Cyber-Physical Systems in Production Engineering

Designing safe, predictable, and high performance embedded platforms for next generation Cyber-Physical Systems.

The Cyber-Physical Systems in Production Engineering chair was founded recently in September 2018. Research activities of the Cyber-Physical Systems in Production Engineering Group in 2018 focused on two main topics: predictable high performance computing with heterogeneous SoC multicore platforms, and design of “real-time software revival” techniques to guarantee the physical safety of a Cyber-Physical System (CPS) in the presence of Cyber attacks. Other research activities are focusing on the secure and safe integration of machine learning algorithms with digital controllers for CPS and development of flexible resource management policies for a broad range of CPS systems.

As a new chair, the first few months were dedicated to setting up office spaces, laboratory, and start recruiting scientific researchers. However, this did not slow down the prompt start of our research, teaching, and service activities. In 2018, we have set up one master course and one PhD seminar regarding Cyber-Physical Systems. Prof. Caccamo is in the editorial board of IEEE Transactions on Computers. Members of the chair were involved in the program committee of several international academic conferences in automation and CPS, including ICCPS 2019, RTAS 2019, and DATE 2019.

On the Predictability of Heterogeneous SoC Multicore Platforms

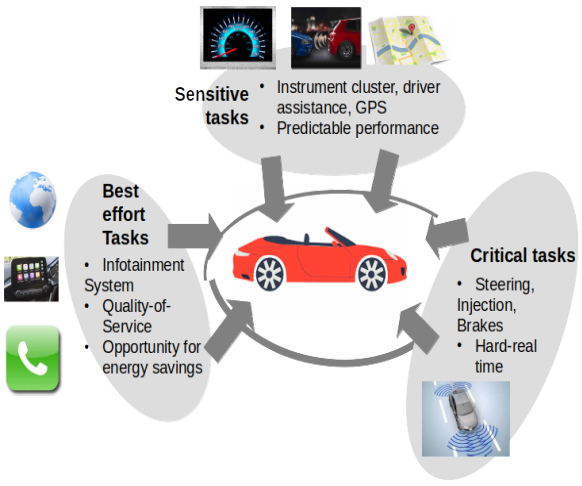


Fig. Example of a Cyber-Physical System (CPS) with conflicting requirements.

Modern Cyber-Physical Systems (CPS) are composed of several sensors, actuators, and microcontrollers or processors. Usually, they are designed to control and interact with a specific environment, integrating physical dynamics with software and networks. Applications include system automation, Internet of Things (IoT), smart buildings, smart manufacturing, smart cities, agriculture, robotics, and autonomous vehicles.

For many emerging CPS domains such as autonomous vehicles, computing platforms have to deal with diverse, often conflicting requirements, as demonstrated in the Figure above. Some tasks, such as autonomous steering, fuel injection, and brakes control, are mission critical and pose hard real-time requirements to the system. Conversely, multimedia infotainment systems demand high-performance and are programmed to tolerate large variations in the Quality-of-Service (QoS) provided by best-effort operating systems such as Linux. In addition, a third rapidly growing class of time sensitive applications require both predictability and performance at the same time. Thus, modeling, designing, and implementing resource-efficient, predictable and safe CPS is not a straightforward task.

In this project, we study how it is possible to leverage latest-generation partially reconfigurable System-on-Chip (SoC) embedded platforms for a system design that combines high-performance and strict real-time requirements. In our approach, we define multiple *criticality domains* to be intended as sub-shells of the computing system. Each criticality domain is designed with a different trade-off between high-performance and strict temporal determinism. For instance, a high-performance domain may run a general-purpose OS with a complex I/O infrastructure. Conversely, a high-criticality domain is comprised of a Real-Time Operating System (RTOS) supporting time-sensitive applications. Figure below demonstrates the proposed hardware architecture to support mixed-criticality applications using the Xilinx ZCU102 Ultrascale+ platform.

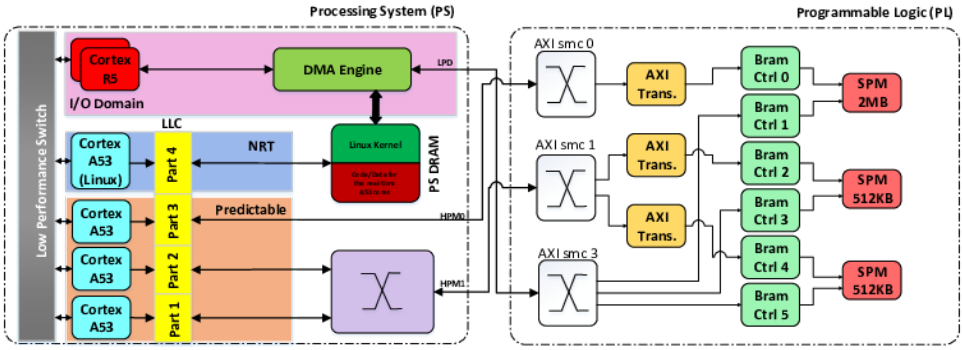


Fig. Proposed hardware design using the Xilinx ZCU102 Ultrascale+ platform.

In our design, we assign one high-performance core (A53) to Linux to take care of best-effort tasks and reserve the other three cores to execute critical/sensitive tasks on top of an RTOS. Jailhouse hypervisor provides isolation among the cores. Several low level resource management techniques, such as page coloring and code relocation, are implemented within Jailhouse to enhance the temporal predictability of running applications and avoid contention at several shared hardware components. We use a

three-phase task execution model, in which a DMA engine transfers data from the main memory (DRAM) to the scratchpad (SPM) before a task is executed by the RTOS. The SPM avoids memory contention in the shared DRAM.

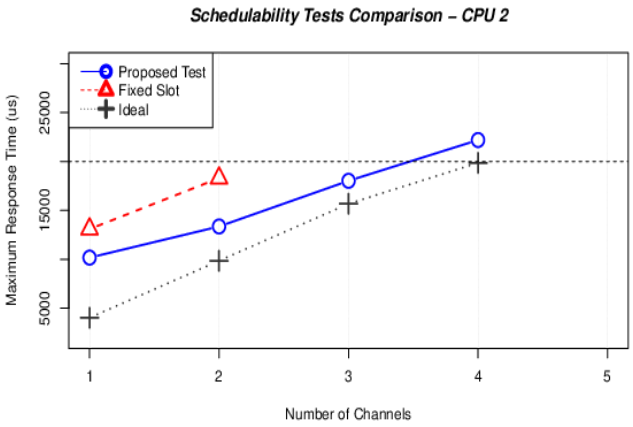


Fig. Comparison of the proposed schedulability test.

Finally, we propose a new schedulability test for event-based real-time tasks using variable TDMA slot sizes for the DMA transfers. We compare the proposed test against an ideal test and existing related work on schedulability tests. The proposed test together with the hardware/software architecture is able to deliver better CPU utilization, while meeting real-time guarantees of time-critical CPS applications.

Preserving Physical Safety under Cyber Attacks

Some of the recent attacks on cyber-physical systems (CPS) are focused on causing physical damage to the plant. Such intruders make their way into the system using cyber exploits but then initiate actions that can destabilize and even damage the underlying (physical) systems. The trend towards the adoption of remote monitoring and control (often via the Internet) of modern cyber-physical systems only further aggravates the safety-related security problems in current and next generation CPS. Many techniques to enhance system security focus on preventing the software platform from being compromised at all times or detecting the malicious behavior as soon as possible and taking recovery actions. Unfortunately, there are often unforeseen vulnerabilities that enable intruders to bypass the security mechanisms and gain administrative access to the controllers.

In this project, we leverage physical properties of the controlled plant (e.g., like inertia in mechanical systems) to guarantee the safety of a controlled plant despite the fact that the control software might be subject to external cyber attacks; in fact, an adversary cannot destabilize or compromise a plant (even with complete control over the software) instantaneously. It often takes finite, even considerable time to do that. Hence, we aim to develop analytical methods that can formally guarantee the safety of the physical

plant even when the controller unit's software has been partially (or even entirely) compromised.

A key idea is to carry out consecutive evaluations of physical safety conditions inside secure execution intervals. Those intervals should be separated in time such that an attacker with full control will not have enough time to destabilize or crash the physical plant in between two consecutive secure intervals. We refer to these intervals by Secure Execution Interval (SEI). In this research work, the time between consecutive SEIs is dynamically calculated in real time, based on the mathematical model of the physical plant and its current state. The key insight for providing formal safety guarantees is to make sure that each SEI takes place before an attacker can cause any physical damage.

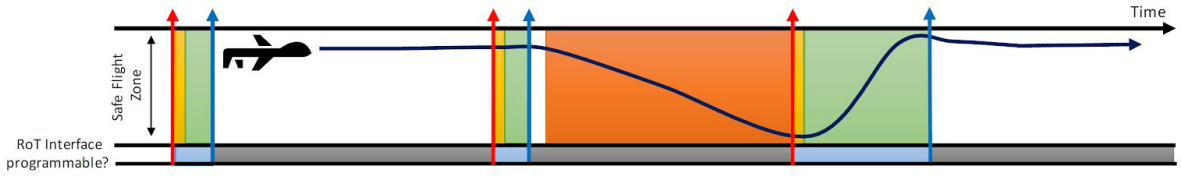
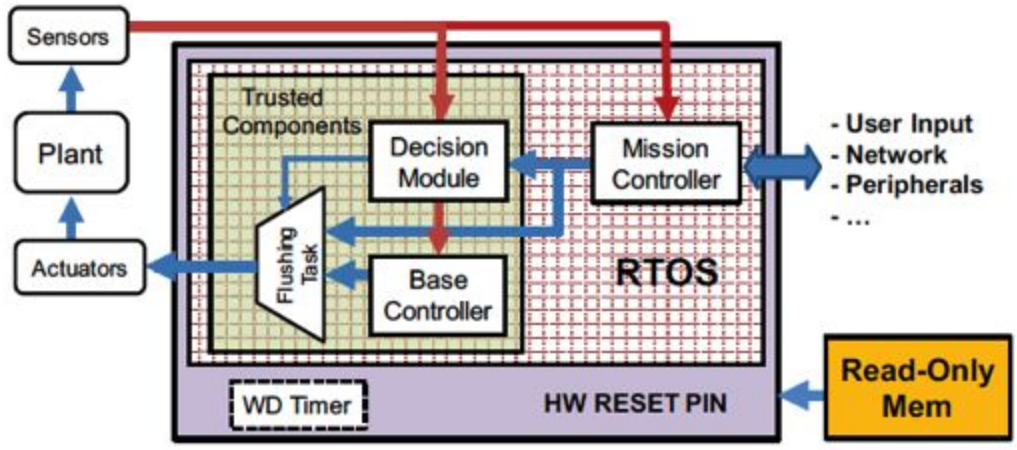


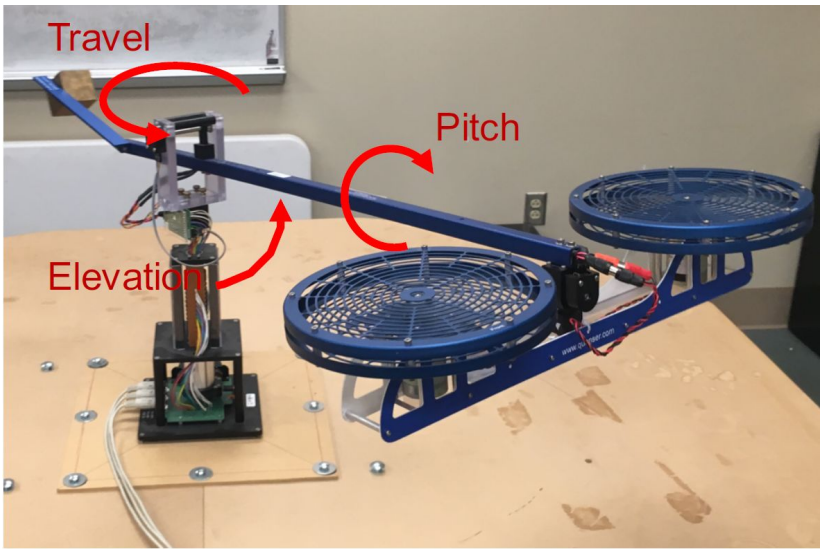
Fig. 1: An example sequence of events for the restart-based implementation of the SEI. White: mission controller is in charge and platform is not compromised. Yellow: system is undergoing a restart. Green: SEI is active, SC and `find_safety_window` are running in parallel. Orange: adversary is in charge. Blue: RoT accepts new restart time. Gray: RoT does not accept new restart time. Red arrow: RoT triggers a restart. Blue arrow: SEI ends, the next restart time is scheduled in RoT, and the mission controller starts.

We utilize two different approaches to create a trusted execution environment for SEIs where the integrity of the executed code can be trusted. Those approaches are based on (i) restart-based implementation which utilizes full system restarts with software reloads and (ii) TEE-based implementation which utilizes Trusted Execution Environment (TEE) such as ARM TrustZone or Intel Trusted Execution Technology (TXT) that are available in some hardware platforms.



The software architecture implements two digital controllers: 1) the base controller (called also safety controller), and 2) the mission controller. Safety (base) controller

guarantees safety and it is periodically executed within each SEI, but mission controller is required to run the controlled system and make progress toward a certain mission goal.



In practice, the proposed software framework implements a good trade-off between safety and performance, i.e. providing guaranteed protection and good performance at the same time. Some experimental results on a 3 degree-of-freedom helicopter and a simulated warehouse temperature management unit show that proposed techniques are robust against multiple emulated attacks – essentially the attackers are not able to compromise the safety of the CPS. Under extreme attack, e.g. abnormal full control input with maximal capacity or sudden shut down of the controller, the system did not make any progress towards its designated goal, but still remained safe which is the primary goal in this situation. Meanwhile, when attacks did not exist, the high availability of the mission controller ensured significant progress toward the control goal. In our experiments, the average availability of the mission controller for the 3 degree-of-freedom helicopter reached 85.1% while for the simulated warehouse temperature management unit, the average availability was 99.1%.

Cyber-Physical Systems in Production Engineering

Prof. Dr. Marco Caccamo

Contact
www.mw.tum.de/cps
mcaccamo@tum.de
Tel: +49.89.289.55170

Management
Prof. Dr. Marco Caccamo, Director

Administrative Staff
Anke Harisch, Secretary

Research Scientists
Giovani Gracioli, Dr.
Mirco Theile, M.Sc.
Bingzhuo Zhong, M.Sc.

Research Focus
Safety-critical cyber-physical systems
Real-time systems
Scheduling and schedulability analysis
Secure and safe integration of machine learning with CPS

Competence

System-level programming
Embedded system software design
Hardware and software integration on FPGAs
Real-time operating systems
Real-time reachability analysis

Infrastructure
3 DOF Helicopter
Embedded and FPGAs multicore development platforms

Lectures
Advanced Seminar on Safe Cyber-Physical Systems
PhD-Seminar on Real-Time Cyber-Physical Systems

Selected Publications 2018
F. Abdi, C.-Y. Chen, M. Hasan, S. Liu, S. Mohan and M. Caccamo, "Preserving Physical Safety Under Cyber Attacks", IEEE Internet of Things Journal, Accepted, 2018.

G. Gracioli, R. Tabish, R. Mirosanlou, R. Mancuso, R. Pellizzoni, and M. Caccamo, “A virtualized scratchpad-based architecture for real-time event-triggered applications,” tech. rep., Lehrstuhl für Cyber-Physical Systems in Production Engineering, 2019.