensing unit to controller are not tampered
G7
Sensing unit is reliable
G8
Upon detection of failure of sensing unit the alternative reliable source of data is used

obtained by the sensors is unaltered by the network. This implies that the sensing unit should be authenticated and the sensed data unaltered.

In the similar way, we have to ensure that the accidental failures are detected and the error recovery procedure triggered.

The other goals can be treated in the similar way and result in derivation of safety and security requirements in the integrated manner.

IV. Discussion

Our current work focuses on defining a structured way to perform the data flow analysis of the control flow. We have proposed [5] to apply HAZOP to analyse the impact of security failures on the data flow. HAZOP performed over the data attributes defined in DFD of a critical system provides us with a structured methodology to analyse causes and consequences of the possible deviations of the data attributes. Per se, it enables a compositional integrated analysis of the impact of accidental and malicious failures on system safety.

The problem of safety and security interactions has recently received a significant research attention. It has been recognized that there is a clear need for the approaches facilitating an integrated analysis of safety and security [1] [2] [6].

This issue has been addressed by several techniques demonstrating how to adapt the traditional safety techniques like FMECA as well as formalize safety-driven engineering of critical systems [6][7] [8][9][10][11][12][13].

In this paper, we have discussed the techniques that facilitate an integrated analysis of safety and security. Further steps, in particularly supporting the development of integrated standards are still needed to address this problem.

References

[1] Troubitsyna, E., Laibinis, L., Pereverzeva, I., Kuismin, T., Ilic, D., Latvala, T.: Towards security-explicit formal modelling of safety-critical systems. In: SAFECOMP 2016, Proceedings. LNCS, vol. 9922, pp. 213–225. Springer (2016)

[2] Vistbakka, I., Troubitsyna, E., Kuismin, T., Latvala, T.: Co-engineering safety and security in industrial control systems: A formal outlook. In: Software Engineering for Resilient Systems - 9th International Workshop, SERENE 2017, Proceedings. LNCS, vol. 10479, pp. 96–114. Springer (2017)

[3] T.P. Kelly, “Arguing Safety -- a Systematic Approach to managing safety cases”, Ph.D. dissertation, Department of Computer Science, University of York, 1998.

[4] Prokhorova, Y., Laibinis, L., Troubitsyna, E.: Facilitating Construction of Safety Cases from Formal Models in Event-B. Information & Software Technology 60, 51–76 (2015)

[5] W. Young and N.G. Leveson, “An Integrated Approach to Safety and Security Based on Systems Theory.” Communication of ACM, vol. 57(2), 2014, pp. 31-35, doi: 10.1145/2556938.

[6] M. Steiner, and P. Liggesmeyer, “Combination of Safety and Security Analysis -- Finding Security Problems that Threaten the Safety of a System,” Proc. Workshop on Dependable, Embedded and Cyber-physical Systems, Sept. 2013, http://hal.archives-ouvertes.fr/SAFECOMP2013-DECS/hal-00848604.

[7] Sere, K., Troubitsyna, E.: Safety Analysis in Formal Specification. In: FM’99, Proceedings, Volume II. LNCS, vol. 1709, pp. 1564–1583. Springer (1999)

[8] Tarasyuk, A., Pereverzeva, I., Troubitsyna, E., Latvala, T., Nummila, L.: Formal Development and Assessment of a Reconfigurable On-board Satellite System. In: SAFECOMP 2012. pp. 210–222. Springer (2012)

[9] Tarasyuk, A., Troubitsyna, E., Laibinis, L.: Towards Probabilistic Modelling in Event-B. In: IFM 2010, Nancy, France. Proceedings. LNCS, vol. 6396, pp. 275– 289. Springer (2010)

[10] Tarasyuk, A., Troubitsyna, E., Laibinis, L.: Integrating Stochastic Reasoning into Event-B Development. Formal Asp. Comput. 27(1), 53–77 (2015)

[11] Troubitsyna, E.: Stepwise Development of Dependable Systems. Tech. rep. (2000)

[12] Troubitsyna, E., Vistbakka, I.: Deriving and Formalising Safety and Security Requirements for Control Systems. In SAFECOMP 2018, Proceedings. LNCS. Springer (2018) . In Press

[13] Laibinis L., Troubitsyna E, Fault Tolerance in a Layered Architecture: A General Specification Pattern in B. SEFM 2004, IEEE Computer, 2004.

Cyber-Physical Control Systems: Vulnerabilities, Threats, and Mitigations

Luis Garcia, Saman Zonouz
Electrical and Computer Engineering
Rutgers University
{lag266, saman.zonouz}@rutgers.edu

ABSTRACT

Cyber-Physical Systems (CPS) are yielding novel problems and solutions for security researchers. CPSs connect computerized controllers and human supervisors with physical systems used in the energy, transportation, water, manufacturing, and other sectors. Recent attacks against CPS, such as the Stuxnet virus, have prompted unprecedented investigation into new threats and mitigations against CPSs. However, Despite the increased interest in CPS security problems, the security community faces significant learning curves in addressing them. Modern CPSs are founded on control theory, real-time systems, and obscure, often ad-hoc programming practices. Furthermore, the traditional definitions of security are often in conflict with the goals and operational constraints of CPSs. A security measure that blocks a system operator from executing a critical action could cause as much or more damage than an actual attack!

We provide an introduction to the most basic and widely deployed application of CPS, control systems, and the emerging problems in their security. We begin with a deep dive description of how control systems are built. This includes Supervisory Control and Data Acquisition (SCADA) architectures, state estimation, and logic controller programming. The participants will come away understanding what is under the hood of a typical control system, and why they work the way they do.

Before going into the security-specific issues with control systems, we provide some motivating examples of real world control system attacks. In particular, we focus on the Maroochy Shire water system attack, the Lodz Poland train derailment, and the Stuxnet virus. In each of these three attacks, the adversary capabilities and objectives, vulnerabilities, attack methods, and final outcomes differ significantly.

Given a solid understanding of how control systems are built, we continue with attacks and vulnerabilities against control systems. Some of these are classic memory exploits and network protocol flaws. However, control systems introduce new classes of attacks as well as new challenges for attackers. To this end, we will cover both attacks and defenses for False Data Injection (FDI), code injection on Programmable Logic Controllers (PLCs), and infiltration of human machine interfaces. One of the most important themes in understanding these attacks is how adversaries must have some understanding of the dynamics of the victim control process. Participants will come away from this section understanding the similarities and differences between attacks on information systems and control systems, as well as what the open research problems are for control system offenses and defenses.

With the above coverage of control systems and their security issues, we briefly review several important topics in arguably the largest and most critical control system currently under development: the *smart grid*. In particular, we look at threats and vulnerabilities ranging from theft of electric service to large-scale disruptions of power. Smart meter privacy issues will also be covered.

We finish by reviewing several recent advances in the general security of control systems. We focus on intrusion detection methods that leverage the regularity of control system behavior, and program analysis techniques for real-time embedded controller code.

1. PROBLEM OVERVIEW

Cyber-physical critical infrastructures integrate networks of computation and physical processes to provide the society with essential functionalities and services. Distributed and embedded computers monitor the physical processes and, at the same time, control them, usually with feedback loops in which physical processes affect computations and vice versa. As a case in point, the power grid infrastructure is a vast and interconnected cyber-physical network for delivering electricity from generation plants to end-point consumers. Protecting the critical infrastructures is a vital necessity because the failure of these systems would have a debilitating impact on economic security and public health and safety.

Due to the insufficiency of the deployed protection solutions, there have been several large-scale outages [26]. For instance, the August 2003 blackout was caused by several unrelated interacting factors such as a transmission line outage as a result of a line-tree contacts followed by computer crashes preventing the control operators from finding out about the line outage, and hence taking corrective actions. This led to a cascading outage and ultimately the blackout, which affected around 50 million people and cost approximately 6 billion dollars [8]. While there was no malicious intent behind the 2003 blackout, it showed several cyber-physical system weakpoints and potential vulnerabilities that could be exploited by the attackers to cause the same catastrophic consequence. Additionally, there have been sophisticated attacks in other industrial settings, giving reason for concern. A recent and well-known example is the Stuxnet computer worm [7], which targeted Siemens industrial software used to control nuclear fuel processing plants. The worm exploited several extremely complicated cyber attack vectors, including four Windows zero-day vulnerabilities [25], to

DSN, CERTS '18

sabotage a suspected uranium processing facility. The scale and complexity of the attack clearly demonstrated the need to fully monitor cyber-physical critical infrastructures in real time for both accidental and malicious failures. Such monitoring would allow the power grid operators to take quick responsive and corrective actions if the power grid is under attack or has experienced failures.

Objectives: This paper aims at reviewing cyber-physical industrial control system networks, and critical power grid infrastructures specifically, cyber-physical vulnerabilities and threats as well as emerging mitigation and intrusion resilience techniques.

2. CONTROL SYSTEMS

The subject of modern control systems involves a broad set of topics in control theory, as well as a history of organically developed practices. Given the relative unfamiliarity of computer scientist and security experts with some of these topics, a solid understanding in the foundations and evolution of control systems is crucial for approaching their security problems. To this end, we will focus on two topics: the architecture and basic units of industrial control systems and the methods of embedded controller programming. These two topics break down as follows.

2.1 Industrial Control Systems

A control system is a computerized means of regulating the behavior of a physical process. The control system gives one or more computers the ability to manipulate physical equipment as needed to control various physical quantities, e.g., a computer may be able to control a heating element to regulate the temperature of a chemical reaction. Modern control systems network the controller computers with corporate networks for organization-wide control and data analytics.

Early control systems were not software based, but instead were implemented as relay circuits, in which some number of Boolean circuits executed in parallel over a set of sensor values from the physical process. The outputs of these circuits dictate the desired behavior of process machinery. Modern software-based controllers achieve the same ends, while being more easily reconfigurable than hardware relays. This has an important implication: *because modern automation controllers attempt to mimic hardware relays, their program design methodologies are significantly different from traditional general purpose computers.* This concept is covered more in the following section.

Large, geographically distributed control systems rely heavily on *state estimators*, which provide accurate measurements of physical quantities, even when some sensors may be erroneous. State estimation is emerging as an important topic in control system security.

In the tutorial, a smart grid process will be used as an example control system. This example will be used to demonstrate the basic concepts of control system architecture and programming methodologies.

2.2 Programmable Automation

The basic unit of automation in a control system is the Programmable Logic Controller (PLC). PLCs are directly connected to physical machinery and are responsible for the real-time control of the process. Additionally, PLCs aggregate process statistics for human operators, and execute sub-processes on their behalf. Many times per second, the PLC re-executes its control software in a procedure known as a *scan cycle*. A scan cycle consists of three steps: (*i.*) Measurements are read from plant sensors. (*ii.*) The software control program is executed over the sensor measurements. (*iii.*) The control program's output values are used to govern plant machinery.

The software control program executed in step (*ii.*) are typically written in graphical languages, the most popular of which is *Relay Ladder Logic* (RLL). In the tutorial, we will give an example of how to write a control program for the smart grid process using RLL. In this tutorial, we will highlight some of the inconsistencies in RLL program execution between vendors, and the weak, and in some cases, absent type systems used in writing PLC code. This is an important topic as it has implications for the analysis of PLC code.

3. CYBER-PHYSICAL VULNERABILITIES

The integration of the cyber and physical components in industrial control systems has resulted in several new cyber-physical system-specific vulnerabilities [3, 4, 14, 15, 18, 20–22].

We will review the existing hardware and software assets and the major involved vendors in the control system domain, and continue with presenting most important cyber-physical security issues in those assets specifically in power control networks. In particular, we will talk about how cyber network vulnerabilities, e.g., vulnerable state estimation server process, and physical system weaknesses, e.g., lack of power system $N-1$ reliability compliance [1], can be exploited simultaneously to cause a cyber-physical impact on the control system. Additionally, we will explain how the *availability* being the most important CIA criterion in most of critical infrastructures can introduce new attack surfaces. As a case in point, to keep control network available all the time and guarantee timely electricity delivery, control system operators face new operational constraints such as easy access to critical functionalities in the case of emergency that hinders deployment of security solutions such as strict global access control policy enforcement.

4. REAL-WORLD ATTACKS

We will cover several examples of real-world attacks against control systems. The first of these is the Maroochy Shire water breach in which a disgruntled former employee spilled nearly a quarter million gallons of sewage into public water ways. Second, we will review the Lodz train attack, in which a high schooler used a modified infrared remote control to manipulate train switches, ultimately derailing four train cars and injuring several people. Finally, we recap the 2009 Stuxnet attack, which leveraged a malicious payload to cause a PLC to harm physical equipment. In each case, we will review the attacker capabilities, leveraged vulnerabilities, and outcomes.

5. MITIGATION TECHNIQUES

We will discuss about the several mitigation techniques that have been proposed in the literature that attempt to mitigate the existing threats against cyber-physical systems.

5.1 Trustworthy Architectures

Following the introduced cyber-physical vulnerabilities, there have many secure architectures proposed for critical infrastructures generally [19, 33, 43]. In this tutorial, we will review the trustworthy control network architectures that have been recommended by several agencies, e.g., NIST [32], and NERC [1] as well as researchers [12, 13]. Additionally, we will talk about our current ongoing project on architectural programmable logic controller protection solution that can stop recent real-world intrusions such as Stuxnet worm. In particular, the solution is deployed as an embedded device sitting between the HMI server and PLC device in the control system and investigates every PLC code that is uploaded by the HMI server using static code analyses and formal verification techniques. If the code is detected to be malicious, the code upload request is rejected and the code along with a violating input vector is sent back to the HMI operator for debugging purposes. Otherwise, the code is uploaded on the PLC for execution and control of the physical system.

5.2 Online Security Assessment

There have been several online security assessment solutions proposed for cyber-physical infrastructures [37, 39, 44]. In this tutorial, we will discuss about security/safety contingency analysis techniques in power systems that has been explored by many researchers in the past (see [31] for a comprehensive survey). A contingency is defined as an accidental or malicious failure/incident on the power network, e.g., an unexpected loss of a power transmission line as a result of an attack or a thunderstorm. The initial efforts were based on first-order performance index sensitivities, i.e., a measure of how critical each incident is given the network topology, to rank contingencies [6]. There have been several follow-up attempts to improve the ranking quality by considering higher order sensitivities [11, 23]. Furthermore, there has been an increasing interest in the analysis of multiple contingencies [9, 10] after the introduction of new NERC standards [27], e.g., a linear sensitivity-based approximate measure of how close the power system is brought to islanding by a particular outage contingency [5]. Recently, there have been security-oriented cyber-physical contingency analysis solutions proposed [2,35] that take into account both cyber and power network topologies to analyze the potential impact of possible cyber attacks on the physical system and consequently come up with the ranked list of contingencies, i.e., cyber vulnerability exploitations and power system contingencies.

5.3 Cyber-Physical Attack Detection

To terminate malicious compromises, several attack detection solutions using cyber and physical sensors have been proposed [16, 17,34,38,40,41]. In this talk, we will review few specific intrusion detection frameworks that concentrate on cyber as well as physical aspects of the control networks. Almost every detection framework includes a solution to the problem of hybrid cyber-physical security modeling of the power-grid [24]. Pasqualetti [28] model a power system under cyber-physical attack as a linear time-invariant descriptor system with unknown inputs, and design a dynamic detection and identification scheme using geometric control theoretic tools. Sridhar et al. [30] review how traditional intrusion detection techniques could be applied in cyber-physical settings, and introduce a layered approach to evaluate risk based on the current state of the power-grid. Recently, cyber-physical detection solutions against false data injection attacks have been proposed that fuse uncertain information from different types of distributed sensors, such as power system meters and cyber-side intrusion detectors, to detect the malicious activities within the cyber-physical system [38]. Specifically, such security-oriented cyber-physical state estimation engines, at each time instant, identify the compromised set of hosts in the cyber network and the maliciously modified set of measurements obtained from power system sensors.

5.4 Proactive Cyber-Physical Intrusion Tolerance

Preserving the availability and integrity of networked computing systems in the face of those fast-spreading intrusions requires advances not only in detection algorithms, but also in intrusion tolerance and automated response techniques. Additionally, the complexity and connectivity of control networks, and their recently increasing integrations with physical systems signify the quest for systems that detect their own compromises and failures and automatically repair themselves. In particular, the ultimate goal of the intrusion tolerant system design is to adaptively react against malicious attacks in real-time, given offline knowledge about the network's topology, and online alerts and measurements from system-level sensors. Cyber-physical intrusion tolerance solutions are relatively less investigated as they require sufficiently accurate attack detection tools that are currently nonexistent; however, there have been several research attempts in the area [29, 36, 42] using extended attack trees called attack-response tree formalism that not only formulate possible attack vectors but also represents possible system and network-level response and recovery actions that can be taken if the network is partially compromised/down.

6. DISCUSSION

Finally, we will conclude the tutorial by highlighting three main points. First, using examples, we will discuss the fact that not every traditional IT security solution fits the cyber-physical security problem, and hence new effective security solutions are required for particular critical infrastructure protection problems. Second, we will describe several top and emerging research problems that remain open in the cyber-physical system security field, and need more research investments in the next few years. Finally, we will discuss different parties, such as government, industry and academic institutes, that are interested in or currently funding and/or working on various aspects of critical infrastructure security problem.

7. REFERENCES

[1] North American Electric Reliability Corporation, CIP-002-5 Cyber Security, 2012.

[2] Berthier, R., Bobba, R., Davis, M., Rogers, K., and Zonouz, S. State estimation and contingency analysis of the power grid in a cyber-adversarial environment. *NIST Workshop on Cybersecurity for Cyber-Physical Systems* (2012).

[3] Brundle, M., and Naedele, M. Security for process control systems: An overview. *IEEE Security Privacy 6* (2008), 24–29.

[4] Cárdenas, A. A., Amin, S., and Sastry, S. Research challenges for the security of control systems. In *HotSec* (2008).

[5] Davis, C., and Overbye, T. Multiple element contingency screening. *IEEE Transactions on Power Systems 26*, 3 (2011), 1294–1301.

[6] Ejebe, G. C., and Wollenberg, B. F. Automatic contingency selection. *IEEE Transactions on Power Apparatus and Systems 65*, 1 (1979), 859–109.

[7] Falliere, N., Murchu, L. O., and Chien, E. W32.Stuxnet dossier. Tech. rep., Symantic Security Response, 2010.

[8] Force, U. C. P. S. O. T. Final report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations, available at `https://reports.energy.gov/BlackoutFinal-Web.pdf`, 2003.

[9] Guler, T., and Gross, G. Detection of island formation and identification of causal factors under multiple line outages. *IEEE Transactions on Power Systems 22*, 2 (2007), 505–513.

[10] Halpin, T., Fischl, R., and Fink, R. Analysis of automatic contingency selection algorithms. *IEEE Transactions on Power Apparatus and Systems PAS-103*, 5 (may 1984), 938–945.

[11] IRISARRI, G., AND SASSON, A. An automatic contingency selection method for on-line security analysis. *IEEE Transactions on Power Apparatus and Systems PAS-100*, 4 (1981), 1838–1844.

[12] KRUTZ, R. L. *Securing SCADA systems*. John Wiley & Sons, 2005.

[13] LANGNER, R. *Robust control system networks*. Momentum Press, 2011.

[14] MCDANIEL, P., AND MCLAUGHLIN, S. Security and privacy challenges in the smart grid. *Security & Privacy, IEEE 7*, 3 (2009), 75–77.

[15] MCLAUGHLIN, S. On dynamic malware payloads aimed at programmable logic controllers. In *Proceedings of the 6th USENIX conference on Hot topics in security. HotSec* (2011), vol. 11, pp. 10–10.

[16] MCLAUGHLIN, S., HOLBERT, B., FAWAZ, A., BERTHIER, R., AND ZONOUZ, S. A multi-sensor energy theft detection framework for advanced metering infrastructures. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 31*, 7 (2013), 1319.

[17] MCLAUGHLIN, S., HOLBERT, B., ZONOUZ, S., AND BERTHIER, R. Amids: A multi-sensor energy theft detection framework for advanced metering infrastructures. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on* (2012), IEEE, pp. 354–359.

[18] MCLAUGHLIN, S., AND MCDANIEL, P. Sabot: specification-based payload generation for programmable logic controllers. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), pp. 439–449.

[19] MCLAUGHLIN, S., MCDANIEL, P., AND AIELLO, W. Protecting consumer privacy from electric load monitoring. In *Proceedings of the 18th ACM conference on Computer and communications security* (2011), ACM, pp. 87–98.

[20] MCLAUGHLIN, S., PODKUIKO, D., DELOZIER, A., MIADZVEZHANKA, S., AND MCDANIEL, P. Embedded firmware diversity for smart electric meters. In *Proceedings of the 5th USENIX Workshop on Hot Topics in Security (HotSec 2010), Washington DC* (2010).

[21] MCLAUGHLIN, S., PODKUIKO, D., AND MCDANIEL, P. Energy theft in the advanced metering infrastructure. In *Critical Information Infrastructures Security*. Springer, 2010, pp. 176–187.

[22] MCLAUGHLIN, S., PODKUIKO, D., MIADZVEZHANKA, S., DELOZIER, A., AND MCDANIEL, P. Multi-vendor penetration testing in the advanced metering infrastructure. In *Proceedings of the 26th Annual Computer Security Applications Conference* (2010), ACM, pp. 107–116.

[23] MIKOLINNAS, T., AND WOLLENBERG, B. An advanced contingency selection algorithm. *IEEE Transactions on Power Apparatus and Systems PAS-100*, 2 (feb. 1981), 608–617.

[24] MO, Y., KIM, T. H.-J., BRANCIK, K., DICKINSON, D., LEE, H., PERRIG, A., AND SINOPOLI, B. Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE 100*, 1 (jan. 2012), 195–209.

[25] NARAINE, R. Stuxnet attackers used 4 Windows zero-day exploits, available at `http://www.zdnet.com/blog/security/stuxnet-attackers-used-4-windows-zero-day-exploits/7347`, 2010.

[26] NERC. 2009 NERC disturbance index, available at `http://www.nerc.com/files/disturb09.pdf`, 2010.

[27] NERC. System performance following loss of two or more bulk electric system elements (category c), 2005.

[28] PASQUALETTI, F., DÖRFLER, F., AND BULLO, F. Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design. *CoRR abs/1103.2795* (2011).

[29] SANDERS, W. H., CAMPBELL, R. H., ABDELZAHER, T. F., KHURANA, H., AND JOSHI, K. R. Game-theoretic intrusion response and recovery.

[30] SRIDHAR, S., HAHN, A., AND GOVINDARASU, M. Cyber-physical system security for the electric power grid. *Proceedings of the IEEE 100*, 1 (2012), 210–224.

[31] STOTT, B., ALSAC, O., AND F.L., A. Analytical and computational improvements in performance index ranking algorithms for networks. *International Journal of Electrical Power and Energy Systems 7*, 3 (1985), 154–160.

[32] STOUFFER, K., FALCO, J., AND SCARFONE, K. Guide to industrial control systems (ics) security. *NIST Special Publication 800*, 82 (2008), 16–16.

[33] YANG, W., LI, N., QI, Y., QARDAJI, W., MCLAUGHLIN, S., AND MCDANIEL, P. Minimizing private data disclosures in the smart grid. In *Proceedings of the 2012 ACM conference on Computer and communications security* (2012), ACM, pp. 415–427.

[34] ZIMMER, C., BHAT, B., MUELLER, F., AND MOHAN, S. Time-based intrusion detection in cyber-physical systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems* (2010), ACM, pp. 109–118.

[35] ZONOUZ, S., DAVIS, M., DAVIS, K., BERTHIER, R., BOBBA, R. B., AND SANDERS, W. Socca: A security-oriented cyber-physical contingency analysis in power infrastructures. *submitted to IEEE Transactions on Smart Grid (minor revision)* (2013).

[36] ZONOUZ, S., HOUMANSADR, A., AND HAGHANI, P. Elimet: Security metric elicitation in power grid critical infrastructures by observing system administrators' responsive behavior. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on* (2012), IEEE, pp. 1–12.

[37] ZONOUZ, S., AND MIREMADI, S. G. A fuzzy-monte carlo simulation approach for fault tree analysis. In *Reliability and Maintainability Symposium, 2006. RAMS'06. Annual* (2006), IEEE, pp. 428–433.

[38] ZONOUZ, S., ROGERS, K., BERTHIER, R., BOBBA, R., SANDERS, W., AND OVERBYE, T. Scpse: Security-oriented cyber-physical state estimation for power grid critical infrastructures. 1790–1799.

[39] ZONOUZ, S. A., BERTHIER, R., AND HAGHANI, P. A fuzzy markov model for scalable reliability analysis of advanced metering infrastructure. In *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES* (2012), IEEE, pp. 1–5.

[40] ZONOUZ, S. A., JOSHI, K. R., AND SANDERS, W. H. Cost-aware systemwide intrusion defense via online forensics and on-demand detector deployment. In *Proceedings of the 3rd ACM workshop on Assurable and usable security configuration* (2010), ACM, pp. 71–74.

[41] ZONOUZ, S. A., JOSHI, K. R., AND SANDERS, W. H. Floguard: cost-aware systemwide intrusion defense via online forensics and on-demand ids deployment. In *Computer Safety, Reliability, and Security*. Springer, 2011,

pp. 338–354.

[42] Zonouz, S. A., Khurana, H., Sanders, W. H., and Yardley, T. M. Rre: A game-theoretic intrusion response and recovery engine. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on* (2009), IEEE, pp. 439–448.

[43] Zonouz, S. A., and Sanders, W. H. A kalman-based coordination for hierarchical state estimation: Agorithm and analysis. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual* (2008), IEEE, pp. 187–187.

[44] Zonouz, S. A., Sanders, W. H., Yardley, T., Berthier, R., and Khurana, H. Seclius: An information flow-based, consequence-centric security metric. *IEEE Transactions on Parallel and Distributed Systems* (2013), 1.

Using Schedule-Abstraction Graphs for the Analysis of CAN Message Response Times

Mitra Nasri, Arpan Gujarati, and Björn B. Brandenburg
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

***Abstract*—In a controller area network (CAN), electromagnetic interference (EMI) can result in message corruptions during transmission. The CAN protocol thus checks each message for corruption (using a checksum) and automatically schedules an erroneous message for retransmission. Retransmissions help tolerate EMI-induced errors with very high probability, but since EMI is stochastic in nature, they can affect the timing properties of the system in unpredictable ways. This work is about effectively quantifying the impact of retransmissions on the schedulability of real-time systems. Prior work focused on coarse-grained worst-case response-time analysis (RTA) of periodic or sporadic CAN message streams in the presence of retransmissions. For example, in case of periodic streams, prior analyses help upper-bound the maximum response time that any message belonging to a message stream might incur in the presence of a specific number of retransmissions. In this work, we present a fine-grained approach to analyze CAN message response times in the presence of retransmissions. The proposed analysis is based on the exploration of schedule-abstraction graphs, a novel abstraction for concisely capturing all possible schedules of CAN messages. Therefore, it enables upper-bounding the response times for each individual CAN message, and is not restricted to only periodic or sporadic message streams. We demonstrate the benefits of a message-specific analysis with a case study based on a simple mobile robot message set, and also discuss future opportunities enabled by such an analysis.**

I. Introduction

Embedded systems often need to operate in harsh environments, e.g., automotive embedded systems are surrounded by spark plugs and electric motors, industrial embedded systems in many cases are deployed in close vicinity to high-power machinery, and autonomous robots may need to operate in radiation-prone environments [1]. As a result, such embedded systems are susceptible to electromagnetic interference (EMI) and must be designed to withstand its effects [2].

In the context of real-time networked systems, EMI may result in frequent message corruptions on the network. To mitigate the effects of such corruptions, network stacks typically detect and retransmit the corrupted messages. For example, in a Controller Area Network (CAN), CAN controllers automatically queue messages for retransmission if any host signals a transmission error [3]. However, retransmissions may sometimes have a negative effect on the system reliability. Since EMI is stochastic in nature, retransmissions affect the timing behavior of the system in unpredictable ways, and may eventually compromise its functional safety due to deadline misses. This work is about effectively quantifying the effect of retransmissions on schedulability of CAN-based real-time systems.

Prior work in this regard [4, 5, 6, 7, 8, 9, 10] has focused on coarse-grained worst-case response-time analysis (RTA) of periodic and sporadic CAN message streams in the presence of retransmissions. For example, in case of periodic streams, prior analyses help upper-bound the maximum response time that any message belonging to a message stream might incur in the presence of a specific number of retransmissions.

The aforementioned analyses primarily rely on the classical technique of computing response times through fixed-point analysis of an iterated function [11]. As a result, although effective in detemining if a message set ever misses any deadline, they are often too coarse-grained for many use cases. For example, in case of weakly-hard systems [12] that can tolerate a few deadline violations (such as real-time control systems), a separate response-time analysis is required for each message in the message stream; as shown by Bernat et al. [12, 13], this is non-trivial with a fixed-point analysis. Even in the case of reliability analysis, using message stream-specific analyses to upper-bound the probability of a message transmission failure results in extremely pessimistic bounds on system realibiltiy [14].

Empirical approaches, e.g., [15, 16], provide detailed profiles of message-specific response times, but are not provably safe and thus problematic in the context of safety-critical systems.

In this work, we present a new approach based on schedule-abstraction graphs [17] to analyse both best-case and worst-case response times of individual messages belonging to each message stream in the workload. Schedule-abstraction graphs [17] are a recently proposed abstraction for concisely capturing all possible schedules of CAN messages. Therefore, they enable upper-bounding the response time of each individual CAN message with both release jitters and offsets, and are not restricted to only periodic or sporadic message streams.

To the best of our knowledge, this is the first instance of an analysis for CAN messages that explores all possible schedules while taking retransmissions into account.

The rest of paper is organized as follows. We start with an overview of prior work on schedule-abstraction graphs that is necessary to understand the proposed analysis (§II), and then explain in detail our retransmissions-aware analysis for CAN messages (§III). We then demonstrate the benefits of the proposed message-specific analysis with a case study based on a simple mobile robot message set and discuss some interesting open problems (§IV). Finally, we conclude with a brief discussion of future work (§V).

II. Schedule-Abstraction Graphs

The CAN protocol schedules message transmissions in the order of their fixed priorities [3]. Thus, schedulability analysis of CAN messages can be reduced to the problem of schedulability analysis of fixed-priority non-preemptive jobs on a uniprocessor platform.

In particular, we base our analysis on a recently proposed exact schedulabiltiy analysis of non-preemptive fixed-priority jobs [17], which uses *schedule-abstraction graphs* for efficiency. Traditionally, exact schedulability analyses have been developed for preemptive jobs using state exploration techniques based on model checking, timed automata, or linear-hybrid automata [18, 19, 20, 21]. These techniques do not apply to non-preemptive jobs and their sclability is limited. In contrast, the schedule-abstraction graph model, along with an effective merging strategy [17], allows for a more efficient representation and exploration of all possible uniprocessor schedules. In the following, we give a brief overview of this analysis.

Suppose a finite set of jobs $\mathcal{J}$ to be scheduled on a uniprocessor based on their priorities. Each job $J_i = ([r_i^{min}, r_i^{max}], [C_i^{min}, C_i^{max}], d_i, p_i)$ can be released at any time $r_i \in [r_i^{min}, r_i^{max}]$, has a transmission time of $C_j \in [C_i^{min}, C_i^{max}]$, an absolute deadline d_i, and a fixed priority p_i (a numerically lower value denotes a higher priority).

The schedule-abstraction graph for a job set $\mathcal{J}$ is a directed acyclic graph $G = (V, E)$ consisting of vertices V denoting the system states and edges E denoting job executions. Each edge (v_p, v_q, J_i) is directed from source vertex v_p to destination vertex v_q and is labeled with a job $J_i \in \mathcal{J}$, impliying that job J_i is executed between system states v_p and v_q, i.e., it is dispatched *next* after v_p or that it *succeeds* v_p. A system state $v_p = [A_p^{min}, A_p^{max}]$ represents an interval during which the processor is *possibly* available and at the end of which the processor is *certainly* available. The graph is rooted at the *initial* state $v_1 = [0, 0]$ denoting an idle processor. A possible schedule of any job set $\mathcal{J}^P \subseteq \mathcal{J}$ is modeled as a path P from the initial state v_1 to any state v_p such that the set of labels of edges on this path corresponds to $\mathcal{J}^P$.

Using the aforementioned graph-based abstraction for schedules, and given a particular scheduling policy, the BCRTs and the WCRTs of the jobs are determined through an iterative algorithm consisting of an expansion phase and a merging phase. In particular, during each iteration, a path P ending at vertex v_p is first *expanded* by deriving all jobs that can potentially be dispatched *next* after v_p through at least one valid execution scenario, i.e., through some valid assignment of release times and excition times. Afterwards, if any two paths P_1 and P_2 share the same set of jobs (i.e., $\mathcal{J}^{P_1} = \mathcal{J}^{P_2}$), their terminal vertices are compared. If the respective terminal vertices, say, v_p and v_q correspond to intersecting processor-availability intervals (i.e., $v_p \cap v_q \neq \emptyset$), then they are *merged* together to create a new state v_{pq} whose processor-availability interval is the union of the two (i.e., $v_{pq} = v_p \cup v_q$). In the merged state, paths P_1 and P_2 both terminate at vertex v_{pq}. The algorithm terminates when each path corresponds to $\mathcal{J}$.

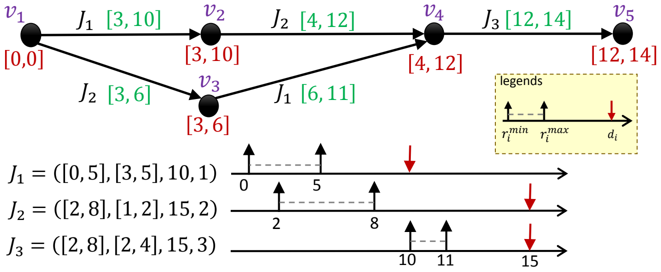


Fig. 1. A schedule-abstraction graph for three jobs $\mathcal{J} = \{J_1, J_2, J_3\}$. An interval after a job on an edge denotes the earliest and latest finish time of the job on that edge and an interval below a vertex represents when the processor becomes possibly and certainly ready in that state.

While constructing the graph, the algorithm stores the earliest and latest finish time of a job J_i for each edge $e = (v_p, v_q, J_i) \in P$, where J_i is dispatched next after v_p. Hence, upon the termination of the algorithm, the BCRT and the WCRT of job J_i is trivially obtained by finding its minimum and maximum finish times over all paths in G.

Example. Suppose that a given job set $\mathcal{J}$ consists of three jobs $J_1 = ([0, 5], [3, 5], 10, 1)$, $J_2 = ([2, 8], [1, 2], 15, 2)$, and $J_3 = ([2, 8], [2, 4], 15, 3)$. The schedule-abstraction graph for this job set is illustrated in Fig. 1. Due to the release jitter of J_1 and J_2, it is possible for either of them to be scheduled first, e.g., job J_2 is scheduled first if J_2 is released at time 3 and J_1 is released at time 4. Both possibilities are modeled in the graph by two outgoing edges labeled J_1 and J_2 incident to the initial state.

For each edge, an earliest and a latest finish time of the job that is scheduled first is calculated. These values are determined by taking into account the time at which a higher-priority job will certainly be released, the candidate job's release jitter interval and execution time variation, and the processor-availability interval of the source state of the edge. For example, when J_2 succeeds v_1 (and creates state v_3), its earliest start time is 2 and its latest start time is 4 since at time 5, a higher-priority job, i.e., J_1, will be certainly released and hence J_2 cannot be the highest-priority pending job from time 5 onward. Thus, the earliest and latest finish times of J_2 when it succeeds state v_1 are $2+1 = 3$ and $4+2 = 6$, respectively. On the other hand, if J_1 succeeds v_1, its earliest and latest finish times are 3 and 10 since its earliest and latest release times are 0 and 5, respectively, and its shortest and longest execution times are 3 and 5, respectively. When J_2 is dispatched after J_1, i.e., it succeeds state v_3, its earliest and latest start times will be 3 and 10, since the earliest and latest times, respectively, at which the processor becomes available after executing J_1 are 3 and 10. Using these values, the earliest and latest finish times of J_2 when it succeeds v_3 are 4 and 12, respectively.

Since both paths $\langle J_1, J_2 \rangle$ and $\langle J_2, J_1 \rangle$ share the same set of jobs and their intervals intersects, i.e., $[4, 12] \cap [6, 10] \neq \emptyset$, their terminal states are merged to obtain state v_4. Since J_3 is the only pending job at state v_4, the graph is trivially extended with edge (v_4, v_5, J_3). From this graph, the response time of J_1 along the path $\langle v_1, v_2, v_4, v_5\rangle$ is $R_1 \in [3, 10]$, whereas its response time along the path $\langle v_1, v_3, v_4, v_5\rangle$ is $R_1 \in [6, 11]$. Thus, J_1's BCRT and WCRT is $\min(3, 6) = 3$ and $\max(10, 11) = 11$, respectively. Jobs J_2 and J_3's response-time bounds can be computed similarly.

A detailed proof of correctness of the analysis can be found in [17]. Further, the paper provides a thorough discussion on how to use the proposed schedulability analysis for periodic tasks with constrained deadlines and/or release offsets.

III. Retransmissions-Aware Analysis

Using the schedule-abstraction graph-based analysis discussed in §II as a black box, we next propose a technique to analyze CAN message response times while accounting for fault-induced retransmissions.

Consider a finite set of CAN messages $\mathcal{M}$ where each message $M_i \in \mathcal{M}$ can be released at any time $r_i \in [r_i^{min}, r_i^{max}]$, has a transmission time of $C_j \in [C_i^{min}, C_i^{max}]$, an absolute deadline d_i, and a fixed priority $p_i \geq 1$. Release time variation for CAN messages is common due to scheduling delays, queuing delays, buffering, etc. Transmission time variation occurs due to changes in data values or bit-stuffing[1] and is typically small. For an 8-byte data frame, for example, the frame size can vary between 108 and 126 bits.

Without retransmissions, an exact value of the WCRTs can be trivially estimated using schedule-abstraction graphs since $\mathcal{J} \triangleq \mathcal{M}$. With retransmissions though, deriving an exact WCRT or even an upper bound on the exact WCRT of a message is challenging since errors that cause retransmissions happen in a non-deterministic way. In the following, we propose an analysis to derive an upper bound on the exact WCRT of any message $M_i \in \mathcal{M}$ given that up to f retransmissions may happen during the time the message set is transmitted over the network. We denote the exact value of this retransmissions-aware WCRT of each message M_i as $R_i(f)$.[2]

We assume that each host transmitting messages on CAN has enough buffer to store all pending messages, including the messages that are scheduled for retransmission.

Analysis. A schedule of $\mathcal{M}$ with f retransmissions means that a total of $n + f$ messages are actually transmitted over CAN, including n successful transmissions and f erroneous transmissions. Thus, to use schedule-abstraction graphs, we consider a revised message set $\mathcal{M}' = \mathcal{M} \cup \mathcal{M}^f$ where $\mathcal{M}^f = \{M_{n+1}, M_{n+2}, \ldots, M_{n+f}\}$. Namely, each message in $\mathcal{M}$ represents a *successful transmission* and each message in $\mathcal{M}^f$ represents an erroneous transmission over the network.

Messages in $\mathcal{M}^f$ are defined as follows. Since messages can be corrupted at any time in a non-deterministic way, we model the release of each message in $\mathcal{M}^f$ using the release jitter interval $[r^{min}, d^{max}]$, where $r^{min} \triangleq \min\{r_i^{min} \mid M_i \in \mathcal{M}\}$ denotes the earliest possible release event and $d^{min} \triangleq \max\{d_i \mid M_i \in \mathcal{M}\}$ denotes the latest possible deadline in message set $\mathcal{M}$. Since any message in $\mathcal{M}$ can be corrupted, the transmission times of messages in $\mathcal{M}^f$ are modeled as ranging from $C^{min} \triangleq \min\{C_i^{min} \mid M_i \in \mathcal{M}\} + \epsilon$ up to $C^{max} \triangleq \max\{C_i^{max} \mid M_i \in \mathcal{M}\} + \epsilon$. The error overhead ϵ denotes the time corresponding to the transmission of an error frame on the network, which happens immediately after the transmission of an erroneous message. We also set the priority of each erroneous message to the highest priority, i.e., zero, to model the corruption of the highest-priority message in the worst case (recall that $\forall M_i \in \mathcal{M}, p_i \geq 1$). The deadline of each erroneous message is irrelevant and set to ∞.

To summarize, the k^{th} erroneous message M_{n+k} is defined as $M_{n+k} \triangleq \big([r^{min}, d^{max}], [C^{min}, C^{max}], \infty, 0\big)$. Thanks to this definition, an erroneous message **(i)** can be released *anytime* in the window of interest, i.e., $[r^{min}, d^{max}]$, and **(ii)** has a priority higher than any message in $\mathcal{M}$ and hence can be transmitted before any message in $\mathcal{M}$. As a result, the transmission of erroneous messages can affect the WCRT of *any successfully transmitted message*. Invoking the schedule-abstraction graph-based analysis [17] with message set $\mathcal{M}'$ thus yields a safe upper bound on the WCRT of messages in $\mathcal{M}$, given that they may be affected by up to f retransmissions. Note that we are not interested in the WCRT of the erroneous messages in $\mathcal{M}^f$, but only on their impact on the response times of correctly transmitted messages.

We next explain the schedule-abstraction graph generated for a set of CAN messages, including the erroneous messages that we model for a black-box analysis, using a simple example.

Example. Suppose that message set $\mathcal{M}$ consists of two messages $M_1 = ([0, 5], [3, 5], 14, 1)$ and $M_2 = ([6, 6], [1, 2], 30, 2)$. The schedule-abstraction graph for $f = 2$ is illustrated in Fig. 2. M_3 and M_4 denote the two erroneous messages in this case. The analysis keeps track of the largest observed value of WCRT of each correct message in all paths. In this example, $R_1(2) = 15$ and $R_2(2) = 22$, which means that M_1 misses its deadline at time 14. This happens when M_1 is transmitted after two erroneous trials, shown by the two paths $\langle M_3, M_4, M_1, M_2\rangle$ and $\langle M_4, M_3, M_1, M_2\rangle$. A scenario in which neither M_1 nor M_2 is transmitted erroneously is represented by the path $\langle M_1, M_2, M_3, M_4\rangle$. In this path, the response times of M_1 and M_2 are not affected by any retransmission. Note that message M_1 is always certainly released before message M_2 is released. Since M_1 has a higher priority than M_2, message M_2 can never be transmitted as long as M_1 has not been transmitted.

IV. Case Study

In this section, we demonstrate the benefits of the proposed fine-grained message-specific analysis with a case study based on a simple mobile robot message set. In particular, we show that the proposed analysis helps to **(i)** account for weakly-hard timing constraints, and to **(ii)** assign message offsets such that

[1] CAN controller inserts a bit of opposite polarity after five consecutive bits of the same polarity while transmitting any data on the network. This practice is called *bit stuffing*, and is necessary due to the non-return to zero (NRZ) coding used with CAN. The stuffed data frames are destuffed by the receiver.

[2] Even though f is non-deterministic in practice, it could be estimated through an empirical analysis of EMI on the CAN bus under different types of operational environments.

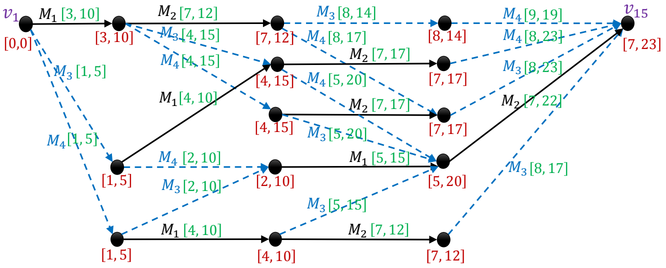


Fig. 2. A schedule-abstraction graph for $f = 2$ and $\mathcal{M} = \{M_1, M_2\}$, where $M_1 = ([0, 5], [3, 5], 14, 1)$ and $M_2 = ([6, 6], [1, 2], 30, 2)$. Dashed lines denote erroneous transmissions.

TABLE I
MOBILE ROBOT MESSAGE SET FROM [6]

Benchmark	id	priority	length(μs)	period(μs)	deadline(μs)
MotorCtrl	$\overline{M}_1$	1	288	2000	2000
Wheel1	$\overline{M}_2$	2	328	4000	4000
Wheel2	$\overline{M}_3$	3	328	4000	4000
RadioIn	$\overline{M}_4$	4	528	8000	8000
Proximity	$\overline{M}_5$	5	248	12000	12000
Logging	$\overline{M}_6$	6	528	240000	12000

the overall distribution of the WCRTs is improved. Later, we discuss some interesting open problems for future work.

The employed mobile robot benchmark [6] is designed for a CAN bus with a bit-transmission rate of 256 Kbps. It consists of six periodic message streams, denoted by $\overline{M} = \{\overline{M}_1, \overline{M}_2, \ldots, \overline{M}_6\}$ as listed in Table I.

We convert these periodic message streams to a finite set of messages (as required by the analysis in §III) as follows. Let T_i denote the periods of the respective message streams, and $H = lcm(T_1, T_2, \ldots, T_6)$ denote their hyperperiod, where lcm denotes the least common multiple. Starting with $\mathcal{M} = \emptyset$, for each message that belongs to message stream $\overline{M}_i$, say, the j^{th} message, and that is released during the hyperperiod, we add a new message $M_{i,j}$ to $\mathcal{M}$. Each message $M_{i,j}$ has a release interval $[(j-1)T_i, (j-1)T_i + \delta]$, where δ denotes the maximum release jitter of $\overline{M}_i$. The best-case transmission time for each message $M_{i,j}$ is $C^{min}_{i,j} = 72$, corresponding to the transmission time of the minimum-sized (one byte) packet, and the worst-case transmission time for each message $M_{i,j}$ is assigned as per the "length" column in Table I.

For our experiments, we considered $\delta = 10\mu$s and $f = 1$, i.e., each message is affected by up to one retransmission in a hyperperiod, or in other words, at most one message is erroneously transmitted. In practice, δ is upper-bounded through a careful analysis of the system processes that generate the messages to be transmitted, and f can be estimated with high confidence based on the peak rate of EMI, which in turn is known from empirical measurements and/or environmental modeling. All experimental results are illustrated in Fig. 3.

As mentioned in §I, the proposed analysis allows a detailed study of the WCRTs of each message stream in the workload. This enables the analysis of other higher-level properties and guarantees, for instance such as whether any message stream incurs more than three consecutive deadline misses (i.e., weakly-hard constraints [12]), or opportunities for a reduction in response-time jitter. We next discuss some relevant examples that demonstrate the use of the proposed analysis in these ways in the context of the case study.

Understanding the response-time distributions. Figs. 3(a)-3(e) show the WCRTs of the first 15 messages belonging to message streams $\overline{M}_1$-$\overline{M}_5$, respectively. For message stream $\overline{M}_6$, only a single message is transmitted during the hyperperiod with a WCRT of $R_{6,1}(1) = 3187\mu$s. We observe that the WCRTs for each message stream follow a repeating pattern depending on the relation between the different message periods. For example, as shown in Fig. 3(g), messages belonging to message stream $\overline{M}_5$ have three different types of WCRTs, and from the second message onward, the WCRTs alternate. In this case, $M_{5,1}$ has the largest WCRT compared to subsequent messages of $\overline{M}_5$ because its arrival time coincides with that of $M_{1,1}, M_{2,1}, \ldots, M_{6,1}$. Since the release jitter $\delta = 10\mu$s for each message, any of these messages can be scheduled before $M_{5,1}$. However, $M_{5,2}$ arrives at time 12000μs, when $M_{4,1}$, $M_{4,2}$, and $M_{6,1}$ are no longer pending and hence their transmission does not affect the WCRT of $M_{5,2}$. Consequently, after $M_{5,1}$'s transmission, the WCRT of each $M_{5,j}$ varies alternately depending on whether a message instance of $\overline{M}_4$ or $\overline{M}_6$ arrives at the same time as $M_{5,j}$. It is worth noting that in the presence of jitter, both lower- and higher-priority messages may interfere with a message instance.

Verifying weakly-hard constraints. Weakly-hard constraints are usually represented in the form of (m, k) constraints, i.e., *at least m message instances must arrive before their deadline in any consecutive sequence of k message instances* [12]. Hence, a fine-grained knowledge about the WCRT of message instances is required to evaluate whether or not a message stream conforms a weakly-hard timing constraint. Since our analysis derives the WCRT of each message during a hyperperiod, verifying an (m, k) constraint for a message stream $\overline{M}_i$ is equivalent to counting the number of deadline misses in each window of k messages and ensuring that at least m messages within that window are transmitted before their deadlines.

Reducing WCRTs using offset assignment. Assigning offsets allows a system designer to avoid creating a large amount of interference (i.e., long busy windows) and hence improves schedulability [22]. This, however, requires having a fine-grained WCRT analysis per message instance in order to understand which individual messages suffer from large interference and how message streams should be aligned using initial offsets so that their response times are reduced.

For example, the WCRT of messages in the mobile robot message set (Table I) can be reduced through offset assignment as follows. We use $O_1 = 0\mu$s, $O_2 = 0\mu$s, $O_3 = 2000\mu$s, $O_4 = 0\mu$s, $O_5 = 2000\mu$s, and $O_6 = 4700\mu$s as the respective offsets. Since every period in $\overline{M}$ is an integer multiple of $T_1 = 2000\mu$s, by aligning other message streams with either the odd or the even message instances of $\overline{M}_1$, the interference among message

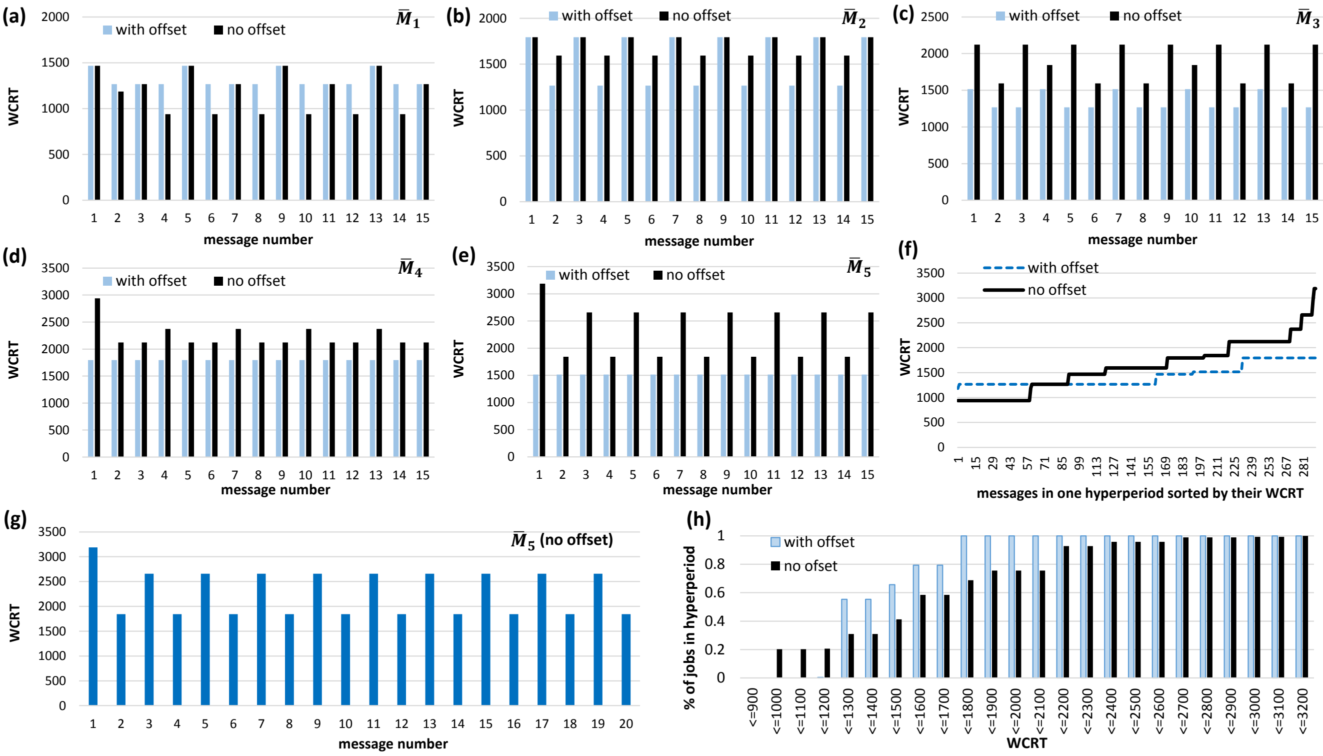


Fig. 3. WCRT of jobs for the benchmark message set in Table I: Experimental results for task sets with zero, small, and large jitter. **(a, b, c, d, e)** WCRT of the first 15 jobs of messages $\overline{M}_1$ to $\overline{M}_5$, **(f)** WCRT of each individual message in one hyperperiod sorted in a non-decreasing order, **(g)** WCRT of all jobs of message $\overline{M}_5$ in a synchronous-release scenario, and **(h)** cumulative distribution function of message instances for each WCRT scale from 900 to 3200.

streams can be reduced (see Fig. 3). In particular, using the chosen offsets, the WCRT of message streams $\overline{M}_i, i \geq 2$ is reduced efficiently while only a few instances of $\overline{M}_1$ experience an increase in their WCRT. Note that assigning large offsets to message streams may result in carry-in workload for the next hyperperiod which, in turn, increases the length of the observation window that must be analyzed [17, 23]. We are hence interested in offset assignments that do not push extra workload to the next hyperperiod.

Fig. 3(f) shows the response time of all individual messages over a hyperperiod. To make the trends clear and visible, we sorted the WCRTs in ascending order. As it can be seen, our offset assignment resulted in a more balanced WCRT distribution over the hyperperiod. In particular, it reduced the tail of the distribution of WCRTs (see Fig. 3(h)).

Discussion. As shown before, a fine-grained response-time analysis, which takes into consideration the WCRT of individual messages rather than a message stream, allows verifying more general forms of timing constraints, e.g., weakly-hard constraints. It also provides directives on how to find offsets such that a more balanced distribution of WCRT is attained for a message stream. In the following, we discuss some open problems that focus on a combination of practical constraints in designing robust and reliable CAN-based systems.

CAN controllers usually have a buffer to store pending messages, including messages that are released but not yet send and messages that have been transmitted erroneously and must be retransmitted. The WCRT of a message instance is an indicator of the lifetime of that individual message in the CAN controller buffer of the node it belongs to. Thus, taking into account the arrival time and WCRT of message instances, it is possible to derive the buffer size of the CAN controller on each node. This leads to the first interesting open problem.

Open Problem 1. *Given a schedulable message set $\mathcal{M}$ and the number of retransmissions f affecting each message, derive an upper bound on the required buffer size of each CAN node.*

Open Problem 2. *Given a fixed buffer size B for each CAN controller, a message set $\mathcal{M}$, and number of retransmissions f, derive the WCRT of each message while accounting for messages dropped due to buffer overflows.*

Open Problem 2 can be extended to systems with weakly-hard constraints in order to evaluate the schedulability of a message set using an (m, k) constraint.

Another direction is to devise a more accurate, ideally exact, response-time analysis for message sets with retransmission, where instead of an upper bound on the WCRT, an exact WCRT of each message is obtained. In our current approach, each erroneous transmission is modeled as a non-deterministic event (erroneous message) which can happen before any message and its worst-case transmission time is as large as the largest message in $\mathcal{M}$. This approach, however, contains two main sources of pessimism: **(i)** it pessimistically increases the response time of (high-priority) messages with short transmission time since the transmission time of the erroneous message is set to be as large as C^{max}, which might be determined by a low-priority message, and **(ii)** it includes scenarios where more than one retransmission of a lower-priority message can possibly happen before a pending higher-priority message is transmitted. The latter situation happens because we assign the highest priority to the erroneous messages. However, in reality, a retransmission can only happen in the order of priority of pending messages. Namely, a higher-priority message can be blocked by at most one retransmission of a lower-priority message. To remove this pessimism, failed transmissions must be incorporated into the generation of the schedule-abstraction graph such that a failed transmission inherits the properties of the last message dispatched on a path. This requires defining new rules for *expanding* and *merging* paths in the graph and hence requires its own proof of correctness afterwards, which we leave to future work.

V. Conclusion

The paper provides a sufficient schedulability analysis for a set of messages transferred over a CAN bus in the presence of message retransmission due to transient errors caused by electromagnetic interference. The analysis derives the worst-case response time (WCRT) of each individual message as a function of the maximum number of bit-flips (errors) that can happen within the given window of time. The paper explains how to use a state-of-the-art exact schedulability analysis of a set of non-preemptive jobs upon a uniprocessor platform in order to obtain an upper-bound on the WCRT of the messages in the presence of transient errors. Since the analysis is message-specific, it opens up an array of opportunities as discussed in §IV. In particular, it would be interesting to derive an exact retransmission-aware response-time analysis by directly modifying the way the schedule-abstraction graph is explored.

Acknowledgements.

The first author is supported by a post-doctoral fellowship awarded by the Alexander von Humboldt Foundation.

References

[1] C. Johnson, *Failure in Safety Critical Systems: A Handbook of Incident and Accident Reporting Glasgow University Press*. Glasgow, 2003.

[2] J. Noto, G. Fenical, and C. Tong, "Automotive EMI shielding–controlling automotive electronic emissions and susceptibility with proper EMI suppression methods," *Laird Technologies*, 2010.

[3] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol*. Springer New York, 2012.

[4] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[5] S. M. T. de Carvalho and G. L. Campos, "Worst case response time approach evaluation for computing CAN messages response time in an automotive network," in *Brazilian Power Electronics Conference*, 2017, pp. 1–6.

[6] I. Broster, A. Burns, and G. RodrÍguez-Navas, "Timing analysis of real-time communication under electromagnetic interference," *Real-Time Systems*, vol. 30, no. 1-2, pp. 55–81, 2005.

[7] S. Punnekkat, H. Hansson, and C. Norstrom, "Response time analysis under errors for CAN," in *IEEE Real-Time Technology and Applications Symposium (RTAS)*, 2000, pp. 258–265.

[8] P. M. Yomsi, D. Bertrand, N. Navet, and R. I. Davis, "Controller area network (CAN): Response time analysis with offsets," in *IEEE International Workshop on Factory Communication Systems (WFCS)*, 2012, pp. 43–52.

[9] Y. Dong, H. Ma, H. Xu, and K. Wen, "Response time analysis of mixed scheduling over CAN," in *International Conference on Future Computer and Communication (ICFCC)*, 2010, pp. 494–498.

[10] L. Pinho, F. Vasques, and E. Tovar, "Integrating inaccessibility in response time analysis of CAN networks," in *IEEE International Workshop on Factory Communication Systems (WFCS)*, 2000, pp. 77–84.

[11] M. Joseph, "Finding response times in a real-time system," *The Computer Journal*, vol. 29, no. 5, pp. 390–395, 1986.

[12] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[13] Guillem Bernat Nicolau, "Specification and analysis of weakly hard real-time systems," PhD thesis, Universitat de les Illes Balears, Spain, 1998.

[14] A. Gujarati, M. Nasri, and B. B. Brandenburg, "Quantifying the resiliency of fail-operational real-time networked control systems," in *Euromicro Conference on Real-Time Systems (ECRTS)*, 2018.

[15] J. Diaz, D. Garcia, Kanghee Kim, Chang-Gun Lee, L. Lo Bello, J. Lopez, Sang Lyul Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *IEEE Real-Time Systems Symposium (RTSS)*, 2002, pp. 289–300.

[16] Y. Lu, T. Nolte, J. Kraft, and C. Norstrom, "A statistical approach to response-time analysis of complex embedded real-time systems," in *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2010, pp. 153–160.

[17] M. Nasri and B. B. Brandenburg, "An exact and sustainable analysis of non-preemptive scheduling," in *IEEE Real-Time Systems Symposium (RTSS)*, 2017, pp. 1–12.

[18] T. P. Baker and M. Cirinei, "Brute-force determination of multiprocessor schedulability for sets of sporadic hard-deadline tasks," in *International Conference on Principles of Distributed Systems (OPODIS)*. Springer, 2007, pp. 62–75.

[19] A. Burmyakov, E. Bini, and E. Tovar, "An exact schedulability test for global FP using state space pruning," in *International Conference on Real-Time Networks and Systems (RTNS)*, 2015.

[20] N. Guan, Z. Gu, Q. Deng, S. Gao, and G. Yu, "Exact schedulability analysis for static-priority global multiprocessor scheduling using model-checking," in *Software Technologies for Embedded and Ubiquitous Systems (SEUS)*, 2007, pp. 263–272.

[21] Y. Sun and G. Lipari, "A pre-order relation for exact schedulability test of sporadic tasks on multiprocessor global fixed-priority scheduling," *Real-Time Systems*, vol. 52, no. 3, pp. 323–355, 2016.

[22] K. W. Tindell, A. Burns, and A. J. Wellings, "An extendible approach for analyzing fixed priority hard real-time tasks," *Real-Time Systems*, vol. 6, no. 2, pp. 133–151, 1994.

[23] J. Goossens, E. Grolleau, and L. Cucu-Grosjean, "Periodicity of real-time schedules for dependent periodic tasks on identical multiprocessor platforms," *Real-Time Systems*, vol. 52, no. 6, pp. 808–832, 2016.