Chair of Cyber-Physical Systems in Production Engineering: Annual Report 2022

March 23, 2023

1 Introduction

"Designing smart, predictable, and high-performance embedded solutions for next generation Cyber-Physical Systems."

Modern Cyber-Physical Systems (CPS) are the next generation of engineered systems in which computing, communication, and control technologies are tightly integrated. Applications include system automation, Internet of Things (IoT), smart buildings, smart manufacturing, smart cities, digital agriculture, robotics, and autonomous vehicles. The chair of Cyber-Physical Systems in Production Engineering was founded in September 2018.

In 2022, research activities of the Chair focused on following topics: a) develop new reinforcement learning architectures for CPS, b) design and implement novel resource management policies for embedded real-time systems running on high-performance heterogeneous platforms, c) design architectures for sandboxing controllers in CPS, and d) develop a 6D pose recognition framework for robotic manipulation. Other research activities are also focusing on the secure and safe integration of machine learning algorithms with digital controllers for CPS.

Members of the chair were involved in the peer review process of several international conferences/journals in real-time embedded systems and CPS, including RTSS 2022, RTAS 2023, ECRTS 2022, EMSOFT 2022, IROS 2022, ICRA 2023, ICCPS 2022, CDC 2022, ECC2022, WCNC 2022, ICC 2022, ISCC 2022, GLOBECOM 2022, as well as IEEE Transactions on Computers, IEEE Transactions on Automatic Control, IEEE Transactions on Aerospace and Electronic Systems, IEEE Transactions on Control Systems Technology, ACM Transactions on Cyber-Physical Systems, Real-Time Systems, Journal of System Architectures, IEEE Embedded Systems Letters, IEEE Wireless Communications Letters, IEEE Control Systems Letters, IEEE Sensors Letters, IEEE Systems Journal, IEEE Access, International Journal of Electrical and Computer Engineering, Advances in Space Research.

2 Cloud-Edge Distributed Training Architecture for Reinforcement Learning

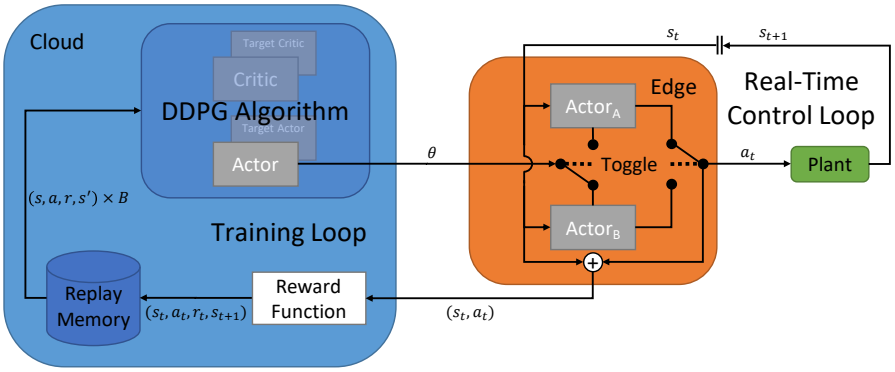


Figure 1: Distributed cloud-edge training architecture displaying the cloud, edge, and plant with the cloud and edge interacting through the training loop, and the edge and plant interacting through the real-time control loop; the loops are disconnected by double-buffering the actor on the edge.

Deep reinforcement learning (DRL) is a promising class of learning algorithms to tackle complex optimization problems for planning and control of Cyber-Physical Systems through interactions with the environment alone. In the robotic control domain, DRL enables robots to master complicated tasks with impressive performances, e.g., locomotion, grasping, and manipulation. However, due to the high demand for training data for DRL, direct training on physical systems presents many challenges. Collecting training data in the real world is expensive with respect to time and labor. Human supervision is usually needed to reset the system and monitor its hardware maintenance and safety status.

To address these real-world training challenges, some work improved DRL algorithms to reduce the learning sensitivities to hyperparameter settings, making training on physical systems more stable. Approaches of off-policy training with replay memory and model-based reinforcement learning are proposed to increase real-world sampling efficiency. Moreover, training using demonstrations and scripted policies can further ease real-world exploration.

In contrast, modern simulations can simulate complex systems and various environments. Recent sim-to-real approaches pretrain agents in simulations and then directly deploy the learned policies for real-world applications without further training. Simulation-based training boosts sampling efficiency significantly since simulations run faster than physical systems and can be further improved via parallel training. Additionally, the system can reset automatically without human intervention. Simulation training also does not require hardware maintenance and safety measures, drastically reducing the need for human supervision. Unfortunately, direct deployment of simulation-trained agents often fails in real-world applications due to the dynamic divergence between the real world and the simulated environment, the so-called *reality gap*. The *reality gap* arises mainly from system under-modeling, where the complex dynamics, e.g., the contact and friction effects, are difficult to measure and model. Moreover, computation and communication delays and environmental noise introduce extra modeling errors.

To bridge the *reality gap*, system identification approaches aim to create more accurate simulations, mitigating modeling errors. Techniques like domain randomization and domain adaptation can improve the agents' robustness in simulations, resulting in a successful direct real-world de-

ployment. Nevertheless, in many scenarios, the domain randomization strategies might not be feasible since there might not exist a single policy that can solve all problems within the domain.

Pretraining in simulation to learn a sub-optimal policy and then continuing the training in the real world can take advantage of the efficient simulation training and the real dynamics. With this approach, all the advantages of simulation learning can be adopted, and the *reality gap* can be closed while training in the real world. We follow the continuous sim-to-real training paradigm in this work, as we believe it is the most promising approach for real-world DRL.

However, the continuous sim-to-real training paradigm suffers from two main problems. First, DRL requires high-performance computation and benefits from dedicated devices such as GPUs or TPUs for its training loop. The plant cannot have a high-performance device onboard in many real-time control systems, e.g., unmanned aerial vehicles or other mobile robots, due to power, weight, and space constraints. It cannot be controlled directly from a remote high-performance device, as perfect, loss-less, and low-delay communication cannot be guaranteed. The second problem is that the transfer of a pretrained agent to the real physical system is non-trivial. The dynamics can change abruptly, making the value estimates of the DRL agent inaccurate, leading to deteriorating performance.

In [6], we addressed the real-world training problem by introducing a novel distributed cloud-edge architecture, as shown in Figure 1. The real-time control loop on the edge is decoupled from the computationally intensive training loop on the cloud. The agent on the edge collects experiences in real-time, sending them to the trainer on the cloud, which periodically updates the edge with the optimized parameters. The agent is double-buffered such that the real-time inference loop is not interrupted by the policy updates. We addressed the sim-to-real transfer problem by delaying the neural network training at the beginning of the real-world interactions. Specifically, we start the policy optimization later than the value estimate training to avoid policy deterioration based on unstable value estimates.

We evaluate our approach on a physical inverted-pendulum system controlled by a DRL agent deployed on a Raspberry Pi 4B. The training loop is offloaded to a high-performance workstation. The agents are pretrained in intentionally under-modeled simulations to induce different levels of the *reality gap* to analyze the sim-to-real transfer. The real-world experiments show that our architecture can adapt the pretrained DRL agents to unseen dynamics consistently and efficiently. We further analyze the impact of the neural network optimization delays, highlighting their necessity. Additionally, we investigate the relevance of the edge update frequency on training performance to show that the architecture can work in constrained bandwidth settings, albeit with an impact on training time.

3 Predictable and high-performance resource management of CPS on heterogeneous platforms

The widespread use of artificial-intelligence (AI) algorithms in many Cyber-Physical Systems (CPS) such as autonomous cars, drones, and smart robots has driven the integration of specialized hardware accelerators (e.g., GPUs, FPGAs) on high-performance multiprocessor boards. Towards ensuring safety and real-time requirements, these heterogeneous multiprocessor systems-on-chips (MPSoC) pose unprecedented challenges. In fact, the implementation of complex CPS using these platforms generates increasing volumes of real-time (e.g., imaging) data flows causing the hardware memory hierarchy (the DRAM, the interconnect, and the cache hierarchy, especially the last level cache shared among multiple cores) to become a bottleneck and a source of temporal unpredictability. This phenomenon is further aggravated by the presence of accelerators (GPUs/FPGAs) that can independently access memory with high-bandwidth requests. Traditional techniques to allocate and optimize the execution of real-time tasks on safety critical CPS do not consider the heterogeneity of the computing elements and the complexity of MPSoCs' memory hierarchy. In addition, classical task models widely adopted in the real-time scheduling domain fail to capture the parallelization and heterogeneous computing needs of the new workloads.

At the integration level, developers and integrators are daunted by the task of finding the right trade-offs between selecting the appropriate scheduling policies, assigning real-time tasks to (heterogeneous) cores, selecting the size of cache partitions, and determining adequate bandwidth to allocate to each communicating resource. Our work tackles these challenges with techniques that could be rapidly adopted by the industry, and that aim to practically simplify the deployment of real-time workload on MPSoC without sacrificing neither predictability nor performance. On the platform side, we research, develop, and evaluate techniques to restore isolation and temporal predictability of safety critical software. We specifically target solutions (e.g., [48, 47, 17]) that prove to perform well "in practice", and we focus our integration effort at both Operating System (OS) and Hypervisor levels. Hypervisors (e.g., [2, 32, 40, 11, 10]) have become the de-facto industry standard to ensure isolation in certified partitioned safety-critical systems, but do not provide satisfactory isolation and predictability properties when contention at cache, interconnect, and DRAM level is considered. To facilitate the evaluation and the adoption of these techniques by the industry, in addition to publications (e.g., [55, 39, 14, 34]), we actively participate in the development of open source hypervisors (e.g., [2, 4]) to make the developed techniques not only readily available, but also supported by an active community [3]. On these topics, we also actively collaborate with highly skilled international teams [25, 44], which pursue objectives close to ours. We additionally actively develop open source real-time frameworks (e.g., [42]) to improve the interoperability and exchange of results among international research groups.

On the application and deployment side, our research considers optimization and scheduling techniques to hide the complexity of the configuration space from the integrators, while enforcing isolation (enacted via the above-mentioned low level solutions) and ensuring the real-time properties of the workload. Problems under analysis are: the optimization of real-time task allocation under simultaneous consideration of cache-size, core, and bandwidth constraints; scheduling of real-time resources with different quality-of-service guarantees to applications with multiple criticality levels (e.g., [36]); support the effective debugging of timing properties for cyber-physical systems to restore system schedulability (e.g., [37]); automated synthesis of distributed embedded controllers and its integration at tooling level (e.g., [38]).

In the remainder of this section, we present more details on our recent works [55, 14, 49]. These papers are representative of our research efforts in 2022 towards predictable and high-

performance resource management of CPS on heterogeneous platforms.

3.1 Microsecond-scale Memory Bandwidth Regulation [55]

In today's multiprocessor systems-on-a-chip (MPSoC), the shared memory subsystem is a known source of temporal interference. The problem causes logically independent cores to affect each other's performance, leading to pessimistic worst-case execution time (WCET) analysis. One of the most practical techniques to mitigate interference is memory regulation via throttling. Traditional regulation schemes (e.g., MemGuard [48]) rely on a combination of timer and performance counter interrupts to be delivered and processed on the same cores running real-time workload. Unfortunately, to prevent excessive overhead, regulation can only be enforced at a millisecond-scale granularity. This problem motivated for the following research question: Can memory bandwidth regulation be enforced following a *drastically different* approach? And, ideally, one that can achieve fine-grained regulation, acceptably low overheads, and widely customizable regulation policies capable of capturing multiple nuances in the performance of complex memory hierarchies.

In [55], we answer this question and present *MemPol*, a novel regulation mechanism from *outside the cores* that monitors performance counters for the application core's activity in main memory at a microsecond scale. The approach is fully transparent to the applications on the cores, and can be implemented using widely available on-chip debug facilities (CoreSight) on ARM processors.

For MemPol, we rely on an extra core or logic component (e.g. FPGA) in the SoC that implements the memory regulation. This regulator accesses the application cores' performance counters via memory mapped registers, and throttles the cores via the on-chip debug facilities, based on the ARM CoreSight specification. Thus, the regulation does not pose any performance overheads to a regulated core in the case that the core does not exceeds its memory bandwidth limits.

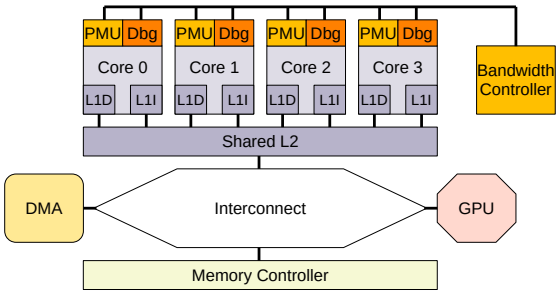


Figure 2: MemPol architecture on ZCU102.

As we evaluate our approach on the Xilinx Zynq UltraScale+ ZCU102 platform, we implement the memory regulator on the small Cortex-R5 real-time core with the goal to regulate the memory accesses of the large Cortex-A53 application cores, see Figure 2. The approach is feasible on a large number of SoCs that support a mix of large and small cores, but we also currently evaluate an implementation of a FPGA-based regulation, as the UltraScale+ SoC also provides an FPGA.

From the vantage point of *not* using an interrupt-based feedback mechanism (i.e. PMU interrupt) that is bound to a single performance counter, MemPol allows to use a more complex composition of metrics to enact load-aware regulation, such as accumulated cache refills and write-backs for exact read and write bandwidth regulation (MemGuard's regulation is based on

one PMU counter for cache refills). Also, it allows redistributing unused bandwidth between cores while keeping the overall memory bandwidth of all cores below a given threshold.

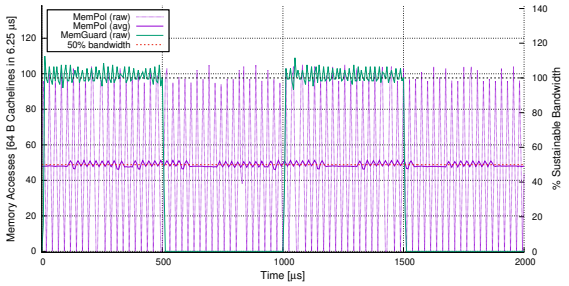


Figure 3: Comparison of MemPol (polling at 6.25 µs, sliding window size 50 µs) and MemGuard (regulation period 1 ms) on ZCU102 regulating a worst-case memory reader at 50% sustainable memory bandwidth. PMU counters are sampled every 6.25 µs by the regulator on the R5 core. Averages over 200 µs.

Figure 3 shows a comparison of the memory bandwidth regulation of MemPol and MemGuard. With MemPol, we can achieve a polling rate of 6.25µs for fine-grained memory regulation. The resulting regulation is comparable to hardware QoS regulation built into current ARM interconnects.

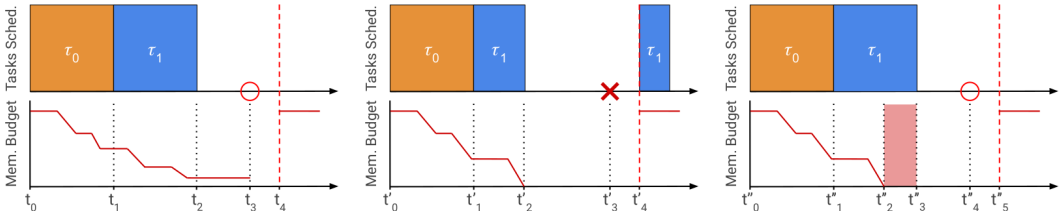
3.2 Computation and Memory Regulation Interplay in Multicore Real-Time Systems [14]

The real-time community has proposed many successful techniques to mitigate the impact of inter-core memory interference (*e.g.*, [48, 24]). Notably, performance counter (PMC) based techniques such as *Memguard* [48] have received significant attention due to their practicality. In fact, PMC-regulation techniques are used to establish *temporal isolation* by mitigating the problem of non-arbitrated memory bandwidth sharing between cores. In the embedded and real-time domain, these techniques are often implemented within a partitioning hypervisor (*e.g.*, Jailhouse [2]) when the consolidation of multiple RTOSs onto the same multicore system-on-a-chip (MPSoC) is required. At the same time, when consolidating complex applications with mixed-criticality requirements onto MPSoCs with rich OSs like Linux, CPU provisioning still remains a fundamental dimension. Here, server abstractions —*e.g.*, the Constant Bandwidth Server (CBS) [1]— are well known and widely used, with the `SCHED_DEADLINE` [21] policy being the most popular example.

Despite combining CBS-based CPU scheduling and PMC-regulation to achieve isolation in *both* time and memory domains being a logical choice, the effective integration proves to be challenging. The need to enact CBS scheduling at the operating system level and PMC-regulation at the hypervisor level results in a detrimental lack of coordination. This leaves the system incapable of handling *memory overload* conditions which we defined as:

*"A memory overload condition occurs when a high-critical task has enough computation budget to continue its execution (*i.e.*, eligible for scheduling), but is unable to proceed as the associated memory budget has been depleted (*i.e.*, throttled by the PMC-regulation)."*

The example depicted in Fig. 4 perfectly illustrates multiple scenarios that can arise during the system life-time. In this example, the considered system is composed of one low- and one high-criticality tasks (respectively τ_0 and τ_1) scheduled on one core using a CBS to absorb



(a) Standard PMC-regulation, normal case.

(b) Standard PMC-regulation, memory budget depletion.

(c) SHCReg regulation, with interconnect policy switch.

Figure 4: Example scenario of a PMC-regulated core, where an increased memory consumption causes τ_1 to miss its deadline.

execution variations. The common PMC-budget assigned is determined beforehand via profiling and the addition of a fixed *safety margin*, which is common practice in industrial applications. While in Fig. 4a, τ_1 is able to complete on time, in Fig. 4b suffers from a *memory overload* as it experiences extra blocking due to the lack of sufficient *memory budget* caused by an unexpected increased memory consumption from τ_0. Such an increase can be due to changing computational needs that require additional memory accesses (for example, consider the case of object detection or object tracking in an almost empty street vs. at a crowded intersection). We note that such an increase in memory consumption cannot be determined apriori without resorting to very pessimistic over-estimations.

In [14], we present a Software Hardware Co-design Regulator (or SHCReg). Our proposed mechanism to address the aforementioned challenge. SHCReg proposes to handle *memory overloads* such as the one presented above via a simple set of rules[1] that adequately instrument well-known and available tools. In the example illustrated by Figure 4, upon the detection of an overload, the critical task τ_1 is allowed to bypass the normal PMC-regulation mechanism, serving τ_1 critical memory access. At the same time, SHCReg programs a smart-interconnect to prioritize τ_1 memory traffic. At the system-wide level, individual priorities are assigned according to the criticality of the tasks running, allowing for a graceful degradation of quality of service.

In [14], we (1) present the precise set of rules enabling memory overloads handling, (2) outline the envisioned architecture, and (3) discuss the foreseen implementation challenges.

3.3 Memory Allocation for Low-power Real-time Microcontroller [49]

Low-power microcontrollers, by design, limit the energy-draining memory features. Typically, in many such devices, when a performance boost is needed for time-critical routines, only little core-coupled memories can be found. In [49], we report our experience in harnessing core-coupled and standard memory components to improve processor utilization on the example of a popular embedded microcontroller: the STMicroelectronics' `STM32F3 DISCOVERY` (`STM32F303VC`).

The `STM32F303VC` microcontroller is based on the Arm Cortex-M4 32-bit RISC core operating at a frequency of up to 72 MHz and incorporates three embedded memories: 256 KB of *Flash* memory, 40 KB of *SRAM*, and 8 KB of *CCM-RAM* (core-coupled memory RAM). The *CCM-RAM* and *SRAM* can be accessed in read/write at CPU clock speed (*i.e.,* zero wait-state). The *CCM-RAM* is typically used to speed up computation-intensive routines by executing their code at the maximum CPU clock frequency. This brings a significant decrease in the execution time compared to code executed from the *Flash* memory, where each instruction fetch has to be adjusted with additional wait-states with respect to the CPU clock frequency: zero up to 24 MHz, one from 24 to 48 MHz, and two wait-states from 48 to 72 MHz. Additionally, placing

[1]The exhaustive proposed set is presented in [14]

data in *SRAM* and instructions in *CCM-RAM* allows fetching instructions while data is being loaded or stored since they are accessed using separate buses (*i.e.,* Harvard configuration, see Figure 5). On the contrary, placing both data and instructions in *CCM-RAM* can lead to a bus contention as the *CCM-RAM* interface is shared between data and instructions. In this regard, the *Flash* memory with two interfaces, I-code and D-code, for reading instructions and data can avoid bus contention. However, as explained above, the *Flash* memory incurs wait-state penalties for higher CPU clock frequencies.

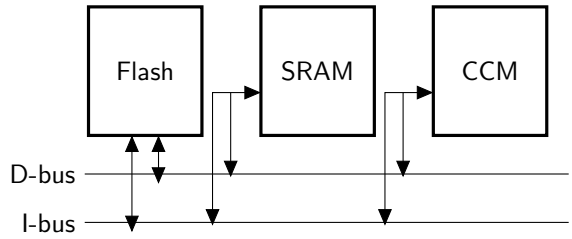


Figure 5: STM32F3 device bus matrix architecture simplified view.

To evaluate the impact of different memory allocation strategies on the execution time, we run a set of experiments based on the real-world embedded application benchmarks under the following memory allocation strategies: i) placing the code into *Flash*/*CCM-RAM* and ii) placing the read-only data into *Flash*/*SRAM*/*CCM-RAM*. We perform the experiments for each benchmark under the CPU clock frequencies of 24, 48, and 72 MHz.

Based on the execution time of each benchmark under different memory allocation strategies, we consider a design optimization problem to determine the optimal memory allocation of a set of real-time tasks running on the microcontroller concurrently, with the objective to improve the system schedulability. To solve the design optimization problem, we develop a 0-1 integer linear programming (ILP) model that can be solved by standard mathematical programming solvers efficiently. The evaluation results are shown in Figure 6. From the figure, we can see that the proposed optimal memory allocation method can achieve an average utilization reduction ratio of 25.3% when $n = 8, f = 72\,\text{MHz}$. Besides, the figure shows a trend that the utilization reduction ratio decreases with the increase of the task set size. The running time of our ILP solver ranges from 1.5 ms to 20 ms as n ranges from 8 to 64.

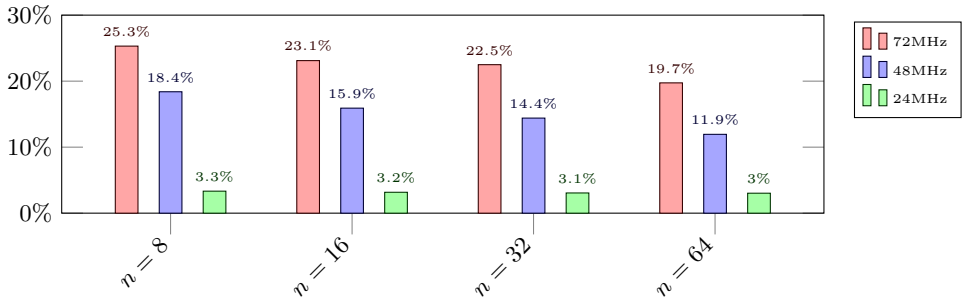


Figure 6: Average utilization reduction ratio with optimal memory allocation.

This work can be extended by using *DMA* to load read-only data and instructions from the *Flash* to the *SRAM* and the *CCM-RAM*, respectively. We also plan to consider a more fine-grained allocation with multiple code sections.

4 Designing Safe and Intelligent Safety-critical Cyber-Physical Systems via Correct-by-construction Synthesis and Safe-visor Architecture

Many modern Cyber-Physical Systems are safety-critical in the sense that failure in this systems (e.g., collision) may results in catastrophic consequences [26]. To enforce formal safety guarantee over these systems, one can deploy the notion of correct-by-construction synthesis techniques to construct controllers enforcing high-level logical properties, including those that can be formulated as Linear Temporal Logics (LTL) formulae [31]. In this year, our chair proposed several results on this direction with the notions of abstraction-based controller synthesis, abstraction-free controller synthesis, data-driven controller synthesis, and Safe-visor architecture for sandboxing unverified controllers in stochastic games.

4.1 Abstraction-based Controller Synthesis

Synthesizing correct-by-construction controllers over continuous-space stochastic systems are becoming more and more important in the past decades due to the increasing demand for providing safety guarantee for controlling safety-critical CPS operating in noisy environments. However, synthesizing these controllers are challenging when high-level logic properties, such as LTL properties, are desired to be enforced. Since closed-form solutions of synthesized policies for general continuous-space stochastic systems are not available, a promising approach is to approximate the original models by simpler ones with finite state sets. To cope with these demands, we propose in [52] a new abstraction and refinement methodology for designing controllers for non-cooperative stochastic games enforcing complex logical properties expressed by deterministic finite automata (DFA). To provide a less conservative safety guarantees compared with the current results, we deploy the notion of so-called (ϵ, δ)-approximate probabilistic relations [12] to quantify the similarity between the original stochastic games and their finite abstraction. Leveraging this relation, we first design controllers with respect to the desired logical properties over the finite abstraction. Then, by leveraging the (ϵ, δ)-approximate probabilistic relations, we provide a lower bound for the probability of satisfying the desired specifications by refining controllers synthesized over the finite abstraction to the original games. The overview of the proposed abstraction-based methodology is depicted in Figure 7.

To show the effectiveness of the proposed results, we applied them to a case study of a drone tracking a (virtual) ground vehicle while enforce the relative distances and velocities between the drone and the ground vehicle satisfying the following logical properties: within 2 seconds (*i.e.,* time horizon $H = 40$), (1) y should reach $[-0.45, 0.45]$ and then stay within $[-0.45, 0.45]$ for 3 time instants after it reaches $[-0.45, 0.45]$; (2) if it reaches $[-0.1, 0.1]$, it only needs to stay within $[-0.45, 0.45]$ for 1 time instant after it reaches $[-0.1, 0.1]$; (3) y is not allowed to leave $[-0.8, 0.8]$. This property can be modeled by a DFA as shown in Figure 8(Left).

To validate the safety guarantee associated with the synthesized controller, we simulated the system from the initial states $[-0.48, 0.45]$ with 10^5 different noise realizations. Accordingly, it is guaranteed that at least 98.75% of the system state trajectories will satisfy the desired property. In the simulation, as shown in Figure 8(Right), all the trajectories fulfill the desired safety properties, indicating the provided probabilistic guarantee is well respected. It is also worth mentioning that we only need 11.26 GB of memory when synthesizing the controller offline, and the controllers being deployed online is a look-up table with a size of 8.38MB. Meanwhile, with existing results, such as [16, 9], they require 6.768×10^{13} GB of memory for synthesizing a controller with reasonable safety guarantee, which is computationally expensive.

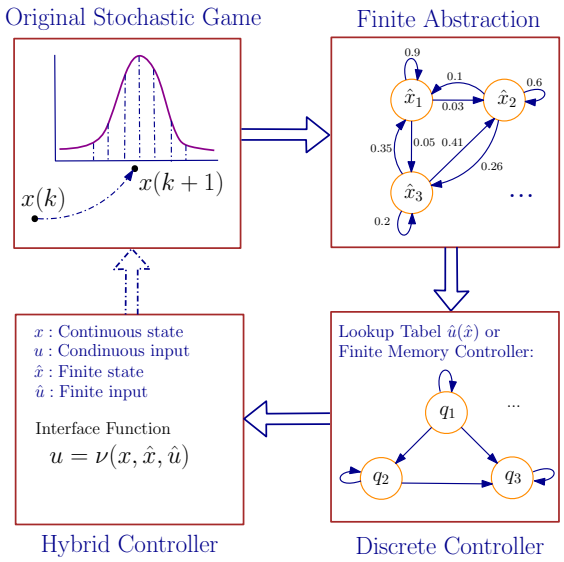


Figure 7: Overview of the abstraction-based correct-by-construction synthesis leveraging the notion of (ϵ, δ)-approximate probabilistic relations

4.2 Abstraction-free Controller Synthesis

While abstraction-based approaches is powerful in synthesizing controllers enforcing high-level logics properties over nonlinear (stochastic) systems, these approaches required building finite abstractions with finite state and input sets by discretizing the original continuous state and input sets. The need for constructing a finite abstraction results in one of the main challenges of these approaches, namely *curse of dimensionality*, since the number of discrete states and inputs grow exponentially with respect to the dimensions of state and input sets, respectively.

Focusing on linear systems affected by bounded disturbances, we proposed a new discretization-free approaches for synthesizing controllers enforcing ω-regular properties [43] (ω-regular properties is a superset of those specifications expressed as linear temporal logic (LTL) formulae, which are widely employed to specify properties for many applications, including [23, 46]). Concretely,

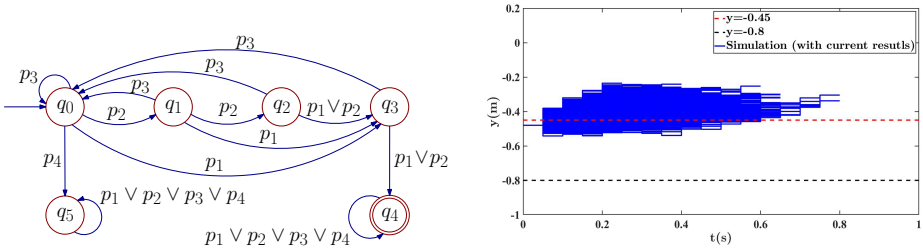


Figure 8: (**Left**: DFA for modeling ψ_3 with accepting state q_4, alphabet $\Pi = \{p_1, p_2, p_3, p_4\}$, and labeling function $L : Y \to \Pi$ with $L(y) = p_1$ when $y \in [-0.1, 0.1]$; $L(y) = p_2$ when $y \in [-0.45, -0.1) \cup (0.1, 0.45]$; $L(y) = p_3$ when $y \in [-0.8, -0.45) \cup (0.45, 0.8]$, and $L(y) = p_4$ when $y \in (-\infty, -0.8) \cup (0.8, +\infty)$. Transitions $q_5 = \tau(q_j, p_4)$, with $j \in \{1, 2, 3\}$, are omitted to keep the figure less crowded; **Right**: Simulations for the case study in Section 4.1.

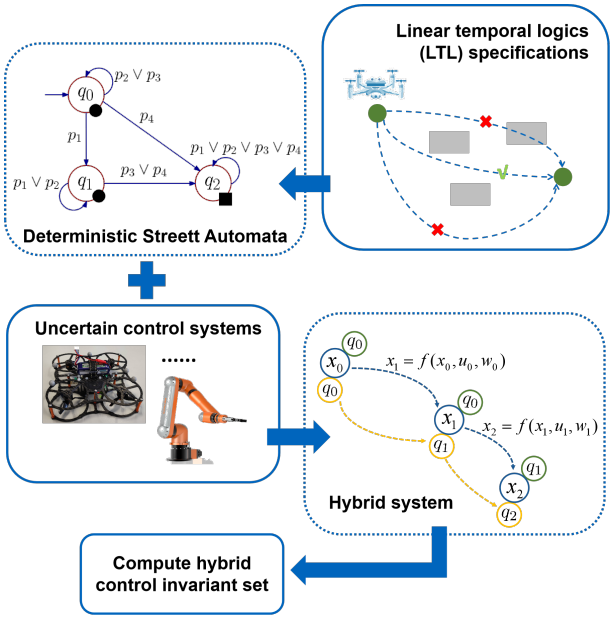


Figure 9: Overview for the abstraction-free approach.

we propose a set-based approach that leverages an iterative scheme to compute so-called hybrid controlled invariant (HCI) sets. An overview of applying this approach is depicted in Figure 9. Given a desired properties, we first convert it to a deterministic Streett automata (DSA) [41]. Then, we construct a product between the automata and the model of the systems, producing a hybrid system evolving in a hybrid state space. With this hybrid system, a controller enforcing the desired properties can be obtained by computing an HCI set over the hybrid system with the results we proposed in [53].

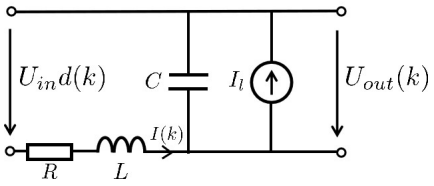


Figure 10: Buck converter modeled by an RLC circuit with a resistance $R = 100\Omega$, an inductance $L = 0.15H$, and a capacitance $C = 2mC$, with $U_{in} = 1000V$ being the input voltage, $d(k) \in [0, 1] \subset \mathbb{R}$ being the duty cycle, $I_l = 5A$ being the constant current load, $I(k)$ being the converter current, and $U_{out}(k)$ being the output voltage.

To validate the proposed algorithm, we applied it to a control problem of a buck converter as in Fig. 10. Over this systems, we are interested in enforcing an ω-regular property modeled by a DSA $\mathcal{A}$ as in Fig. 11. In English, this property requires that (i) $x(k)$ should not go into the region $X_2 := L^{-1}(p_3) \cup L^{-1}(p_5)$ after it has been to the region $X_1 := L^{-1}(p_1)$; (ii) $x(k)$ should not reach the region $X_f := L^{-1}(p_6)$. With the synthesized controller, we simulate the system for 30 time steps from 10 randomly selected initial states. The simulation results are depicted in Figure 12, indicating that the desired property is enforced.

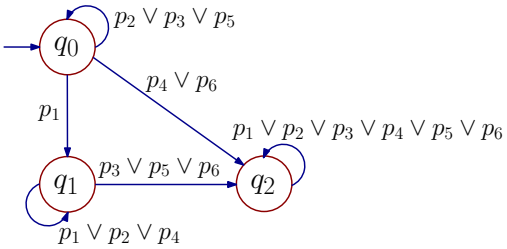


Figure 11: DSA $\mathcal{A}$ modeling ψ, with $E = \{q_2\}$, $F = \{q_0, q_1\}$, alphabet $\Pi = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, and labelling function $L : X \to \Pi$ with $L(x) = p_1$ when $x \in [221.001, 222] \times [-10, 10]$, $L(x) = p_2$ when $x \in [219, 221] \times [-10, 10]$, $L(x) = p_3$ when $x \in [218, 218.999] \times [-10, 10]$, $L(x) = p_4$ when $x \in (221, 221.001) \times [-10, 10]$, $L(x) = p_5$ when $x \in (218.999, 219) \times [-10, 10]$, and $L(x) = p_6$ when $x \in \mathbb{R}^2 \backslash ([218, 222] \times [-10, 10])$.

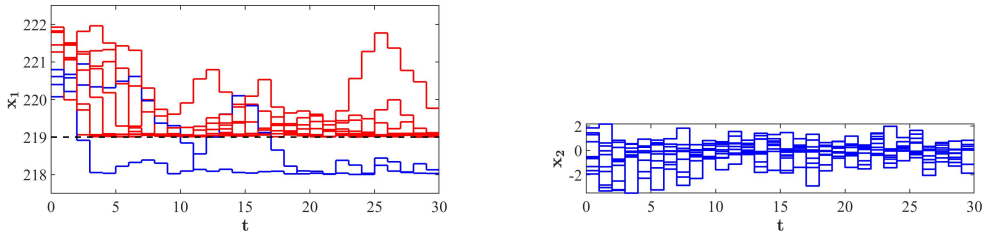


Figure 12: Simulations for the case study.

4.3 Data-driven Controller Synthesis

The abstraction-based and abstraction-free approaches introduced above require models of the systems being a prior. Nevertheless, in some cases, obtaining accurate models requires many efforts [15]. Moreover, even if the model is available, it may be too complex to be of any use. Such difficulties in applying model-based approaches motivate the researchers to enter the realm of data-driven control methods, with which the controller's parameters can be obtained directly based on data without system identification as an intermediate phase.

In this year, we have proposed a direct data-driven approach for constructing safety controllers enforcing invariance properties over unknown linear systems affected by unknown-but-bounded disturbances (i.e., systems are expected to stay within a safe set). Concretely, we first propose a notion of so-called γ-robust safety invariant (γ-RSI) sets. In particular, state-feedback controllers can be constructed on top of these sets, which can be deployed to enforce invariance properties modeled by (possibly unbounded) polyhedral safety sets. Then, we proposed an optimization algorithm with which the γ-RSI set can be computed by leveraging a single trajectory collected from the underlying systems with unknown dynamics. Here, the numbers of constraints and optimization variables grow linearly with respect to the numbers of hyperplanes defining the safety set and the cardinality of the finite data set. Additionally, we also investigate the relation between our data-driven approach and the condition of persistency of excitation [45], which is a crucial concept in most literature about direct data-driven approaches.

To demonstrate the proposed results, we consider a four dimensional linearized model of the inverted pendulum as in Figure 13. Here, we denote by $x(k) = [x_1; x_2; x_3; x_4]$ is the state of the inverted pendulum, with x_1 being position of the cart, x_2 being the velocity of the cart, x_3 being the angular position of the pendulum with respect to the upward vertical axis, and x_4 being the

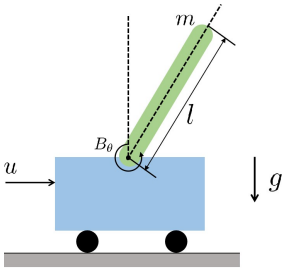


Figure 13: Inverted pendulum, where $m = 0.1314\,\text{kg}$ is the mass of the pendulum, $l = 0.68\,\text{m}$ is the length of the pendulum, $g = 9.81\,\text{m/s}$ is the gravitational constant, and $B_\theta = 0.06\,\text{Nm/s}$ is the damping coefficient of the connection between the cart (the blue part) and the pendulum (the green part).

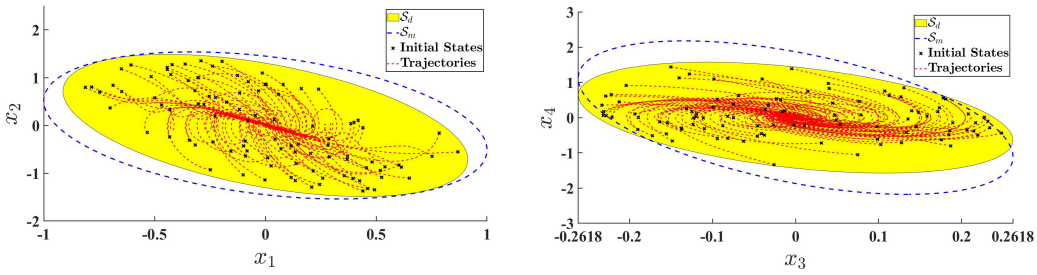


Figure 14: Projections of the data-driven γ-RSI set $\mathcal{S}_d$, the model-based γ-RSI set $\mathcal{S}_m$, initial states, and state trajectories on $x_1 - x_2$ plane (**Left**), and $x_3 - x_4$ plane (**Right**).

angular velocity of the pendulum. Given the acceleration limit ($[-5, 5]$m/s) for the cart, which is used as the control input of the system, it is desirable that 1) x_3 should stay $[-15°, 15°]$ with respect to the upward position; 2) x_1 should stay in the region $[-1, 1]$ m. To synthesized the controller, we collect an input-state trajectory of the system over a time horizon of 2.14 seconds, with sampling frequency of 50 Hz. The obtained data-based γ-RSI set and the simulation of the systems with the safety controller associated with the data-based γ-RSI set are depicted in Figure 14 (We are also depicting the model-based γ-RSI set here for comparison purposes). One can readily see that the desired safety property is respected over the underlying systems while the constraints for the control input is not violated.

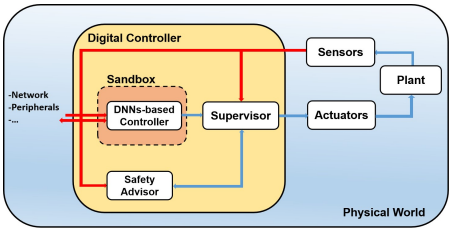


Figure 15: Safe-visor architecture for sandboxing DNNs-based controllers.

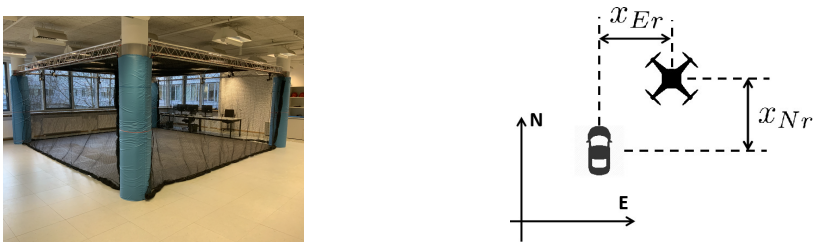


Figure 16: The physical test-bed of drone (**Left**); Coordinate for a quadrotor tracking a ground vehicle (**Right**).

4.4 Safe-visor architecture for stochastic games

Following the line of research in previous years, we also focus on designing Safe-visor architecture, as depicted in Figure 15, to provide safety guarantee for safety critical Cyber-Physical Systems whenever unverified controllers, such as deep-neural-networks-based (DNNs-based) controllers, are deployed in the control loop for complex missions instead of only ensuring overall safety of the systems.

Compared with the results in the last year, we propose new results in [50], with which one is able to cope with systems modeled as general discrete-time stochastic games (gDTSG). Here, gDTSG is a general framework for modeling various types of stochastic systems, including discrete time stochastic control systems [19], stochastic switched systems [20], and randomly switched control systems [30] affected by bounded disturbances. Leveraging this modeling framework, we are able to use the Safe-visor architecture to provide a safety guarantee being robust to those noises and disturbances that often affect the performance of DNNs in real-life applications.

Additionally, except for using Monte Carlo simulation to validate the provided safety guarantee, we also applied the current results on a physical test-bed (see Figure 16) of drone which is newly constructed at the end of 2021. Concretely, we considered a case study of a drone tracking a (virtual) ground vehicle while enforcing the relative distances and velocities between the drone and the (virtual) ground vehicle[2] (cf. Figure 16) satisfying the following logical properties: Within 1 min (time horizon $H = 600$), the following safety specifications are desired: (1) (ϕ_E, H): y_E should be within $[-0.5, 0.5]$ (m); (2) (ϕ_N, H): y_N should be within $[-0.5, 0.5]$ (m); additionally, if y_N reaches $[-0.3, 0.3]$ (m) at any time instant k, then y_N should be within $[-0.4, 0.4]$ (m) at time instant $k + 1$ and within $[-0.45, 0.45]$ (m) at time instant $k + 2$, instead of $[-0.5, 0.5]$ (m). Their associated DFAs $\mathcal{A}_E$ and $\mathcal{A}_N$ are shown in Figure 17.

In the experiment on the physical test-bed, we use a DNNs-based agent to control the quadrotor to track the vehicle, as shown in Figure 18, in which the agent is trained as a setpoints provider for low-level position controller. Here, the agent observes the current positions and the velocities of the quadrotor and the virtual ground vehicles and provide, accordingly, the position and velocity setpoints for the quadrotor. The agent deployed here is trained in simulation, in which the vehicle follows random trajectories, with DDPG algorithm [22]. The results of Monte Carlo simulation and the experiment on the physical test-bed are depicted in Figure 19. By applying the Safe-visor architecture, the desired safety guarantee can be respected, while some of the input provided by the DNNs-based controller can still be deployed. Here, the interested readers are also referred to `https://youtu.be/d-CjuKuhm_w` for a demonstration of more experiments on the physical test-bed in which Safe-visor architecture is used to sandbox unverified controllers.

[2]The absolute position and the velocity of the virtual ground vehicle are computed based on the dynamics of the vehicle.

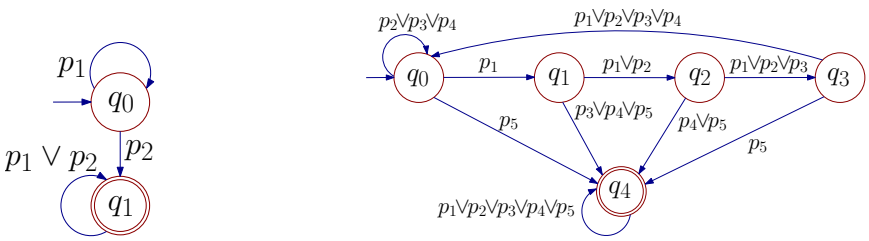


Figure 17: **Left:** DFA $\mathcal{A}_E$, with accepting state q_1, alphabet $\Pi = \{p_1, p_2\}$, and labeling function $L : Y \to \Pi$ with $L(y) = p_1$ when $y \in [-0.5, 0.5]$, and $L(y) = p_2$ when $y \in (-\infty, -0.5) \cup (0.5, +\infty)$. **Right:** DFA $\mathcal{A}_N$, with accepting state q_4, alphabet $\Pi = \{p_1, p_2, p_3, p_4, p_5\}$, and labeling function $L : Y \to \Pi$ with $L(y) = p_1$ when $y \in [-0.3, 0.3]$, $L(y) = p_2$ when $y \in [-0.4, -0.3) \cup (0.3, 0.4]$, $L(y) = p_3$ when $y \in [-0.45, -0.4) \cup (0.4, 0.45]$, $L(y) = p_4$ when $y \in [-0.5, -0.45) \cup (0.45, 0.5]$, and $L(y) = p_5$ when $y \in (-\infty, -0.5) \cup (0.5, +\infty)$.

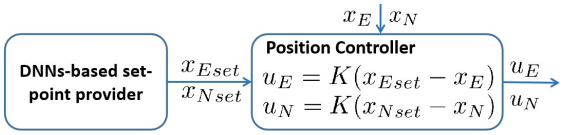


Figure 18: DNNs-based controller in real-world experiments.

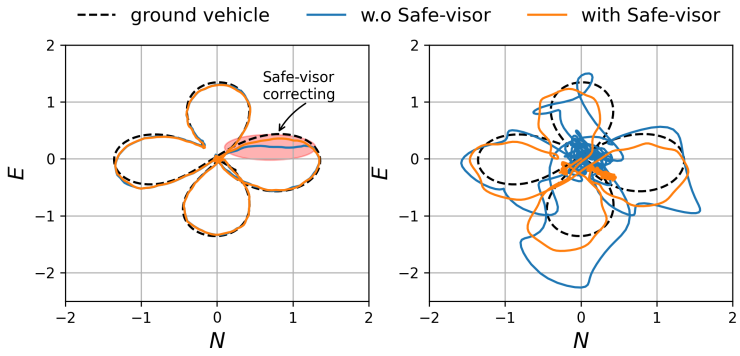


Figure 19: Trajectories of the quadrotor and the ground vehicle in simulation (**Left**), and real-world experiment(**Right**).

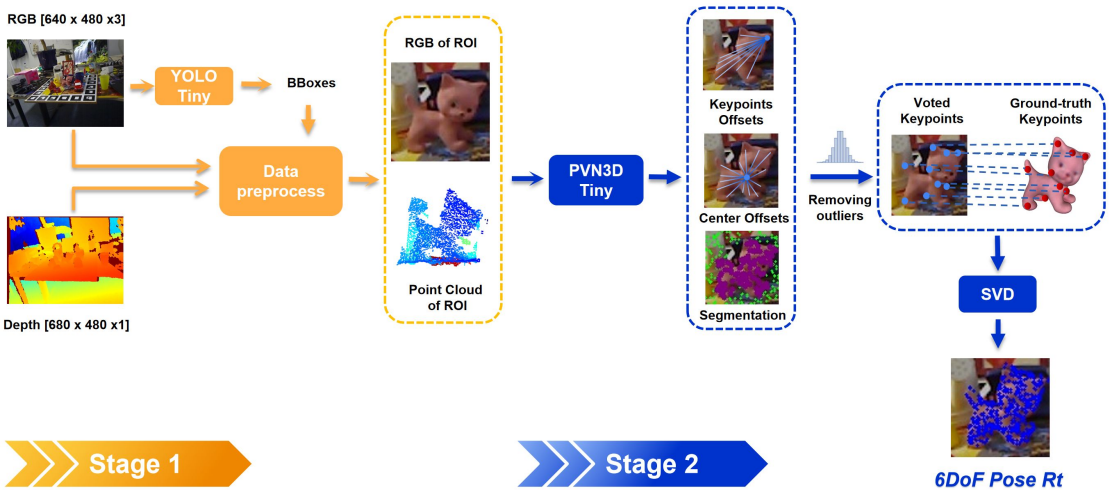


Figure 20: A two-stage pose estimation approach: object detection with YOLO-tiny followed by the 6D object pose estimation with PVN3D-tiny at the second stage.

5 6D Pose Estimation for Robotic Manipulation

We continued our effort to develop an integrated 6D pose estimation system for robotic grasping. This year we have completed a robotic testbed, created an integrated baseline 6D pose vision system and developed a Blender pipeline for the generation of synthetic data. Additionally, we have been and are still exploring other possibilities of leveraging simulations to approach new end-to-end vision paradigms and synthetic data generation.

The proposed pipeline can efficiently generate large amounts of photo-realistic RGBD images for the object of interest. In addition, a collection of domain randomization techniques is introduced to bridge the gap between real and synthetic data. Furthermore, we develop a real-time two-stage 6D pose estimation approach for time sensitive robotics applications (figure 20). This algorithm is based on PVN3D [13], which we integrated with YOLO [33] and optimized in order to achieve real-time capable inference, necessary for reactive robotic manipulation in later work.

With the proposed data generation pipeline, our pose estimation approach can be trained from scratch using only synthetic data without any pre-trained models. The resulting network shows competitive performance compared to state-of-the-art methods when evaluated on the LineMod dataset. We also demonstrate the proposed approach in a robotic experiment, grasping a household object from cluttered background under different lighting conditions (figure 21). In the study, we perform 750 grasping attempts on these five objects in three different challenging lighting conditions and achieve a remarkable success rate of 87.06% or 93% if collisions are neglected, which were not the focus of this research (figure 22).

These efforts and the robotic experiments have been summarized in a preprint [5], which is submitted for peer review. The software for this experiment, which includes a synthetic data generation pipeline with Blender, robot control software, and a computer vision pipeline, will be made available alongside the submission of the paper.

To demonstrate our current capabilities, we have created a video report which shows the robustness of our system to different lighting conditions, which is here available.

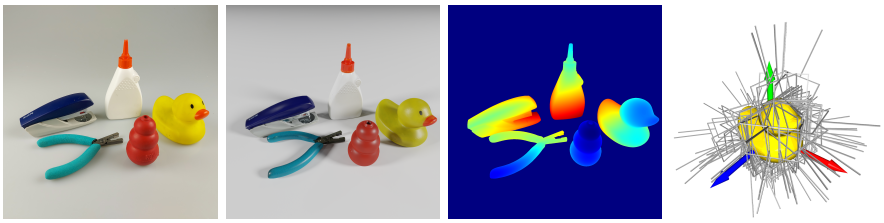
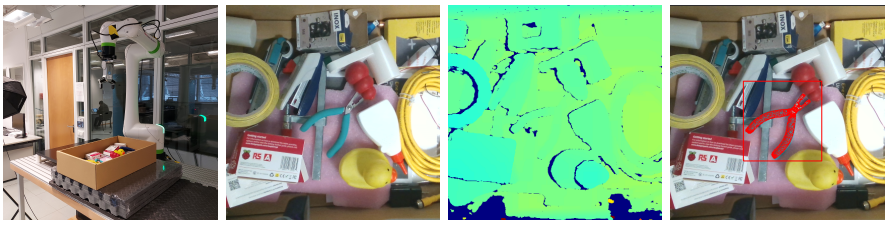
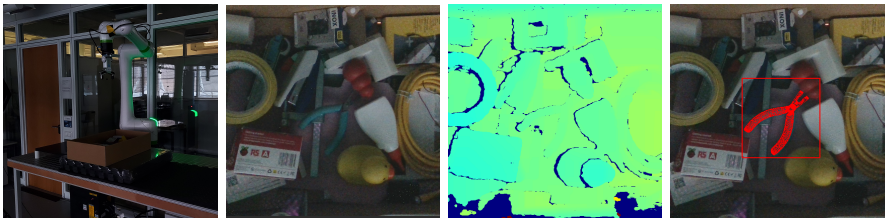


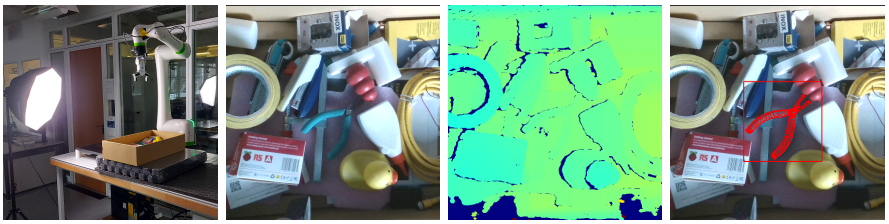
Figure 21: From left to right: the five selected objects photographed, photo-realistically rendered, depth rendered and automatically generated grasp poses for the duck.



(a) Experiments under diffused lighting condition.



(b) Experiments under low lighting condition.



(c) Experiments under spot lighting condition.

Figure 22: The figure showing the real world grasping experiments under diffused **(a)**, low **(b)**, spot **(c)** lighting conditions. The experimental setup, RGB view, depth view and the predicted pose of pliers on RGB are shown from the left to the right.

Figure 23: IPA2X Robot during Ljubljana demonstration.

6 European Project: Intelligent Pedestrian Assistant to Everyone (IPA2X)

During 2022, the Chair has been leading the EIT European Project "IPA2X" (www.ipa2x.eu). The project is supported by EIT Urban Mobility an initiative of the European Institute of Innovation and Technology (EIT), a body of the European Union. EIT Urban Mobility acts to accelerate positive change on mobility to make urban spaces more liveable. The project is co-funded by EIT and has an overall budget of 1.1 million euros.

IPA2X addresses the problem of road traffic injuries and the lack of pedestrian safety, especially regarding people with disabilities or elderly people. IPA2X creates an alliance among the most important living labs on autonomous driving, research institutes and OEMs, to improve pedestrian crossings via the development of a new intelligent pedestrian assistant robot (IPA2X). More specifically, the intelligent autonomous rover facilitates crossing of intersections for children and elderly people, thus promoting zero accident cities by increasing their safety. The rover will reduce costs for traffic helpers currently used as risk mitigation. On the technical side the project features an autonomous zebra-crossing pedestrian assistant composed of:

- An intelligent rover equipped with cutting-edge technologies (next generation computing platform, artificial intelligence, 5G)
- Distributed sensing and increased awareness via vehicle to "X" (V2X)

The rover interacts with incoming vehicles and the project develops an in-vehicle Human Machine Interface (HMI) and user interface to display warnings.

The pedestrian assistant robot has been demonstrated (including assessment of user-acceptance) with demos in the three partner cities (Milano, Modena, Ljubljana) and has achieved a considerable public reach (see www.ipa2x.eu for the relevant news/newspapers links).

In addition to the Technical University of Munich, the consortium consists of the following European partners: Cities: City of Milan and AMAT, Municipality of Modena, Av Living Lab Ljubljana; Companies: SKODA AUTO, Lifetouch Srl, Hipert Srl, Evidence Srl; Universities: Czech Technical University in Prague, Technical University of Munich.

7 Basic Information of the Chair of 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

- Tomasz Kloda, Dr.
- Andrea Bastoni, Dr.
- Alexander Züpke, Dr.
- Harald Bayerlein, Dr.
- Mirco Theile, M.Sc.
- Bingzhuo Zhong, M.Sc.
- Denis Hoornaert, M.Sc.
- Hongpeng Cao, M.Eng.
- Daniele Bernardini, M.Sc.

- Binqi Sun, M.Sc.
- Raphael Trumpp, M.Sc.
- Lukas Dirnberger, M.Sc.
- Hanyu Wu (MS student)
- Federico Wyrwal (MS student)
- Andres Zapata Rodriguez (MS Student)
- Tobias Mascetta (MS Student)

Research Focus

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

Competence

- System-level programming
- Embedded system software design
- Hardware modules design for FPGAs
- Real-time operating systems
- Reinforcement learning for CPS

Infrastructure

- 3 DOF helicopter
- Embedded and FPGA multi-core development platforms
- High-performance servers
- Linear inverted pendulum
- Fused filament fabrication, dual-head 3D printer
- F1/10 autonomous cars
- FANUC CRX collaborative robot arm

Collaborations

- University of Illinois at Urbana-Champaign, USA
- University of California, Berkeley, USA
- Boston University, USA
- University of Colorado Boulder, USA
- University of Waterloo, Canada
- University of Modena and Reggio Emilia, Italy
- Federal University of Santa Catarina, Brazil
- EURECOM, Sophia Antipolis, France
- LAAS–CNRS, France

Courses

- Concepts and Software Design for Cyber-Physical Systems
- Tutorial Concepts and Software Design for Cyber-Physical Systems
- Advanced Seminar on Safe Cyber-Physical Systems
- PhD-Seminar on Real-Time Cyber-Physical Systems
- Cyber-Physical Systems Lab: Autonomous Applications
- Simplex: Fault-Tolerant Control Strategy for Real-Time Cyber-Physical Systems - Laboratory
- Design and Analysis of Digital Control Systems
- Tutorials on Design and Analysis of Digital Control Systems
- Simulation and Control of Mechanical Systems

Humboldt Sponsored Research

Selected Publications 2022

Journal

- Bingzhuo Zhong, Abolfazl Lavaei, Majid Zamani, and Marco Caccamo. Automata-based controller synthesis for stochastic systems: A game framework via approximate probabilistic relations. *Automatica*, 147:110696, 2023
- Or D. Dantsker, Mirco Theile, and Marco Caccamo. A cyber-physical prototyping and testing framework to enable the rapid development of UAVs. *Aerospace*, 9(5), 2022
- Jiyang Chen, Tomasz Kloda, Rohan Tabish, Ayoosh Bansal, Chien-Ying Chen, Bo Liu, Sibin Mohan, Marco Caccamo, and Lui Sha. Schedguard++: Protecting against schedule leaks using linux containers on multi-core processors. *ACM Transactions on Cyber-Physical Systems*, 2022

Conference proceeding

- Bingzhuo Zhong, Abolfazl Lavaei, Majid Zamani, and Marco Caccamo. Controller synthesis for nonlinear stochastic games via approximate probabilistic relations. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–2, 2022
- Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. A set-based approach for synthesizing controllers enforcing ω-regular properties over uncertain linear control systems. In *2022 American Control Conference (ACC)*, pages 1575–1581. IEEE, 2022
- Bingzhuo Zhong, Hongpeng Cao, Majid Zamani, and Marco Caccamo. Towards safe ai: Sandboxing dnns-based controllers in stochastic games. In *Proceedings of the Thirty-Seven AAAI Conference on Artificial Intelligence*, 2023
- Ameneh Nejati, Bingzhuo Zhong, Marco Caccamo, and Majid Zamani. Controller synthesis for unknown polynomial-type systems: A data-driven approach. In *2022 2nd International Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems (CAAD-CPS)*, pages 11–12. IEEE, 2022
- Ameneh Nejati, Bingzhuo Zhong, Marco Caccamo, and Majid Zamani. Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates. In *Learning for Dynamics and Control Conference*, pages 763–776. PMLR, 2022
- Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. Synthesizing safety controllers for uncertain linear systems: A direct data-driven approach. In *Proceedings of the 6th IEEE Conference on Control Technology and Applications*, 2022
- Hongpeng Cao, Mirco Theile, Federico G. Wyrwal, and Marco Caccamo. Cloud-edge training architecture for sim-to-real deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9363–9370, 2022
- Mattia Nicolella, Shahin Roozkhosh, Denis Hoornaert, Andrea Bastoni, and Renato Mancuso. RT-Bench: An extensible benchmark framework for the analysis and management of real-time applications. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems (RTNS)*, pages 184–195, 2022
- Shahin Roozkhosh, Denis Hoornaert, Ju Hyoung Mun, Tarikul Islam Papon, Ahmed Sanaullah, Ulrich Drepper, Renato Mancuso, and Manos Athanassoulis. Relational memory: Native in-memory accesses on rows and columns. In *Proceedings 26th International Conference on Extending Database Technology (EDBT)*, pages 66–79, March 2023
- Shahin Roozkhosh, Denis Hoornaert, and Renato Mancuso. CAESAR: Coherence-aided elective and seamless alternative routing via on-chip FPGA. In *Proceedings 43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022
- Zhishen Zhang, Yuwen Shen, Binqi Sun, Tomasz Kloda, and Marco Caccamo. Memory allocation for low-power real-time embedded microcontroller: a case study. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2022
- Tomasz Kloda, Jiyang Chen, Antoine Bertout, Lui Sha, and Marco Caccamo. Latency analysis of self-suspending task chains. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1299–1304, 2022

Software Development

- Jailhouse Real-Time. https://gitlab.com/minervasys/public/jailhouse
- RT-Bench Framework. https://gitlab.com/rt-bench/rt-bench
- Jailhouse Cache-coloring. Jailhouse ML.
  https://groups.google.com/g/jailhouse-dev/c/K4rqZxpxa0U
- Virtio prototype (see [39]). https://github.com/gschwaer/rt-virtio
- 6D Pose Estimation Training Pipeline. https://github.com/HP-CAO/6IMPOSE
- Synthetic Data Generation. https://github.com/LukasDb/BlenderSyntheticData
- Robotic Grasping. https://github.com/LukasDb/HumanRobotInteraction

References

[1] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *Proceedings 19th IEEE Real-Time Systems Symposium (Cat. No.98CB36279)*, pages 4–13, 1998.

[2] Siemens AG. Jailhouse hypervisor. https://github.com/siemens/. Accessed: 2022-12-03.

[3] Andrea Bastoni. Jailhouse Cahe-Coloring Proposal. https://groups.google.com/g/jailhouse-dev/c/K4rqZxpxa0U.

[4] Andrea Bastoni. Jailhouse Public Repository with Real-Time Extensions. https://gitlab.com/minervasys/public/jailhouse.

[5] Hongpeng Cao, Lukas Dirnberger, Daniele Bernardini, Cristina Piazza, and Marco Caccamo. 6impose: Bridging the reality gap in 6d pose estimation for robotic grasping, 2022.

[6] Hongpeng Cao, Mirco Theile, Federico G. Wyrwal, and Marco Caccamo. Cloud-edge training architecture for sim-to-real deep reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9363–9370, 2022.

[7] Jiyang Chen, Tomasz Kloda, Rohan Tabish, Ayoosh Bansal, Chien-Ying Chen, Bo Liu, Sibin Mohan, Marco Caccamo, and Lui Sha. Schedguard++: Protecting against schedule leaks using linux containers on multi-core processors. *ACM Transactions on Cyber-Physical Systems*, 2022.

[8] Or D. Dantsker, Mirco Theile, and Marco Caccamo. A cyber-physical prototyping and testing framework to enable the rapid development of UAVs. *Aerospace*, 9(5), 2022.

[9] Jerry Ding, Maryam Kamgarpour, Sean Summers, Alessandro Abate, John Lygeros, and Claire Tomlin. A stochastic games framework for verification and control of discrete time stochastic hybrid systems. *Automatica*, 49(9):2665–2674, 2013.

[10] Bosch GmbH. ETAS RTA Hypervisor. https://www.etas.com/en/products/rta-vrte.php. Accessed: 2021-02-08.

[11] SYSGO GmbH. PikeOS Hypervisor. https://www.sysgo.com.

[12] Sofie Haesaert, Sadegh Esmaeil Zadeh Soudjani, and Alessandro Abate. Verification of general Markov decision processes by approximate similarity relations and policy refinement. *SIAM Journal on Control and Optimization*, 55(4):2333–2367, 2017.

[13] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. PVN3D: A Deep Point-wise 3D Keypoints Voting Network for 6DoF Pose Estimation, March 2020.

[14] Denis Hoornaert, Golsana Ghaemi, Andrea Bastoni, Renato Mancuso, Marco Caccamo, and Giulio Corradi. On the interplay of computation and memory regulation in multicore real-time systems. In *The 15th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, page 29, 2022.

[15] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235:3–35, 2013.

[16] Maryam Kamgarpour, Jerry Ding, Sean Summers, Alessandro Abate, John Lygeros, and Claire Tomlin. Discrete time stochastic hybrid dynamical games: Verification & controller synthesis. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6122–6127, 2011.

[17] T. Kloda, M. Solieri, R. Mancuso, N. Capodieci, P. Valente, and M. Bertogna. Deterministic memory hierarchy and virtualization for modern multi-core embedded systems. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, page 1–14, 2019.

[18] Tomasz Kloda, Jiyang Chen, Antoine Bertout, Lui Sha, and Marco Caccamo. Latency analysis of self-suspending task chains. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1299–1304, 2022.

[19] Abolfazl Lavaei, Sadegh Soudjani, Alessandro Abate, and Majid Zamani. Automated verification and synthesis of stochastic hybrid systems: A survey. *Automatica*, 2022. to appear.

[20] Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Compositional abstraction-based synthesis for networks of stochastic switched systems. *Automatica*, 114:108827, 2020.

[21] Juri Lelli, Claudio Scordino, Luca Abeni, and Dario Faggioli. Deadline scheduling in the Linux kernel. *Software: Practice and Experience*, 46(6):821–839, 2016.

[22] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of International Conference on Learning Representations(Poster)*, 2016.

[23] Sebastian Maierhofer, Paul Moosbrugger, and Matthias Althoff. Formalization of intersection traffic rules in temporal logic. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1135–1144, 2022.

[24] R. Mancuso, R. Dudko, E. Betti, M. Cesati, M. Caccamo, and R. Pellizzoni. Real-time cache management framework for multi-core architectures. In *2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, page 45–54, 2013.

[25] Renato Mancuso. Boston University. http://www.bu.edu/cs/profiles/renato-mancuso/.

[26] National Transportation Safety Board. Preliminary report highway-wy18mh010, 2018.

[27] Ameneh Nejati, Bingzhuo Zhong, Marco Caccamo, and Majid Zamani. Controller synthesis for unknown polynomial-type systems: A data-driven approach. In *2022 2nd International Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems (CAAD-CPS)*, pages 11–12. IEEE, 2022.

[28] Ameneh Nejati, Bingzhuo Zhong, Marco Caccamo, and Majid Zamani. Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates. In *Learning for Dynamics and Control Conference*, pages 763–776. PMLR, 2022.

[29] Mattia Nicolella, Shahin Roozkhosh, Denis Hoornaert, Andrea Bastoni, and Renato Mancuso. RT-Bench: An extensible benchmark framework for the analysis and management of real-time applications. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems (RTNS)*, pages 184–195, 2022.

[30] Panagiotis Patrinos, Pantelis Sopasakis, Haralambos Sarimveis, and Alberto Bemporad. Stochastic model predictive control for constrained discrete-time markovian switching systems. *Automatica*, 50(10):2504–2514, 2014.

[31] Amir Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57. IEEE, 1977.

[32] The Linux Foundation Projects. ACRN hypervisor. https://projectacrn.org/. Accessed: 2021-02-08.

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[34] Shahin Roozkhosh, Denis Hoornaert, and Renato Mancuso. CAESAR: Coherence-aided elective and seamless alternative routing via on-chip FPGA. In *Proceedings 43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022.

[35] Shahin Roozkhosh, Denis Hoornaert, Ju Hyoung Mun, Tarikul Islam Papon, Ahmed Sanaullah, Ulrich Drepper, Renato Mancuso, and Manos Athanassoulis. Relational memory: Native in-memory accesses on rows and columns. In *Proceedings 26th International Conference on Extending Database Technology (EDBT)*, pages 66–79, March 2023.

[36] Debayan Roy, Sumana Ghosh, Qi Zhu, Marco Caccamo, and Samarjit Chakraborty. Goodspread: Criticality-aware static scheduling of CPS with multi-qos resources. In *41st IEEE Real-Time Systems Symposium (RTSS)*, page 178–190. IEEE, 2020.

[37] Debayan Roy, Clara Hobbs, James H Anderson, Marco Caccamo, and Samarjit Chakraborty. Timing debugging for Cyber-Physical Systems. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, page 1893–1898. IEEE, 2021.

[38] Debayan Roy, Licong Zhang, Wanli Chang, Dip Goswami, Birgit Vogel-Heuser, and Samarjit Chakraborty. Tool integration for automated synthesis of distributed embedded controllers. *ACM Trans. Cyber-Phys. Syst.*, 6(1), nov 2021.

[39] Gero Schwaericke, Rohan Tabish, Rodolfo Pellizzoni, Renato Mancuso, Andrea Bastoni, Alexander Zuepke, and Marco Caccamo. A real-time virtio-based framework for predictable inter-VM communication. In *2021 IEEE International Real-Time Systems Symposium (RTSS)*, 2021.

[40] Green Hills Software. GHS Integrity. https://www.ghs.com/products/rtos/ integrity_virtualization.html.

[41] Robert S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and control*, 54(1-2):121–141, 1982.

[42] RT-Bench Team. RT-Bench Framework. https://gitlab.com/rt-bench/rt-bench.

[43] Wolfgang Thomas. Automata on infinite objects. In *Formal Models and Semantics*, pages 133–191. Elsevier, 1990.

[44] Unimore. HiPeRT Lab. https://hipert.unimore.it.

[45] Jan C. Willems, Paolo Rapisarda, Ivan Markovsky, and Bart L. M. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[46] Pian Yu and Dimos V. Dimarogonas. Distributed motion coordination for multirobot systems under LTL specifications. *IEEE Transactions on Robotics*, 38(2):1047–1062, 2022.

[47] H. Yun, R. Mancuso, Z. P. Wu, and R. Pellizzoni. PALLOC: DRAM bank-aware memory allocator for performance isolation on multicore platforms. In *2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, page 155–166, 2014.

[48] H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha. Memory bandwidth management for efficient performance isolation in multi-core platforms. *IEEE Transactions on Computers*, 65(2):562–576, 2016.

[49] Zhishen Zhang, Yuwen Shen, Binqi Sun, Tomasz Kloda, and Marco Caccamo. Memory allocation for low-power real-time embedded microcontroller: a case study. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2022.

[50] Bingzhuo Zhong, Hongpeng Cao, Majid Zamani, and Marco Caccamo. Towards safe ai: Sandboxing dnns-based controllers in stochastic games. In *Proceedings of the Thirty-Seven AAAI Conference on Artificial Intelligence*, 2023.

[51] Bingzhuo Zhong, Abolfazl Lavaei, Majid Zamani, and Marco Caccamo. Controller synthesis for nonlinear stochastic games via approximate probabilistic relations. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–2, 2022.

[52] Bingzhuo Zhong, Abolfazl Lavaei, Majid Zamani, and Marco Caccamo. Automata-based controller synthesis for stochastic systems: A game framework via approximate probabilistic relations. *Automatica*, 147:110696, 2023.

[53] Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. A set-based approach for synthesizing controllers enforcing ω-regular properties over uncertain linear control systems. In *2022 American Control Conference (ACC)*, pages 1575–1581. IEEE, 2022.

[54] Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. Synthesizing safety controllers for uncertain linear systems: A direct data-driven approach. In *Proceedings of the 6th IEEE Conference on Control Technology and Applications*, 2022.

[55] Alexander Zuepke, Andrea Bastoni, Weifan Chen, Marco Caccamo, and Renato Mancuso. MemPol: Policing Core Memory Bandwidth from Outside of the Cores. In *29th IEEE Real-Time and Embedded Technology and Applications Symposium RTAS (to appear)*, 2023.