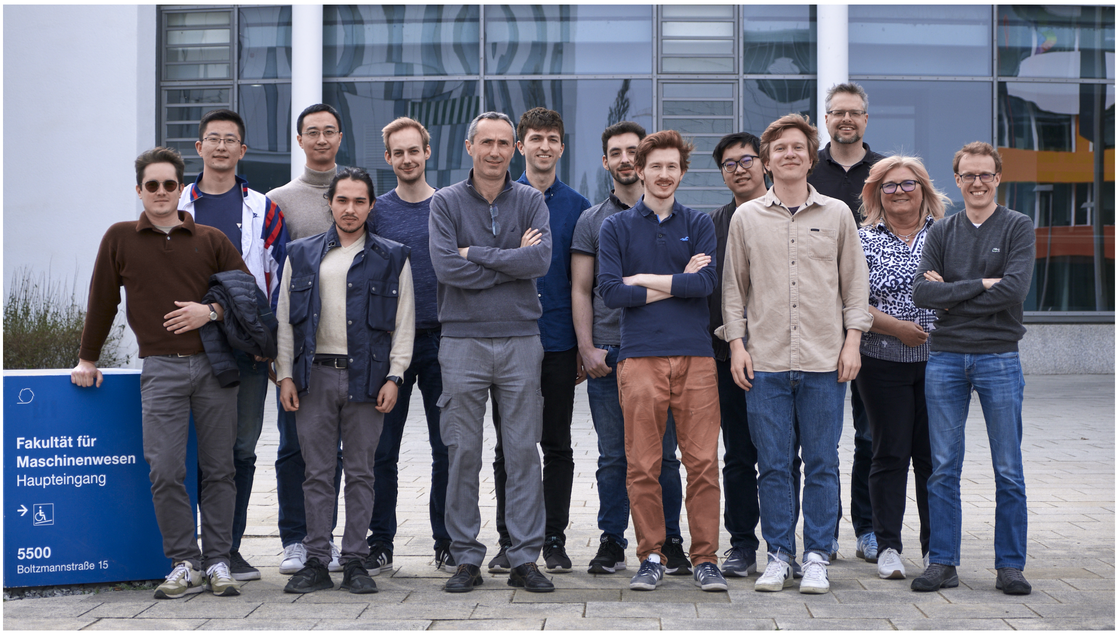
Chair of Cyber-Physical Systems in Production Engineering: Annual Report 2023

January 17, 2024



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, the 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 2023, the research activities of the Chair focused on the following topics: a) design and implement novel resource management policies for embedded real-time systems running on high-performance heterogeneous platforms, b) develop new reinforcement learning architectures for CPS, c) design architectures for sandboxing controllers in CPS, and d) develop synthetic training paradigms for 6D pose recognition and policy learning in robotic manipulation.

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 2023, RTAS 2024, ECRTS 2023, DAC 2024, DATE 2024, AAAI 2023, IROS 2023, ICRA 2023, ICCPS 2023, IEEE GLOBECOM 2023, ACC 2023, CDC 2023, ECC 2023, HSCC 2023, ICAR 2023, ITSC 2023, as well as Journal of Real-Time Systems, ACM Transaction on Embedded Computing Systems, IEEE Transactions on Automatic Control, IEEE Transactions on Systems, Man and Cybernetics: Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, IEEE Transactions on Intelligent Vehicles, IEEE Robotics and Automation Letters, IEEE Embedded Systems Letters, IEEE Control Systems Letters, International Journal of Digital Earth, IEEE Access, and Intelligent Computing.

2 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.

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 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 (see [70]); the optimization of shared resources such as caches while simultaneously providing real-time guarantees(see [68]); scheduling of real-time resources on new classes of heterogeneous computing elements (see [69]).

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 (see [82, 81, 37]) that prove to perform well "in practice", and we focus our integration effort at both Operating System (OS) and Hypervisor levels. Hypervisors (see [4, 59, 67, 25, 24]) 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 (see [49, 3, 88, 31, 65]), we actively participate in the development of open source hypervisors (see [9, 58]) to make the developed techniques not only readily available, but also supported by an active community [8]. On these topics, we also actively collaborate with highly skilled international teams [45, 77], which pursue objectives close to ours. We additionally actively develop open source real-time frameworks (see [74]) to improve the interoperability and exchange of results among international research groups.

In the remainder of this section, we present more details on our recent works [70, 68, 69, 49]. These papers are representative of our research efforts in 2023 towards predictable and high-performance resource management of CPS on heterogeneous platforms.

2.1 Minimizing Cache Usage for Real-time Systems [68]

The main benefit of cache partitioning in real-time systems is that it removes inter-task interference: preempting task will not evict the cached memory blocks of preempted task if both tasks use separate cache partitions. Cache partitioning can be implemented using specific hardware extensions (*e.g.*, Intel's Cache Allocation Technology [33] or ARM's Lockdown by master [42]) [26] or in software by exploiting address mapping between main memory and cache lines (*e.g.*, cache coloring) [39]. However, if the task's working set does not fit into the task's private cache partition, the task will see an increased number of cache misses and, consequently, increased execution time. To mitigate this problem, various optimization techniques are used to allocate cache partitions of adequate size to tasks in function of their timing constraints.

Cache partitioning optimization methods for real-time systems focus on finding the cache partitioning under the assumption that all available cache segments can be allocated to the tasks. Despite the wealth of the literature, reducing cache usage is not part of the optimization criteria. However, for a variety of reasons, unrestrained cache usage might be of concern to embedded engineers. Multilevel caches often consume about half the processor energy [29], and choosing the processors with a last-level cache size fitting the application requirements or appropriately selecting its size can largely reduce power dissipation. Otherwise, the unallocated partitions can be used to improve the quality of service of the best-effort tasks. In the context of partitioned multi-core systems where the task-to-core allocation is fixed beforehand, the cache partitioning problem boils down, in fact, to minimize the cache usage of every core while ensuring its schedulability (see Figure 1). When a task-to-core allocation is not given, the cache minimization can be used as a sub-procedure in the task and cache co-allocation method.

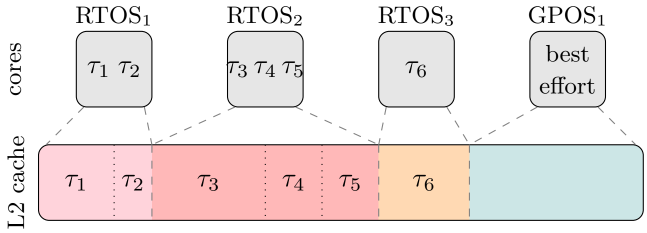


Figure 1: Last-level cache dimensioning in a multicore mixed-criticality system.

An apparent solution at hand for minimizing the cache usage is to invoke iteratively one of the standard cache partitioning methods with a smaller (bigger) cache size at each step until the system becomes unschedulable (schedulable). Indeed, several cache allocation methods are easily amenable to this approach or can stop earlier when schedulability is guaranteed, and there is no point in further reduction of the system utilization (*e.g.*, gradient descent for minimizing system utilization [36]). In this research, we also report such methods and describe the required modifications. On the other hand, some methods use remaining cache segments to allow faster convergence (*e.g.*, branch-and-bound [5]), and specific approaches must be proposed. Moreover, restarting the search for each cache size without any knowledge of the previous iterations might not be particularly efficient, and certain proprieties of the schedulability tests, in particular, sustainability as suggested in [5], can be exploited to skip the redundant tests when going from one cache partition size to another.

In [68], we make the following contributions. For single-core preemptive scheduling, we formulate the cache minimization problem as an integer quadratically constrained program (IQCP), which can be solved optimally by a standard mathematical programming solver. To improve the

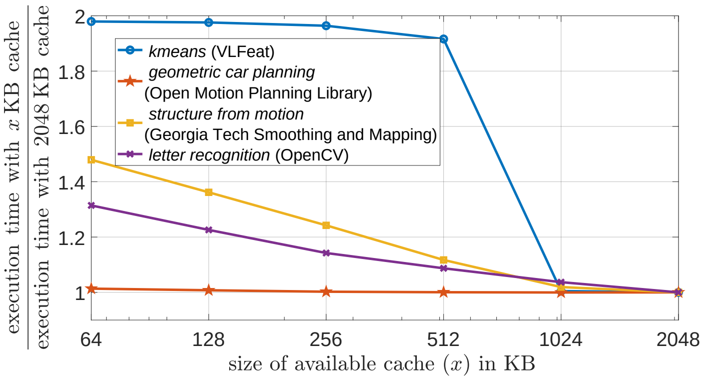


Figure 2: Benchmark's execution slowdown with x KB cache compared to full (2048 KB) cache.

efficiency of the IQCP solution, we propose a guided local search (GLS) heuristic that can obtain near-optimal solutions in a fraction of the solver's run time. Moreover, we apply the branch-and-bound (BB) and dynamic programming (DP) methods to the cache minimization problem. For single-core non-preemptive scheduling, we derive pseudo- and fully-polynomial time search algorithms that incorporate different schedulability tests. To evaluate the proposed methods, we conduct simulation experiments based on embedded programs to quantitatively compare different approaches in terms of their cache usage, schedulability ratio, and run time.

2.2 Co-Optimizing Cache Partitioning and Multi-Core Task Scheduling [70]

Nowadays, heterogeneous multiprocessor system-on-a-chip (MPSoC) platforms are routinely used for all those workloads that require performance, real-time capabilities, and limited size and power consumption. These workloads include, *e.g.*, applications found in autonomous driving, intelligent robotics, and unmanned aerial vehicles domains. Towards guaranteeing real-time performance, these platforms pose an unprecedented challenge to the management of the memory hierarchy. With a focus on the core complex of such MPSoCs, sharing caches among cores prevents analyzing tasks in isolation, thus complicating an accurate estimation of the tasks' worst-case execution times (WCETs). Unsurprisingly, therefore, to mitigate this problem, both software-based [44, 38] and hardware-based [80, 66] cache partitioning techniques have been exploited. Although effective, cache partitioning limits the amount of cache available to (groups of) real-time tasks. Therefore, the impact of cache partitioning on the WCET can be non-negligible for a *cache-sensitive* workload. This effect is illustrated in Figure 2, which reports the *slowdown* due to reduced cache availability of four benchmark applications. For example, *kmeans* is almost two times slower when it runs with a cache partition smaller than 256 KB instead of 1024 KB.

In [70], we study the *integrated problem* of (1) assigning real-time tasks to cores and (2) reserving cache for tasks running on each core. The goal is to achieve a solution where the tasks are *schedulable*, *i.e.*, each task meets its real-time requirements, *e.g.*, deadline. The main focus of the proposed optimization strategies is on *non-preemptive fixed-priority* (NP-FP) scheduling. We further assume that tasks are *statically assigned* to cores. Partitioned schedulers have simpler implementations and generally lower overheads [7], and non-preemption naturally separates computation from data management phases (*e.g.*, [73]). Also, we assess the flexibility of our framework to support other scheduling policies such as preemptive and non-preemptive *earliest deadline first* (EDF). Given the *interdependencies* of the three sub-problems, viz., task allocation, cache partitioning, and schedulability analysis, we present an *integrated* solution to improve the likelihood of establishing system schedulability. In particular, we propose a *nested multi-layer, hybrid* optimization framework to *explore the interplay* between the sub-problems. In this framework, (i) the outer layer partitions the shared cache, (ii) the middle layer allocates tasks, and (iii) the inner layer performs the schedulability analysis.

Although the selection of tasks is optimized for NP-FP scheduling, the multi-layer framework can be easily adapted to other scheduling policies by plugging in an appropriate schedulability test in the inner layer. To demonstrate this, we show experimental results under *preemptive EDF* (P-EDF) and *non-preemptive EDF* (NP-EDF) scheduling by using the tests adopted by [80] and [56, 10], without any modifications to the outer and middle layers. Results for NP-FP scheduling indicate that the performance of the framework depends on the cache-sensitivity of workloads with a schedulability improvement of up to 14.5% for tasks with low cache sensitivity and of up to 233.6% for highly cache-sensitive tasks, with an average improvement of 15.2%. NP-FP experiments also show that focusing on compatibility leads to better results (by 7.6% on average) than cache sensitivity. For P-EDF scheduling, the framework improves the schedulability by 8.7% on average compared to the approach in [80], while for NP-EDF scheduling, the average improvement is 19.2% compared to the techniques in [56, 10].

2.3 Schedulability Analysis of Sporadic Non-preemptive Gang Tasks on Hardware Accelerators [69]

Gang scheduling appeared as an efficient solution to the problem of job scheduling on highly parallel embedded architectures. Application threads grouped into a single gang are scheduled concurrently on distinct processing units. Scheduling them at the same time can avoid the overhead of context switching [79] or busy waiting at synchronization points [20] and help in utilizing the inter-thread cache benefit [34]. Due to the prohibitively expensive preemption cost of the hardware accelerators, running the gang tasks non-preemptively is often the only feasible solution. In [69], we propose new schedulability tests for fixed-priority non-preemptive gangs and show their applicability and potential on the Edge Tensor Processing Units (TPUs) [1].

Edge TPU is a custom ASIC designed for accelerating neural network inference on edge devices. It can improve the inference time by 30x compared with embedded CPUs [2]. Integrating Edge TPU accelerators into edge devices presents several challenges and can incur large memory and scheduling overhead if configured incorrectly. To investigate these issues, we benchmarked eight representative convolutional neural network models of various sizes on commercial off-the-shelf (COTS) Edge TPU hardware and outline some key findings related to real-time scheduling on multiple Edge TPUs.

Preemption cost. Edge TPU has small on-chip memory (8MB), thus limiting the size of the neural network model that can be stored in internal SRAM. Switching between different models causes a significant overhead as the model needs to be loaded into Edge TPU's internal cache every time. This can be observed in our experiments by comparing the neural network inference latency with and without parameter loading. Take Inception-v1 [72] as an example. The average inference latency without parameter loading is 6.58 ms. However, it grows to 20.17 ms when including parameter loading time.

Parallel processing. Besides the context switch overhead, running models larger than 8MB on an Edge TPU requires fetching the model parameters from the main memory for every inference, which incurs a high memory transaction latency. One approach to avoid latency is to pipeline a large model with multiple Edge TPUs. The model is divided into multiple segments using the Edge TPU Compiler, and each segment runs on a different Edge TPU. Although a model can be segmented with as many Edge TPUs as one likes, using more accelerators does not necessarily mean better performance. Figure 3 shows the relationship between neural network inference

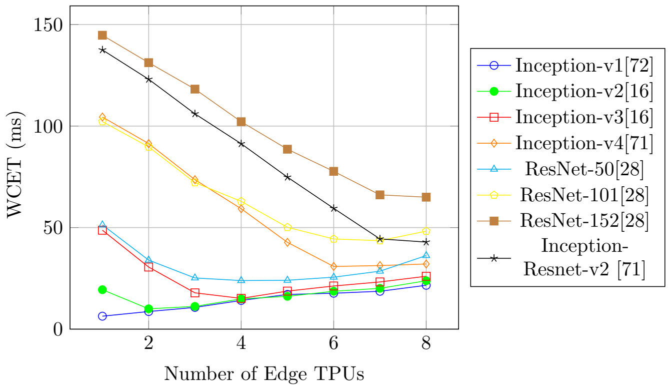


Figure 3: Edge multi-TPU neural network benchmarks executed on ASUS AI Accelerator CRL-G18U-P3D with 8 Edge TPUs.

time and the number of Edge TPUs used by the networks. For a large network, *e.g.*, ResNet-152 (57.53MB), having more Edge TPUs can indeed reduce inference time. However, for a small network such as Inception-v1 (5.72MB), the inference time increases with the number of Edge TPUs. This is because pipelining a model requires sending intermediate tensors from one Edge TPU to another and adds I/O latency.

Memory space limitation. It can be beneficial to change the number of segments of a model at run time when multiple models are executed on multiple Edge TPUs. However, a different segmentation of the same model generates a separate model executable file. This will occupy extra disk space and precious memory of the edge devices during run time. Given these trade-offs, we assume only one segmentation is used for each network model. We also assume that the segmentation is given by the system designer. Determining the optimal number of segments is out of the scope of this paper.

Rigid non-preemptive gang. In light of the above-discussed Edge TPU characteristics (high preemption cost, parallel execution, and constrained memory space), in [69], we propose to model neural networks running on Edge TPUs as *non-preemptive rigid gang* tasks. The non-preemptive execution avoids the slowdown caused by the model reloading. With pipelining, each neural network task can take more than one accelerator (intra-task parallelism). Such gang task runs simultaneously on a fixed number of distinct processing units in parallel as the model segmentation is fixed. In contrast to traditional multi-threaded scheduling, where the threads can execute independently between the synchronization points, gang scheduling starts all task threads simultaneously. While we believe that new customized scheduling policies can be proposed to efficiently overcome certain disadvantages of non-preemptive gang execution, in this work, we assume the traditional *fixed-priority* policy used in most embedded real-time applications and propose a gang task carry-in limitation technique to reduce the pessimism of the analysis.

To summarize, our paper [69] makes the following contributions:

- We present the first TPU-Pipelining runtime performance benchmarks on multi-TPU edge AI accelerators.

- We provide a linear-time utilization bound test for any work-conserving non-preemptive rigid gang scheduling.

- We propose two schedulability tests with quadratic and pseudo-polynomial complexity, respectively, for non-preemptive fixed-priority (NP-FP) rigid gang scheduling.

- We demonstrate the effectiveness of our schedulability test on both synthetic data and Edge TPU benchmarks. Our proposed schedulability test can achieve up to 46.5% and 85.7% additional task sets schedulable on synthetic task sets and Edge TPU benchmarks, respectively, compared to the state-of-the-art non-preemptive gang schedulability analysis.

2.4 Arm MUCH: Full-spectrum hardware-event-based Armv8 application profiler [49]

Software profiling is a dynamic program analysis technique where a program's behavior is modeled using data monitored at runtime. Hardware-based software profiling enables developers to better understand the execution of a program, monitoring how the hardware layer reacts to the application. The Arm architecture exposes Performance Monitor Units (PMUs) as architecture-specific, on-chip solution for low-overhead hardware event-based profiling. In this context, Hardware Event Monitors (HEMs) constitute the hardware events to be observed, and Performance Monitor Counters (PMCs) represent the actual architectural counter where the monitored information is stored. The Armv8 architecture defines a standard, common set of HEMs that should be present in any implementation, and a set of HEMs that could be optionally implemented by manufacturers.

Compared to other software profiling and debugging solutions (*e.g.*, Arm CoreSight), Arm hardware-event-based profiling has low overhead and generates less execution interference in terms of resource usage and bus accesses. Unfortunately, extensive application profiling and monitoring using HEMs is limited by the low number of PMCs that are typically available on Arm boards. In fact, PMCs are often a magnitude order fewer than the HEMs that could possibly be monitored. Therefore, despite the standard availability of PMCs and HEMs on Arm-based platforms, monitoring the *full hardware execution context* observing one single run of an application (or a benchmark) is not possible. This limitation makes it extremely hard to understand the *full hardware execution context* from a single application or benchmark's run.

MUltiCorrelation HEM reading and merging (MUCH) is a recent approach [78] that relies on statistical analysis to reconstruct the full-hardware application context across multiple runs of an application or benchmark. The approach has been developed to target explicitly complex multiprocessor systems-on-chip (MPSoC), where HEM monitoring and profiling is becoming progressively more important to master interference among mixed-criticality (real-time) applications [52]. In [78], the approach has been validated in a bare-metal setup (without operating system) on a PowerPC NXP T2080.

Given the increasing relevance of the Arm architecture for complex MPSoC used in safety critical automotive, industrial, and avionics domains, we propose Arm MUCH, an application profiler for the Armv8 architecture that adopts the MUCH-approach and runs on a complex operating systems such as Linux. With Arm MUCH we:

- Validate whether the MUCH approach can be applied to real-world Arm architectures together with a complex operating system such as Linux. Our results on a Raspberry PI 3 with Linux 5.9.93 confirm the applicability of the MUCH methodology to the Arm architecture even with a complex operating system.

- Contribute a framework for implementing MUCH on Armv8. In particular, we automated the HEM allocation at application runtime, the gathering of the retrieved data and performing the statistical MUCH methodology.

- Interface the Arm MUCH framework with multiple profiling technologies for Arm Linux, namely *perf*, *eBPF*, and inline assembly for manual HEMs allocation.

- Explore how to derive the minimal set of HEMs that best characterize an application. This enables confidence in capturing the key properties of an application despite the limited number of monitored PMCs.

- Implement AI-based HEM-prediction systems that use the statistical data retrieved by MUCH to reconstruct the *complete set of HEMs inside one benchmark execution* by forecasting all non-monitored HEMs.

3 Deep Reinforcement Learning for Cyber-Physical Systems

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, autonomous driving, and manipulation. However, the training of the DRL agents is typically sample inefficient and unsafe during the exploration. Moreover, the learned agents are parameterized with deep neural networks, which is hard to predict and verify, imposing safety risks for the deployment on physical-systems.

Recently, a research focus has been shifted to the integration of data-driven DRL and model-based policy, leading to a residual diagram, which holds the promise for dealing with complex dynamics while retaining the (provable and verifiable) advantages of the model-based approaches. Such a residual control diagram can take advantage of both model-based controllers and data-driven DRL, as the model-based policy can guide the exploration of DRL agents during training and regulate the behavior of the DRL controller. Meanwhile, the DRL controller learns to effectively deal with the uncertainties and compensate for the modelling errors faced by the model-based policies. Our recent research investigated the potentials of residual DRL for improving the safety-assurance and performance of the DRL-enabled cyber-physical systems.

Another related research project was focused on reducing training data demand in DRL multi-agent drone trajectory planning for IoT networks [14].

3.1 Safety-aware Residual Reinforcement for Reliable Cyber-Physical Systems [13]

DRL learns control policies from interacting with the environment to tackle the non-linearity and uncertainties presented in complex control tasks, achieving impressive performance. However, applying DRL to safety-critical autonomous systems remains a challenging problem. A critical reason is that the control policy of DRL is typically parameterized by DNNs, whose behaviors are hard to predict and verify, raising concerns about safety and stability.

DRL-enabled control systems should satisfy some safety constraints and also feature stability. Model-based approaches focus on constructing a safety set, and the DRL agent can only act in this constrained space. According to this direction, the control Lyapunov function (CLF) is often used to constrain the state space with the objective that all actions will lead the system towards a stable point. However, finding such CLF is often a challenging task for nonlinear systems. Given the desired safety specification, one can also leverage the control barrier function and reachability analysis to certify the control command to satisfy the safety requirement. The approaches are mainly designed to ensure the system's safety, whereas how to guarantee stability remains an open problem. Moreover, the model-based approaches are generally limited by modeling errors and rely on a more accurate dynamics model to expand the safety region.

Learning-based approaches aim to embed the knowledge of safety and stability during training, such that the agent is guided to learn to stabilize the system. In learning-based methods, some approaches aim to learn a Lyapunov function from sampled data and use it as an additional critic network to regulate the optimization of control policy toward decreasing the Lyapunov function. Alternatively, incorporating CLF constraint in the reward function encourages the agent to learn to stabilize the system close to the equilibrium point. Recently, researchers discovered that if the reward of DRL is CLF-like, the system controlled by a well-trained DRL agent can be proved to retain stability. Building on those findings, the challenges moving forward are twofold:

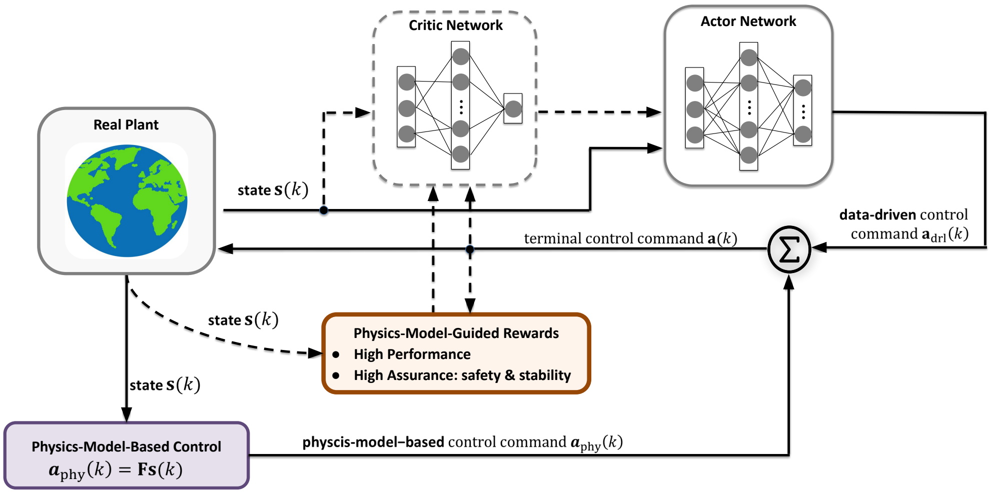
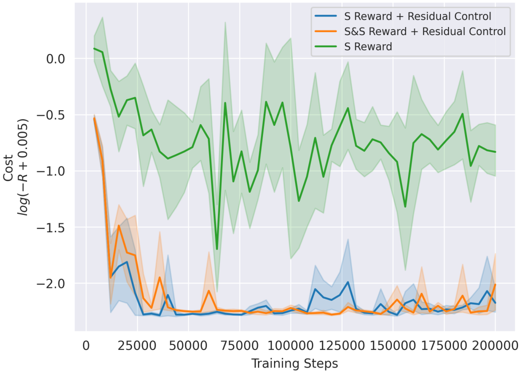

Critic Network
Actor Network
Real Plant
state $\mathbf{s}(k)$
data-driven control command $\mathbf{a}_{\mathrm{drl}}(k)$
terminal control command $\mathbf{a}(k)$
Σ
state $\mathbf{s}(k)$
Physics-Model-Guided Rewards
● High Performance
● High Assurance: safety & stability
state $\mathbf{s}(k)$
Physics-Model-Based Control
$\boldsymbol{a}_{\mathrm{phy}}(k) = \mathbf{Fs}(k)$
physcis-model–based control command $\boldsymbol{a}_{\mathrm{phy}}(k)$

Figure 4: The plot shows the diagram of the proposed Phy-DRL framework. It consists of a real plant, a physics-model-based controller, a DRL algorithm of *actor-critic* architecture, and a physics-model-guided reward module. The terminal control command is computed by taking the summation of the action generated from the model-based controller and the action output from the actor-network of DRL. The states, control actions and the reward computed from the Physical-Model-Guided Reward module are saved as training data for optimizing the critic and actor networks. The dashed lines indicate the additional procedures for training.



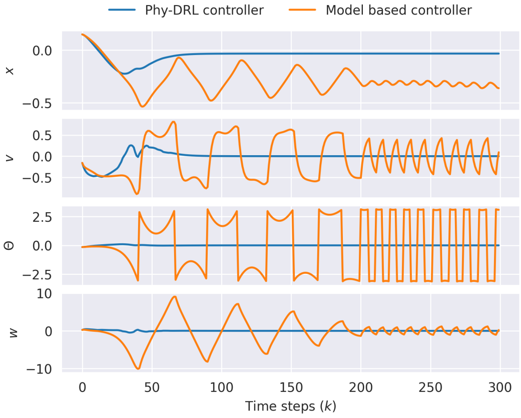


Figure 5: The plot illustrates the training progress with and without residual mechanisms. In the residual control diagram, the model-based control guides the exploration of the DRL, significantly improving the converging speed and leading to the enlarged reward (reduced cost).

Figure 6: The plot shows an example of state trajectories of the system controlled by the proposed Phy-DRL. The trajectory of Phy-DRL satisfies the introduced safety and stability conditions.

what is the formal guidance for constructing such CLF-like rewards for DRL, and how do we regulate DRL to concurrently guarantee safety and stability?

Inspired by the residual control diagram, we propose a novel physics-model-regulated DRL framework to guide and regulate the pure data-driven approach using model-based knowledge, as shown in Fig 4. Specifically, we leverage Lyapunov stability theory with a linearized model to design a Lyaponov-like reward function that can encourage the DRL to learn to stabilize the system and stay within the safety envelope directly. Furthermore, we mathematically derive safety and stability testing conditions using model knowledge for guaranteeing safety. At last, we make the model-based controller and DRL work under the residual control diagram to output more robust control commands. We evaluate the effectiveness of the Phy-DRL on an inverted pendulum system. The experimental results suggest that the Phy-DRL features remarkably accelerated training and enlarged reward, as shown in Fig 6.

3.2 Residual Reinforcement Learning for High-Performance Cyber-Physical Systems

The domain of autonomous racing provides a challenging test bed for real-world applications of cyber-physical systems. In the F110 racing series [54], RC cars of 1/10 scale race (see Fig. 7) autonomously against the clock, pushing the car's physics to its limits when tracking an optimized race line. Classical controllers for path-tracking, *e.g.*, Pure Pursuit, only achieve adequate performance with tedious racetrack-dependent parameter tuning. Modern model-based controllers like model-predictive control require accurate vehicle models and have high computational requirements.

Figure 7: RC car used in the F110 racing series; equipped with LiDAR sensor.

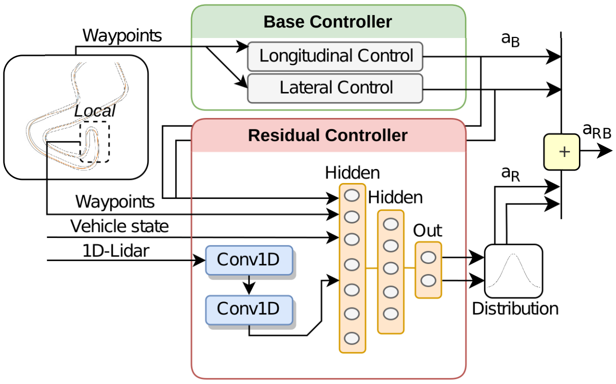


Figure 8: The base controller's action a_{B} is amended by learned residual action a_{R} to form the combined action a_{RB}

In [76], we demonstrate that the use of residual policy learning (RPL) combines the *reliability* of classical control approaches with the *adaptability* of learning-based controllers in the context of simulated F1TENTH racing. Our approach uses a pure pursuit controller [17] as the base controller, while the residual controller uses a model architecture as shown in Fig. 8. The residual part is trained with PPO [63] for 10 million interaction steps with the environment on nine different real-world replicated racetracks in parallel and evaluated on three additional, new racetracks. The residual head takes the common vehicle dynamics states as input as well as a 1D-LiDAR as an additional modality to learn an understanding of the car's surroundings. We propose a new simplified reward term that maximizes the longitudinal velocity while penalizing crashes and excessive lateral velocities; typically, other works [11] require a more complex reward signal like lap progress to work well.

Our experiments show that compared to the base controller, adding the learned residual part reduces lap times on the training race tracks by up to 7.06 %; performance does not drop when evaluated on the new test race tracks. Overall, the relative improvement in lap time is 4.55 %. This evaluation demonstrates the capability of our proposed approach to generalize to new racetracks with comparable performance to the results observed on the training racetracks. A detailed analysis for the Sao Paulo racetrack in Fig. 9 shows that the residual controller gains the lap time improvements in the curvey section; the controller has learned a sophisticated turning behavior by decelerating until the curves' apex and then accelerating strongly.

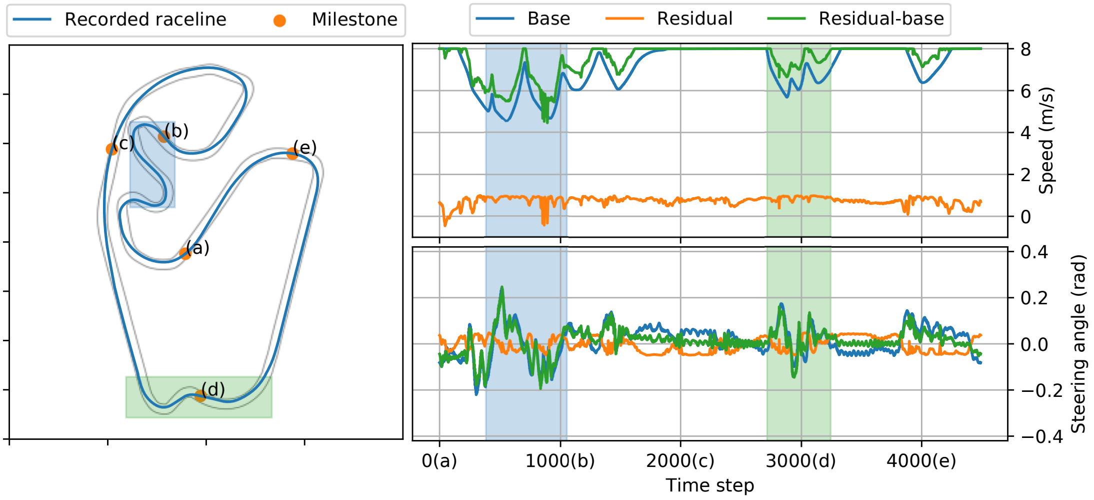


Figure 9: Action profile for the Sao Paulo racetrack.

We presented a new approach for vehicle control of autonomous racing cars using RPL with zero-shot generalization capability to unknown racetracks. Additionally to an ablation study of the design of the residual model, the envisioned future work will aim at exploring the applicability of RPL to bridge the sim2real gap on real F110 cars.

3.3 Recurrent Network Architectures for Reinforcement Learning in Highly-Interactive Multi-Agent Scenarios

Many cyber-physical systems interact with other agents/systems in multi-agent scenarios, *e.g.*, warehouse robots and workers or autonomous vehicles with other road users. Defining the behavior of an agent in such scenarios is challenging due to the highly interactive decision-making; decisions depend on negotiations and actions of the other agents.

Our work [75] proposes a recurrent learning scheme for multi-agent scenarios in autonomous driving for efficient learning from bird's-eye-view (BEV) state representations as shown in Fig. 10 called RecurrDriveNet. Learning in an interactive decision process is challenging because the agents need to reason in a long-horizon manner while fixed-size state representation for varying numbers of agents in the observed range of the ego vehicle must be found. Typically, a BEV representation that encodes the other agents' states into an image with multiple channels is chosen.

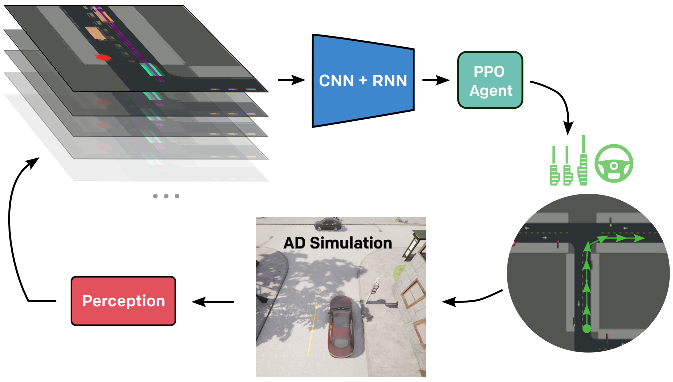


Figure 10: Our approach takes BEV images to learn driving policies in a simulated CARLA environment. Utilizing a CNN-based encoder while ensuring temporal understanding via an RNN, we learn longitudinal control using a PPO agent that predicts throttle and braking commands.

With autonomous driving (AD) as a simulated testbed in CARLA, we evaluated a novel recurrent architecture for efficient DRL in AD based on semantic BEV maps that minimizes collisions with other road users and respects traffic rules. Long-term decisions are especially challenging in highly interactive scenarios, *e.g.*, the autonomous vehicle (AV) must decide to yield to another vehicle and stick to that decision until the other vehicle has passed. While previous works typically used frame stacking, *i.e.*, creating a history of previous observations, as shown in Fig. 11, we demonstrated that our recurrent architecture (see Fig. 12 using a long short-term memory (LSTM) unit) is advantageous, leading to improved learning behavior.

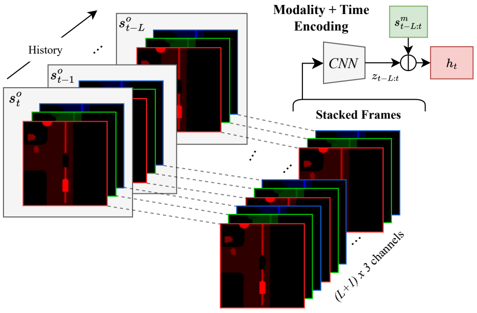


Figure 11: Frame-stacking of sequences.

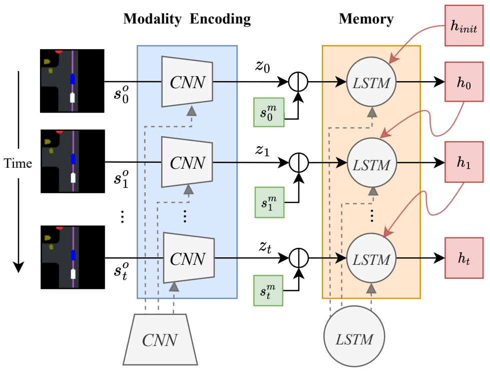


Figure 12: LSTM-based encoding of trajectories (ours).

The evaluation shows that our proposed RecurrDriveNet learns in one million steps to reduce infractions, *e.g.*, collisions with other vehicles or running red traffic lights, to a minimum. Our results also demonstrate that the agent's performance is improved when encoding the BEV image into separate channels per property (see Fig. 13), *i.e.*, the multi-channel BEV encodes map geometry, future waypoints, traffic lights, ego vehicle, other vehicles, and pedestrians separately, instead of creating an RGB representation as shown in Fig. 14.

RecurrDriveNet causes less than one infraction per driven kilometer by interacting safely with other road users. As previous frame stacking-based approaches do not converge to safe driving

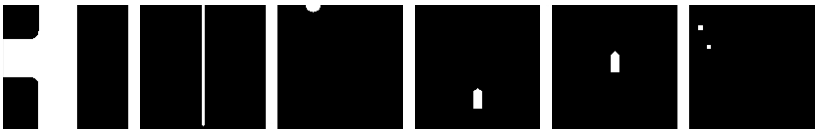


Figure 13: Multi-channel BEV (ours).

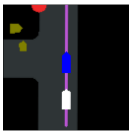


Figure 14: RGB BEV.

behavior in one million training steps, our work demonstrated the advantage of using recurrent architecture for efficient DRL in AD from BEV images.

4 Secure-by-construction Controller Synthesis in Safety- and Security-Critical Cyber-Physical Systems

Nowadays, many modern Cyber-Physical Systems are safety-critical in the sense that failure in these systems (*e.g.*, collision) may result in catastrophic consequences [53]. Meanwhile, these systems are also security-critical and more prone to various security threats and challenges [62, 43] due to the tight interaction and large information exchange between their cyber and physical components. In this year, we start investigating the notion of secure-by-construction controller synthesis, which aims at synthesizing controllers in a correct-by-construction manner while enforcing desired safety and security properties simultaneously. In this line of research, we have two main results this year. Firstly, by leveraging the notion of (augmented) control barrier functions, we proposed a secure-by-construction scheme to synthesize controllers that provide formal safety and security guarantees over invariance properties and initial state opacity properties. Based on these results, we further propose, for the first time, a secure-by-construction architecture, namely Safe-sec-visor architecture, for sandboxing unverified AI-based controllers while still ensuring that the systems are both safe and secure.

4.1 Secure-by-construction synthesis via (augmented) control barrier functions

Focusing on safety- and security-critical CPS, we develop a *secure-by-construction* scheme for controller synthesis. This scheme generalizes the existing *correct-by-construction controller synthesis* paradigm by integrating security requirements with safety ones in the controller synthesis framework. Here, we consider an important class of security property called *opacity* [40, 27], which is concerned with the information flow of the CPS. Essentially, opacity is a confidentiality property that captures the plausible deniability of a system for its secret behaviors in the presence of a malicious outside observer (intruder). In other words, a system is said to be opaque if the system's secrets cannot be revealed to an outside intruder based on the information flow. The concept of opacity was initially introduced in computer science literature [47] to study the performance of cryptographic protocols. Later on, various notions of opacity were proposed to capture different types of security requirements and information structures in discrete-event systems (DES) [40, 27, 6]. To the best of our knowledge, our work is the first result that introduces an *abstraction-free* scheme for synthesizing secure-by-construction controllers enforcing both safety and security properties *simultaneously* (*i.e.*, in *one-step*) over control systems with *continuous* state and input sets. To achieve this goal, we propose notions of (augmented) control barrier functions (referred to as ACBF and CBF), based on which we construct secure-by-construction controllers enforcing both properties. Moreover, we introduce conditions as sum-of-square (SOS) constraints under which the desired (augmented) control barrier functions can be constructed.

To showcase the effectiveness of the control synthesis scheme, we apply it to synthesize a secure-by-construction controller for a car moving on a single-lane road, which can be modeled as

$$\begin{aligned}\begin{bmatrix}x_1(k+1)\\x_2(k+1)\end{bmatrix} &= \begin{bmatrix}1 & \Delta\tau\\0 & 1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix} + \begin{bmatrix}\Delta\tau^2/2\\\Delta\tau\end{bmatrix}\nu(k),\\ y(k) &= x_1(k),\end{aligned} \tag{1}$$

in which $x = [x_1; x_2]$ is the state of the system, with x_1 and x_2 being its absolute position and velocity (in the road frame), respectively; $u \in [-3, 3]$ m/s^2 is the acceleration of the car that is used as the control input; $\Delta\tau = 0.1s$ is the sampling time; and y is the output of

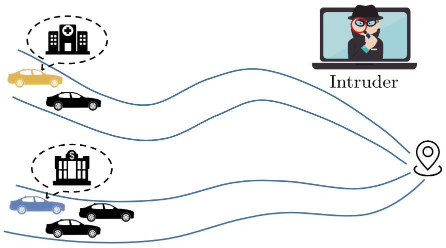


Figure 15: The plausible deniability of a car in terms of its secret initial conditions.

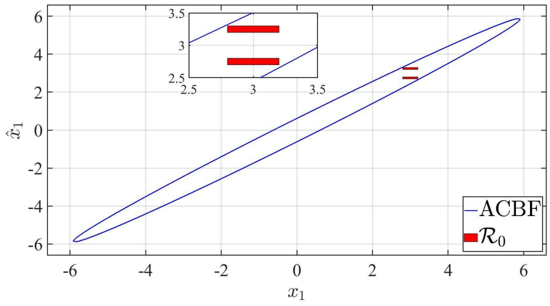


Figure 16: Cross section of the level set associated with the ACBF, with $x_2 = \hat{x}_2 = 0$.

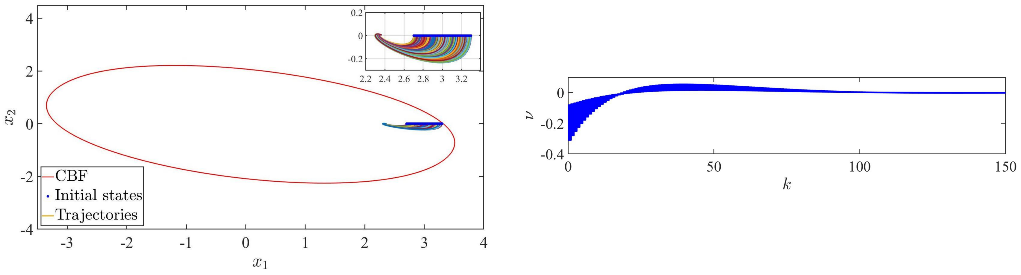


Figure 17: The level set associated with the CBF, trajectories of $x_1(k)$ and $x_2(k)$, and the corresponding input sequences $\nu(k)$.

the system that a malicious intruder can observe. Moreover, we are interested in the state set $X := [-10, 10] \times [-5, 5]$, initial set $X_0 := [2.7, 3.3] \times \{0\}$, secret set $X_s := [2.8, 3.2] \times \{0\}$, and unsafe set $X_d := ([-10, -5) \cup (6.5, 10]) \times ([-5, -3) \cup (3, 5])$. Accordingly, the safety specification requires that the position of the car should stay within the region $[-5, 6.5]$, while the absolute velocity should not exceed 3 m/s. The motivation of the desired security property is briefly explained as follows and depicted in Figure 15.

Consider a car moving on a single-lane road. We implicitly assume that the initial locations of the car contain certain secret information, where some confidential assignments might have been executed by the car. For example, the car might be a cash transit vehicle that transfers money initially from a bank to an ATM or a patient who was initially in a hospital but unwilling to leak private information to outsiders. Meanwhile, an intruder with δ measurement precision

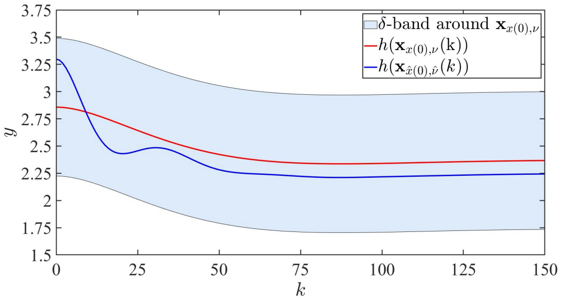


Figure 18: A state-run $\mathbf{x}_{x(0),\nu}$ of the system initiated from a secret location along with its equivalent (δ-close output) trajectory $\mathbf{x}_{\hat{x}(0),\hat{\nu}}$ started from a non-secret region, *i.e.*, $x(0) \in X_0 \cap X_s$ and $\hat{x}(0) \in X_0 \backslash X_s$. The shaded area in light blue is a δ-band around $\mathbf{x}_{x(0),\nu}$.

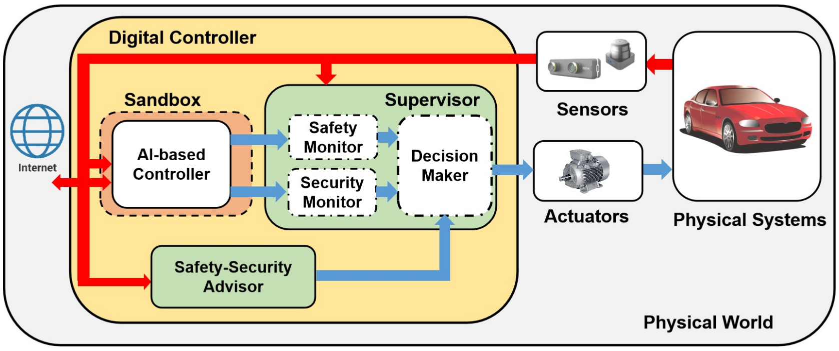


Figure 19: Safe-Sec-visor architecture for sandboxing AI-based controllers concerning both safety and security properties.

is observing the output (position) of the car remotely to infer whether the car initiated from a secret location. Hence, we aim to design a controller that enforces the system to avoid revealing its secret initial information to the intruder while ensuring safety.

Such a security property can be characterized by δ-approximate initial-state opacity, where $\delta \geq 0$ captures the security guarantee level in terms of the measurement precision of the intruder. Here, we select $\delta = 0.64$. Then, we obtain a CBF $\mathcal{B}(x)$ as in Figure 17 and an ACBF $\mathcal{B}_O(x, \hat{x})$ as in Figure 16. To validate the obtained CBF and ACBF, we randomly selected 1000 initial states $x(0)$ from the initial state set X_0 and simulated the system for 150 time steps. For those $x(0) \in X_0 \cap X_s$, we randomly selected $\hat{x}(0)$ such that $(x(0), \hat{x}(0)) \in \mathcal{R}_0$; for those $x(0) \in X_0 \backslash X_s$, we simply set $\hat{x}(0) = x(0)$ since the initial state itself does not contain any secret information. The trajectories of $x_1(k)$ and $x_2(k)$ and the corresponding input sequences ν are depicted in Figure 17, showing that the desired safety properties and input constraints are respected. Moreover, the desired 0.64-approximate initial-state opacity property is satisfied, since for each collected $x(0)$ and its corresponding trajectory $\mathbf{x}_{x(0),\nu}$, there exists $\hat{x}(0) \in X_0 \backslash X_s$ and trajectory $\mathbf{x}_{\hat{x}(0),\hat{\nu}}$, in which $\hat{\nu}(k) \in U$, such that $\|h(\mathbf{x}_{x(0),\nu}(k)) - h(\mathbf{x}_{\hat{x}(0),\hat{\nu}}(k))\| \leq 0.64$ holds for all $k \in [0, 150]$. Here, we depict in Figure 18 an example of such pair of trajectories $\mathbf{x}_{x(0),\nu}$ and $\mathbf{x}_{\hat{x}(0),\hat{\nu}}$.

4.2 Towards trustworthy AI via Safe-sec-visor architecture

Based on the secure-by-construction synthesis scheme discussed above, we now focus on those CPSs in which AI-based controllers are deployed. The past decades have witnessed remarkable achievements in artificial intelligence (AI) in many domains, such as natural language processing and image recognition. In the near future, plenty of AI-based controllers are also expected to be deployed in modern cyber-physical systems (CPSs) to accomplish complex control missions; typical scenarios include autonomous driving vehicles and smart buildings [55]. Nevertheless, verifying many AI-based controllers, particularly those developed based on deep neural networks, is a challenging task shown to be nondeterministic polynomial-time complete (NP-complete) in general [19]. The lack of verifying AI-based controllers might lead to disastrous consequences in real-life CPSs regarding safety and security concerns.

Here, we aim at providing formal guarantees regarding safety and security *simultaneously* over those CPSs without formally verifying the AI-based controllers. Concretely, inspired by the results in [86, 83], we propose a new architecture utilizing the idea of *sandbox* [60], namely *Safe-Sec-visor architecture* (see Figure 19), for sandboxing AI-based unverified controllers. In particular, the proposed architecture consists of a safety-security advisor and a supervisor that contains a safety monitor and a security monitor. At run-time, the supervisor decides whether to deploy the AI-based controller to control the system based on the decision of the safety and security monitors. The safety monitor rejects the AI-based controller whenever it endangers the overall safety of the system. Similarly, the security monitor is in charge of rejecting those control inputs from the AI-based controllers that would result in a violation of the desired security properties. If the AI-based controller is rejected, the safety-security advisor is responsible for ensuring the overall safety and security of the system. Note that the safety-security advisor is supposed to be deployed as less as possible since it *only* focuses on keeping the system safe and secure, and we, therefore, need to exploit the functionalities offered by the AI-based controllers. To the best of our knowledge, we are the first who introduce an architecture for sandboxing AI-based unverified controllers to provide formal guarantees regarding safety and security properties *simultaneously* over control systems with *continuous* state and input sets. Here, it is also worth mentioning those existing results (*e.g.*, [32, 35]) in which formal guarantees are achieved by appropriately incorporating the desired objectives in the reward functions when training AI-based controllers. Compared with these results, the desired safety and security properties are decoupled from the construction of the AI-based controllers in our proposed architecture. Therefore, our architecture can provide formal guarantees for any type of AI-based controllers regardless of their design.

To demonstrate the new architecture, we consider a case study in which a quadrotor tracks a series of changing targets on a 2-dimensional plane (x-y plane), as shown in Figure 20. Meanwhile, the positions of the quadrotor are observed by a malicious intruder, with an observation precision $\delta = 2$. Here, it is not desired to reveal to the intruder whether the quadrotor starts from the secret region X_s, since a quadrotor starting from the secret region X_s indicates that the quadrotor belongs to a company and operates a special task. At the same time, due to air traffic regulations, the quadrotor is required to stay within a safety region X_{safe}.

By leveraging the feedback linearization technique in [23], the quadrotor can be modeled by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k), \quad k \in \mathbb{N}, \end{aligned}$$

with

$$A := \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, B := \begin{bmatrix} \Delta t^2/2 \\ \Delta t \\ \Delta t^2/2 \\ \Delta t \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

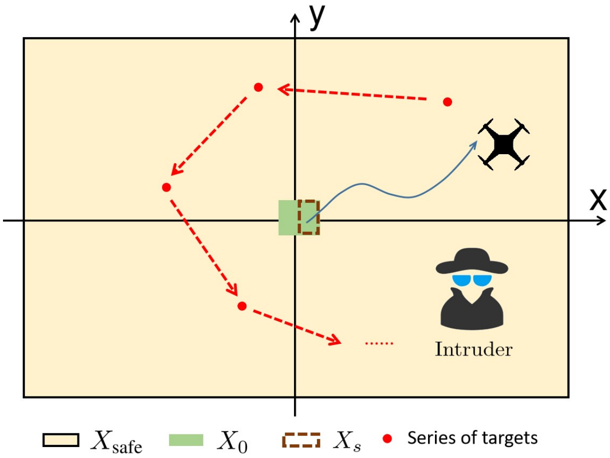


Figure 20: A quadrotor tracks a series of targets. Meanwhile, the positions of the quadrotor are remotely observed by a malicious intruder.

	With Safe-Sec-visor architecture	Without Safe-Sec-visor architecture
Percentages of satisfying the safety properties	100%	0%
Percentages of satisfying the opacity properties	100%	71.66%

Table 1: Simulation results with and without using the Safe-Sec-visor architecture.

Here, $\Delta t = 0.1$s is the sampling time; $x := [x_{\mathsf{x}}; v_{\mathsf{x}}; x_{\mathsf{y}}; v_{\mathsf{y}}]$ and $u := [u_{\mathsf{x}}; u_{\mathsf{y}}]$ denote the state and the control input of the quadrotor, respectively, with x_{i}, v_{i}, and u_{i} being the position, velocity, and acceleration of the drone on the i axis, $\mathsf{i} \in \{\mathsf{x}, \mathsf{y}\}$, respectively; y is the output of the systems, which can be remotely observed by a malicious intruder. Here, we consider the state set $X := [-120, 120] \times [-5, 5] \times [-120, 120] \times [-5, 5]$, initial set $X_0 := \big([-0.5, 0] \times \{0\} \times [-0.05, 0.05] \times \{0\}\big) \cup \big([0, 0.5] \times [-0.85, 0.85] \times [-0.3, 0.3] \times [-0.85, 0.85]\big)$, secret set $X_s := [0, 0.5] \times [-0.85, 0.85] \times [-0.3, 0.3] \times [-0.85, 0.85]$, and unsafe set $X_d := X \setminus \big([-100, 100] \times [-3, 3] \times [-100, 100] \times [-3, 3]\big)$ (note that $X_d := X \setminus X_{\text{safe}}$). In brief, the position of the quadrotor $(x_{\mathsf{x}}, x_{\mathsf{y}})$ is required to be within the region $[-100, 100] \times [-100, 100]$, while the velocity of the quadrotor cannot exceed 3 m/s on both axis. Moreover, we consider the constraints for the control inputs as $u_{\mathsf{i}} \in [-2, 2]$ m/s^2, with $\mathsf{i} \in \{\mathsf{x}, \mathsf{y}\}$.

As for the AI-based controller, we deploy one that is supposed to enforce the quadrotor tracking a series of changing targets. Here, the AI-based controller contains a deep-neural-network-based (DNNs-based) agent, which is trained by leveraging DDPG algorithm [41] and works as a setpoint provider for low-level position controller, with $K = [1.4781; 1.7309]^\top$. This agent takes the desired target, the current positions and velocities of the quadrotor as inputs and provides the position and velocity setpoints for the quadrotor. Note that we are not describing how to train the agent here since designing and improving the performance of AI-based unverified controllers are out of the scope of this paper. The AI-based controller deployed here is only for demonstration purposes. In particular, our architecture can be deployed to any “off-the-shelf” AI-based controller regardless of their performance, while a formal guarantee for ensuring the desired safety and security properties can still be provided.

For the simulation, we randomly selected 300 initial states $x(0)$ from the set $X_0 \cap X_s$ and

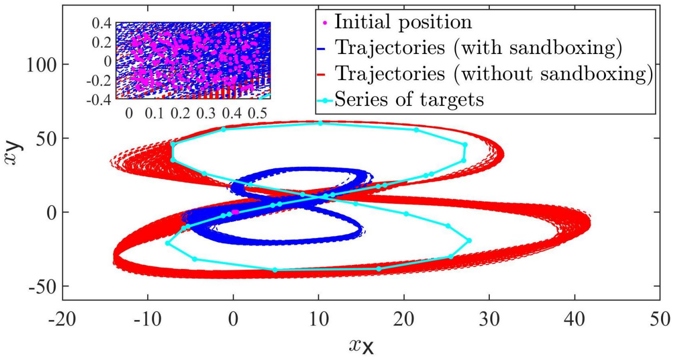


Figure 21: Trajectories of the quadrotor's positions with and without using the Safe-Sec-visor architecture.

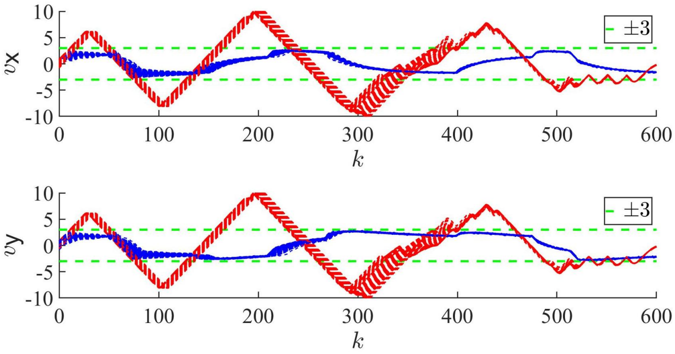


Figure 22: Trajectories of the quadrotor's velocities with (blue) and without using the Safe-Sec-visor architecture (red).

simulated the system for 600 time steps (we are not considering those initial states in the set $X_0 \setminus X_s$ since the initial state there does not contain any secret information so that the desired initial-state opacity is satisfied trivially). The simulation results are summarized in Table 1 and depicted in Figures 21 and 22. When deploying the Safe-Sec-visor architecture, 30.78% of the control input provided by the AI-based controllers are accepted, while the desired safety and initial-state opacity properties are satisfied. In contrast, when the Safe-Sec-visor architecture is not deployed to sandbox the AI-based controller, all the trajectories violate the desired safety properties. Additionally, 28.34% of the trajectories reveal the fact that the quadrotor starts from the secret region. Here, we demonstrate one such trajectory in Figure 23 in which the reachable sets are computed using `MPT3` [30].

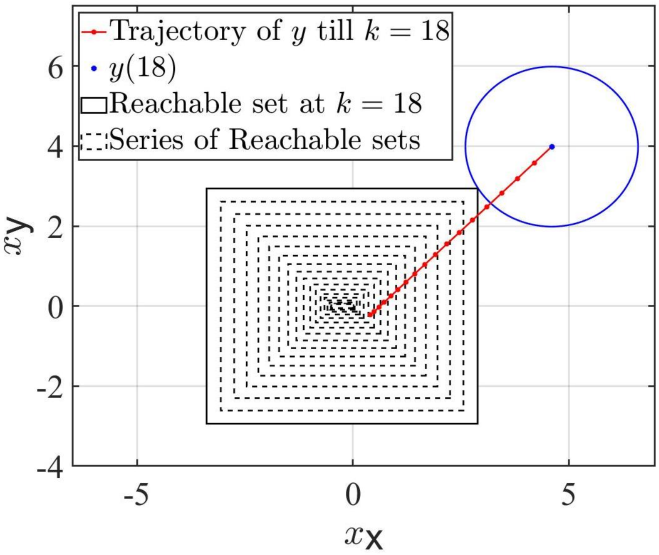


Figure 23: A trajectory of the system without using the Safe-Sec-visor architecture. This trajectory reveals that the quadrotor started from the secrete region. Concretely, the distance between y at time step $k = 18$ and the k-step reachable sets of the quadrotor is larger than $\delta = 2$ (the radius of the blue circle is 2).

5 Computer Vision for Robotic Manipulation

Vision-based perception provides promise for robots to understand working scenarios in complex manipulation tasks and thereby enhancing overall task success rates. However, training learning-based computer vision components need a huge amount of labeled data, which is expensive regarding time and labor cost. Our research leverages modern simulators to generate synthetic data for training. We aim to systematically examine the efficacy of synthetic training paradigms in preparing robotic systems for real-world manipulation tasks.

5.1 6D Pose Estimation for Robotic Grasping [12]

In the past year, significant progress has been made in the field of 6D pose estimation for robotic manipulation. Building on the existing foundation, the Chair of Cyber-Physical Systems in Production Engineering developed an integrated baseline 6D pose vision system and a Blender pipeline for generating synthetic data. These advancements have facilitated new methods for robotic grasping and manipulation.

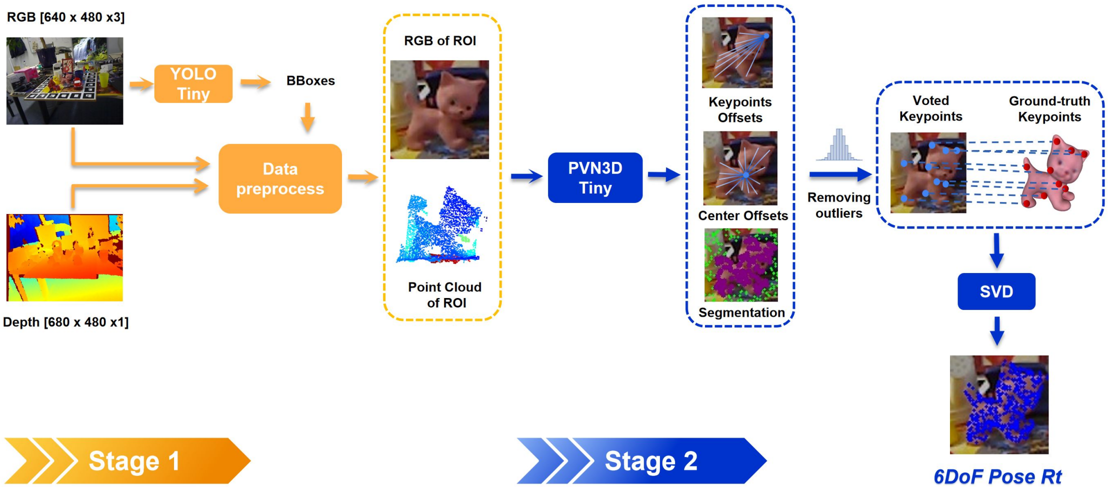


Figure 24: A two-stage pose estimation approach: object detection with YOLO-tiny followed by the 6D object pose estimation with PVN3D-tiny at the second stage.

Key highlights of the year's work include:

- Development of an Integrated 6D Pose Estimation System: An efficient and effective system for 6D pose estimation was developed, integrating PVN3D with YOLO and optimized for real-time performance. This system is crucial for reactive robotic manipulation applications.
- Synthetic Data Generation Pipeline: A pipeline using Blender was created to generate large amounts of photorealistic RGBD images for objects of interest, some examples are shown in 25. This pipeline includes various domain randomization techniques to minimize the gap between synthetic and real data.
- Robotic Testbed and Experiments: A robotic testbed was completed, and experiments were conducted to validate the pose estimation system under different lighting conditions.

The experiments involved 750 grasping attempts on various objects, achieving a success rate of 87% (or 93% when excluding collisions not related to pose estimation).

- 6IMPOSE Framework: The ”6IMPOSE” paper details the methodologies and technologies developed. The framework demonstrates the robustness of the system in various challenging scenarios.

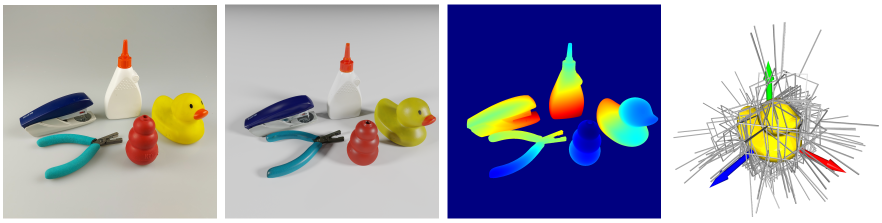
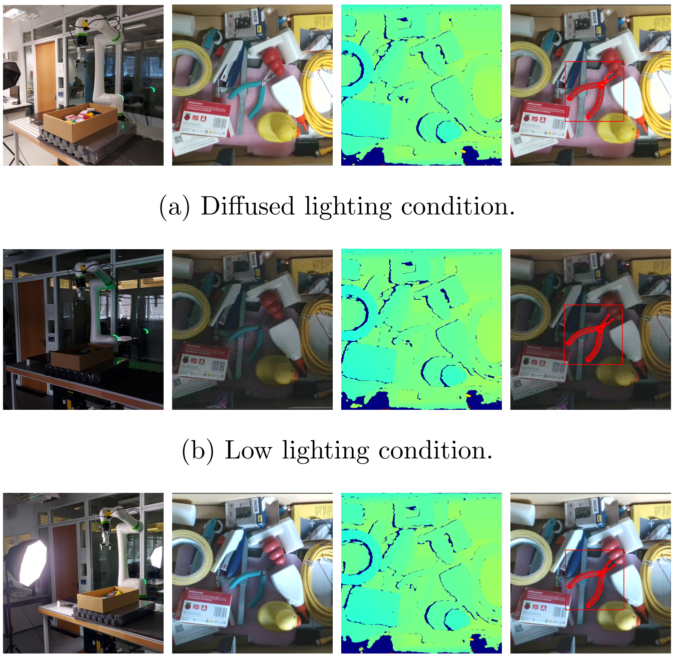


Figure 25: From left to right: the five selected objects photographed, photo-realistically rendered, depth rendered and automatically generated grasp poses for the duck.

These advancements represent a significant step forward in the application of 6D pose estimation in robotics, particularly in the context of real-world scenarios and varying environmental conditions, as shown in Fig. 26.



(a) Diffused lighting condition.

(b) Low lighting condition.

(c) Spot lighting condition.

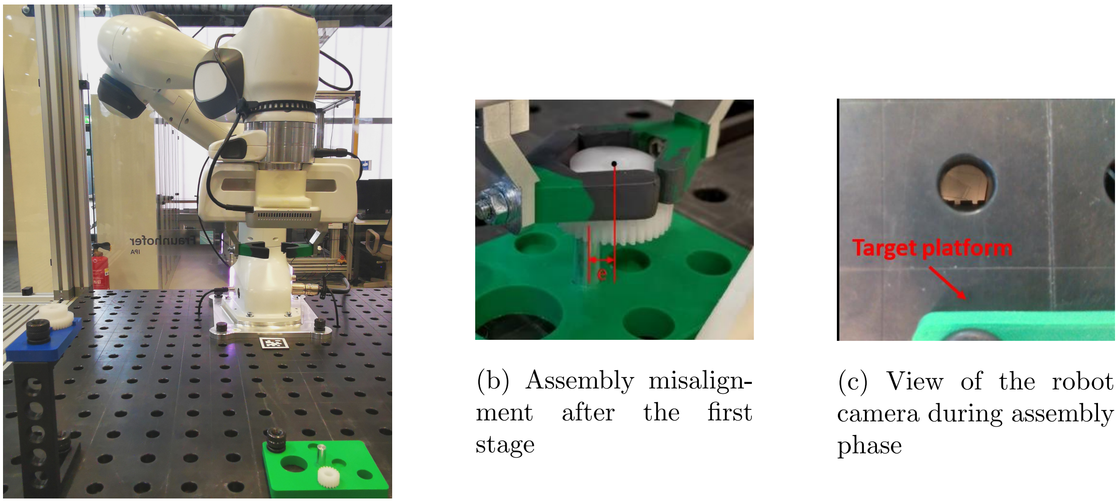
Figure 26: The figure showing the real world grasping experiments under diffused lighting conditions. The experimental setup, RGB view, depth view and the predicted pose of pliers on RGB are shown from the left to the right.

5.2 Vision-Guided Policy for Gear Assembly [48]

In recent years, industrial robotic automation has witnessed many achievements. Nevertheless, a report conducted in North America and Europe suggests that only about 6 % of robotic operational stocks are deployed for assembly tasks. Many assembly operations require high precision to avoid damaging the assembled parts. One such high-precision task is the assembly and meshing of gears. These components are used in many products, *e.g.*, electronics, automotive, machining, etc. In general, gear assembly can be classified as an extended insertion problem, representing 35 % of assembly tasks, but with the additional challenge of aligning the gear teeth, where high precision and flexibility are essential.

Typically, a gear assembly process involves searching and insertion. To complete these two procedures, conventional gear assembly relies on tedious position teaching, which has the limitations of time cost and low flexibility. Recent work leverages visual information and force/torque (FT) feedback in robotic assembly to avoid the reliance on tedious position teaching, achieving promising performance.

In this work, we aim to develop an efficient solution to realize the gear searching and insertion for high precision and flexible gear assembly tasks, as shown in Fig. 27. The assembly scenario consists of a placement tray with an initial gear, a Frank Emika Panda Robot, and a target platform mounted with a gear near the peg to be assembled. The robot has an RGBD camera on the wrist to observe the environment and a force sensor on the gripper to detect contact. This assembly aims to grasp a gear from the placement tray and insert it onto the peg to finish the



(a) Task setting

(b) Assembly misalignment after the first stage

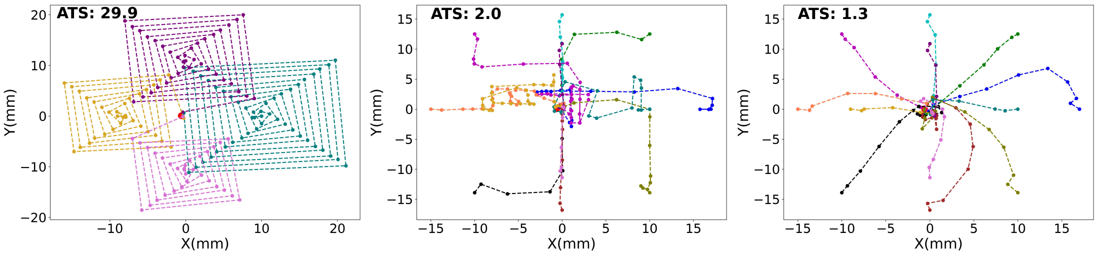
(c) View of the robot camera during assembly phase

Figure 27: The figure provides an overview of the gear assembly task, with the objective of grasping a gear from the placement tray, observing the environment using the on-wrist RGBD camera, and inserting the gear held by the robot onto the peg using visual servoing with force feedback to complete the assembly process.

assembly with the gear on the platform under a clearance requirement of 0.3 mm. The clearance is defined as the distance between the assembled peg and gear that enables a free rotation of the gear on the peg, thus transmitting forces. We simplify the grasping by initializing the gear with a known position on the placement tray and programming a fixed motion to grasp the gear as we mainly focus on the gear assembly.

We propose a vision-guided two-stage approach with force feedback to solve the above gear assembly problem. Specifically, in the first stage, we use YOLO with depth information to roughly localize the object. With the coarse position, we can efficiently drive the robot using position control to reach the neighbor of the peg. As shown in Fig. 27-(b), after the first stage, the centers of the gear and peg are not aligned perfectly due to the perception error. Meanwhile, the camera loses the direct view of the peg, which imposes the challenge of obtaining the relative position of the peg. To address this problem, we propose a novel solution by training DRL agents to learn the underlying relative positions from partially visible target platforms in RGB images, as shown in Fig. 27-(c). Through interaction with the environment, DRL learns to output movement setpoints for position control to complete the insertion. During the whole pipeline, a force sensor is used to improve the robustness of the vision-guided assembly process. Specifically, we use force feedback to determine whether the gear touches the peg, which initiates the transition between the first and second stages. The force feedback also ensures the contact between the gear and peg for finishing the insertion. Furthermore, We explore DQN [50] algorithm with discretized action space and PPO [64] with continuous action space for the insertion to study the potential of the proposed method.

Training deep neural networks requires a large amount of data. To reduce the data collection effort on real robots, we adopt a sim-to-real approach to train YOLO, where the network is trained solely using synthetic data with domain randomizations and directly transferred to the real world. We pre-train the DRL agents in an offline interaction environment constructed using sampled



(a) Spiral searching

(a) Real-world tuned DQN

(a) Real-world tuned PPO

Figure 28: The figure shows the trajectories of gear assembly from various initial starting points using different methods, where PPO achieves the most efficient performance with an average traveling score (ATS) of 1.3.

real-world data, and continually fine-tune in the real world to mitigate domain gap. The real-world experiments show that the proposed gear assembly approach with the proposed training pipeline can achieve high robustness and efficiency when tested in an industrial gear assembly task from arbitrary initial positions. The visualization of the trajectories of the proposed approach and the baseline is shown in Fig. 28.

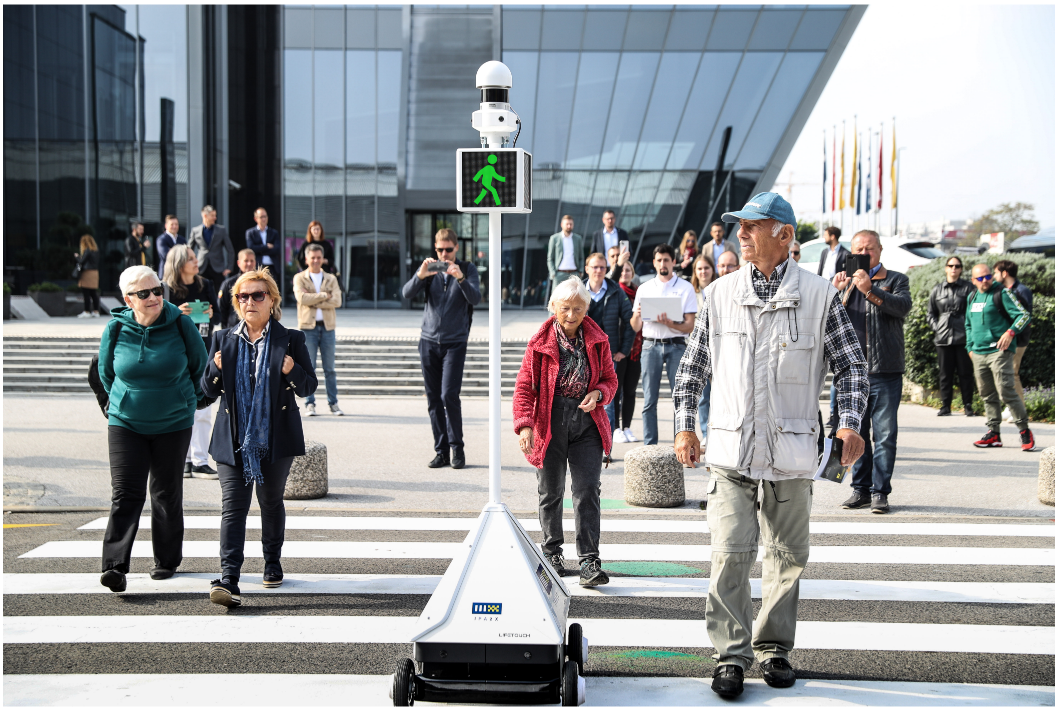


Figure 29: IPA2X Robot during Ljubljana demonstration.

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

The Chair has been leading the EIT European Project ”IPA2X” (www.ipa2x.eu). The project was executed during 2022 and finalized during 2023. Final activities for the project were completed in September 2023.

The project has been 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 was co-funded by EIT and had 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) communication.

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

Management

Prof. Dr. Marco Caccamo, Director
`www.mw.tum.de/cps`
mcaccamo@tum.de
Tel: +49 89 289 55170

Administrative Staff

Anke Harisch, Secretary

Research Scientists

- 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.
- Andres Zapata Rodriguez (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 and Robco Modular Robot Arm
- Unitree A1 Quadruped Explorer

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, Toulouse, France
- Université Paris Cité, France
- Nantes Université, France
- University of Freiburg, Germany

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 2023

Journal

- Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. Formal synthesis of controllers for uncertain linear systems against omega-regular properties: A set-based approach. *IEEE Transactions on Automatic Control*, 2023
- Hongpeng Cao, Lukas Dirnberger, Daniele Bernardini, Cristina Piazza, and Marco Caccamo. 6impose: bridging the reality gap in 6d pose estimation for robotic grasping. *Frontiers in Robotics and AI*, 10:1176492, 2023
- Lucas Matheus dos Santos, Giovani Gracioli, Tomasz Kloda, and Marco Caccamo. Supporting single and multi-core resource access protocols on object-oriented RTOSes. *Journal of Design Automation for Embedded Systems*, pages 31–50, 2023
- Mohammed Foughali, Pierre-Emmanuel Hladik, and Alexander Zuepke. Compositional verification of embedded real-time systems. *Journal of Systems Architecture*, 142:102928, 2023

Conference proceeding

- Bingzhuo Zhong, Siyuan Liu, Marco Caccamo, and Majid Zamani. Secure-by-construction controller synthesis via control barrier functions. *IFAC-PapersOnLine*, 56(2):239–245, 2023
- Bingzhuo Zhong, Siyuan Liu, Marco Caccamo, and Majid Zamani. Towards trustworthy ai: Sandboxing ai-based unverified controllers for safe and secure cyber-physical systems. In *Proceedings of 62nd IEEE Conference on Decision and Control(CDC)*, 2023
- Binqi Sun, Debayan Roy, Tomasz Kloda, Andrea Bastoni, Rodolfo Pellizzoni, and Marco Caccamo. Co-optimizing cache partitioning and multi-core task scheduling: Exploit cache sensitivity or not? In *IEEE Real-Time Systems Symposium (RTSS)*, 2023

- Binqi Sun, Tomasz Kloda, Sergio Arribas Garcia, Giovani Gracioli, and Marco Caccamo. Minimizing cache usage for real-time systems. In *Proceedings of the 31st International Conference on Real-Time Networks and Systems (RTNS)*, pages 200–211, 2023

- Binqi Sun, Tomasz Kloda, Jiyang Chen, Cen Lu, and Marco Caccamo. Schedulability analysis of non-preemptive sporadic gang tasks on hardware accelerators. In *IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 147–160, 2023

- Andrea Misuraca and Andrea Bastoni. Arm much: Full-spectrum hardware-event-based armv8 application profiler. In *The 17th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, pages 19–24, 2023

- Hazem Abaza, Abhinaba Habishyashi, Debayan Roy, Andrea Bastoni, Zain A. H. Hammadeh, Shiqing Fan, Selma Saidi, and Sergey Tverdyshev. Rdma-based deterministic communication architecture for autonomous driving. In *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 137–146, 2023

- Ju Hyoung Mun, Konstantinos Karatsenidis, Tarikul Islam Papon, Shahin Roozkhosh, Denis Hoornaert, Ulrich Drepper, Ahmed Sanaullah, Renato Mancuso, and Manos Athanassoulis. On-the-fly data transformation in action. *Proc. VLDB Endow.*, 16(12):3950–3953, aug 2023

- Sergio Arribas García, Giovani Gracioli, Denis Hoornaert, Tomasz Kloda, and Marco Caccamo. Improving the execution time of industrial applications through planned cache eviction policy selection. In *IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1–6, 2023

- Renato Mancuso, Shahin Roozkhosh, Denis Hoornaert, Ju Hyoung Mun, Tarikul Islam Papon, and Manos Athanassoulis. Software-shaped platforms. In *Proceedings of Cyber-Physical Systems and Internet of Things Week*, CPS-IoT Week '23, page 185–191, New York, NY, USA, 2023. Association for Computing Machinery

- Tarikul Islam Papon, Ju Hyoung Mun, Shahin Roozkhosh, Denis Hoornaert, Ahmed Sanaullah, Ulrich Drepper, Renato Mancuso, and Manos Athanassoulis. Relational fabric: Transparent data transformation. In *IEEE International Conference on Data Engineering (ICDE)*, pages 3688–3698, 2023

- Weifan Chen, Ivan Izhbirdeev, Denis Hoornaert, Shahin Roozkhosh, Patrick Carpanedo, Sanskriti Sharma, and Renato Mancuso. Low-Overhead Online Assessment of Timely Progress as a System Commodity. In *35th Euromicro Conference on Real-Time Systems (ECRTS)*, volume 262, pages 13:1–13:26, Dagstuhl, Germany, 2023

- Ahsan Saeed, Denis Hoornaert, Dakshina Dasari, Dirk Ziegenbein, Daniel Mueller-Gritschneder, Ulf Schlichtmann, Andreas Gerstlauer, and Renato Mancuso. Memory Latency Distribution-Driven Regulation for Temporal Isolation in MPSoCs. In *35th Euromicro Conference on Real-Time Systems (ECRTS)*, volume 262, pages 4:1–4:23, Dagstuhl, Germany, 2023

- Jichao Chen, Omid Esrafilian, Harald Bayerlein, David Gesbert, and Marco Caccamo. Model-aided federated reinforcement learning for multi-UAV trajectory planning in IoT networks. In *IEEE Global Communications Conference (GLOBECOM) Workshops*. IEEE, 2023

- Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-model-regulated deep reinforcement learning towards safety and stability guarantees. In *IEEE Conference on Decision and Control (CDC)*. IEEE

- Junjie Ming, Daniel Bargmann, Hongpeng Cao, and Marco Caccamo. Flexible gear assembly with visual servoing and force feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023

- 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)*, 2023

- Raphael Trumpp, Denis Hoornaert, and Marco Caccamo. Residual policy learning for vehicle control of autonomous racing cars. *IEEE Intelligent Vehicles Symposium (IV)*, 2023

- Raphael Trumpp, Martin Büchner, Abhinav Valada, and Marco Caccamo. Efficient learning of urban driving policies using bird's-eye-view state representations. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023

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 [65]). 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] Edge TPU. `https://cloud.google.com/edge-tpu`. Accessed: 2022-10-25.

[2] Edge TPU Performance Benchmarks. `https://coral.ai/docs/edgetpu/benchmarks`. accessed: 2022-10-25.

[3] Hazem Abaza, Abhinaba Habishyashi, Debayan Roy, Andrea Bastoni, Zain A. H. Hammadeh, Shiqing Fan, Selma Saidi, and Sergey Tverdyshev. Rdma-based deterministic communication architecture for autonomous driving. In *2023 IEEE 29th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 137–146, 2023.

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

[5] Sebastian Altmeyer, Roeland Douma, Will Lunniss, and Robert I. Davis. On the Effectiveness of Cache Partitioning in Hard Real-Time Systems. *Real-Time Systems*, 52(5):598–643, 2016.

[6] J. Balun and T. Masopust. Comparing the notions of opacity for discrete-event systems. *Discrete Event Dynamic Systems*, 31(4):553–582, 2021.

[7] A. Bastoni, B. B. Brandenburg, and J. H. Anderson. An empirical comparison of global, partitioned, and clustered multiprocessor EDF schedulers. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 14–24, 2010.

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

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

[10] Brice Berna and Isabelle Puaut. PDPA: Period driven task and cache partitioning algorithm for multi-core systems. In *International Conference on Real-Time and Network Systems (RTNS)*, pages 181–189, 2012.

[11] Axel Brunnbauer, Luigi Berducci, Andreas Brandstátter, Mathias Lechner, Ramin Hasani, Daniela Rus, and Radu Grosu. Latent imagination facilitates zero-shot transfer in autonomous racing. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7513–7520. IEEE, 2022.

[12] Hongpeng Cao, Lukas Dirnberger, Daniele Bernardini, Cristina Piazza, and Marco Caccamo. 6impose: bridging the reality gap in 6d pose estimation for robotic grasping. *Frontiers in Robotics and AI*, 10:1176492, 2023.

[13] Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-model-regulated deep reinforcement learning towards safety and stability guarantees. In *IEEE Conference on Decision and Control (CDC)*. IEEE.

[14] Jichao Chen, Omid Esrafilian, Harald Bayerlein, David Gesbert, and Marco Caccamo. Model-aided federated reinforcement learning for multi-UAV trajectory planning in IoT networks. In *IEEE Global Communications Conference (GLOBECOM) Workshops*. IEEE, 2023.

[15] Weifan Chen, Ivan Izhbirdeev, Denis Hoornaert, Shahin Roozkhosh, Patrick Carpanedo, Sanskriti Sharma, and Renato Mancuso. Low-Overhead Online Assessment of Timely Progress as a System Commodity. In *35th Euromicro Conference on Real-Time Systems (ECRTS)*, volume 262, pages 13:1–13:26, Dagstuhl, Germany, 2023.

[16] Szegedy Christian, Vanhoucke Vincent, Sergey Ioffe, Shlens Jon, and Wojna Zbigniew. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[17] R Craig Coulter. Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, 1992.

[18] Lucas Matheus dos Santos, Giovani Gracioli, Tomasz Kloda, and Marco Caccamo. Supporting single and multi-core resource access protocols on object-oriented RTOSes. *Journal of Design Automation for Embedded Systems*, pages 31–50, 2023.

[19] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

[20] Dror G. Feitelson and Larry Rudolph. Gang scheduling performance benefits for fine-grain synchronization. *Journal of Parallel and Distributed Computing*, 16(4):306–318, 1992.

[21] Mohammed Foughali, Pierre-Emmanuel Hladik, and Alexander Zuepke. Compositional verification of embedded real-time systems. *Journal of Systems Architecture*, 142:102928, 2023.

[22] Sergio Arribas García, Giovani Gracioli, Denis Hoornaert, Tomasz Kloda, and Marco Caccamo. Improving the execution time of industrial applications through planned cache eviction policy selection. In *IEEE International Symposium on Industrial Electronics (ISIE)*, pages 1–6, 2023.

[23] Azad Ghaffari. Analytical design and experimental verification of geofencing control for aerial applications. *IEEE/ASME Transactions on Mechatronics*, 26:1106–1117, 2021.

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

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

[26] G. Gracioli, A. Alhammad, R. Mancuso, A. A. Fröhlich, and R. Pellizzoni. A Survey on Cache Management Mechanisms for Real-Time Embedded Systems. *ACM Computing Surveys*, 48(2), 2015.

[27] C.N. Hadjicostis. *Estimation and Inference in Discrete Event Systems*. Springer, 2020.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[29] John L. Hennessy and David A. Patterson. A New Golden Age for Computer Architecture. *Communications of the ACM*, 62(2):48–60, 2019.

[30] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari. Multi-parametric toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. `http://control.ee.ethz.ch/~mpt`.

[31] 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.

[32] Subin Huh and Insoon Yang. Safe reinforcement learning for probabilistic reachability and safety specifications: A Lyapunov-based approach. *arXiv:2002.10126*, 2020.

[33] Intel. Improving real-time performance by utilizing cache allocation technology. white paper. Technical report, Apr 2015.

[34] Morris A. Jette. Performance characteristics of gang scheduling in multiprogrammed environments. In *ACM/IEEE Conference on Supercomputing*, pages 1–12, 1997.

[35] Milad Kazemi and Sadegh Soudjani. Formal policy synthesis for continuous-state systems via reinforcement learning. In *International Conference on Integrated Formal Methods*, pages 3–21. Springer, 2020.

[36] David Blair Kirk. SMART (strategic memory allocation for real-time) cache design. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 229–237, 1989.

[37] 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.

[38] 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 *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–14, 2019.

[39] Tomasz Kloda, Marco Solieri, Renato Mancuso, Nicola Capodieci, Paolo Valente, and Marko Bertogna. Deterministic memory hierarchy and virtualization for modern multi-core embedded systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–14, 2019.

[40] S. Lafortune, F. Lin, and C.N. Hadjicostis. On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45:257–266, 2018.

[41] 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 *International Conference on Learning Representations(Poster)*, 2016.

[42] ARM Limited. Primecell level 2 cache controller (pl310) technical reference manual. Technical report, 2008.

[43] S. Liu, A. Trivedi, X. Yin, and M. Zamani. Secure-by-construction synthesis of cyber-physical systems. *Annual Reviews in Control*, 53:30–50, 2022.

[44] 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.

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

[46] Renato Mancuso, Shahin Roozkhosh, Denis Hoornaert, Ju Hyoung Mun, Tarikul Islam Papon, and Manos Athanassoulis. Software-shaped platforms. In *Proceedings of Cyber-Physical Systems and Internet of Things Week*, CPS-IoT Week '23, page 185–191, New York, NY, USA, 2023. Association for Computing Machinery.

[47] L. Mazaré. Using unification for opacity properties. In *Workshop on Issues in the Theory of Security*, volume 4, pages 165–176, 2004.

[48] Junjie Ming, Daniel Bargmann, Hongpeng Cao, and Marco Caccamo. Flexible gear assembly with visual servoing and force feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[49] Andrea Misuraca and Andrea Bastoni. Arm much: Full-spectrum hardware-event-based armv8 application profiler. In *The 17th Annual Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, pages 19–24, 2023.

[50] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[51] Ju Hyoung Mun, Konstantinos Karatsenidis, Tarikul Islam Papon, Shahin Roozkhosh, Denis Hoornaert, Ulrich Drepper, Ahmed Sanaullah, Renato Mancuso, and Manos Athanassoulis. On-the-fly data transformation in action. *Proc. VLDB Endow.*, 16(12):3950–3953, aug 2023.

[52] Laurence H. Mutuel, Xavier Jean, Vincent Brindejonc, Anthony Roger, Thomas Megel, and E. Alepins. Assurance of Multicore Processors in Airborne Systems. Technical Report DOT/FAA/TC-16/51, FAA and Thales Avionics, 2017.

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

[54] Matthew O'Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. In *NeurIPS 2019 Competition and Demonstration Track*, pages 77–89. PMLR, 2020.

[55] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[56] Marco Paolieri, Eduardo Quiñones, Francisco J. Cazorla, Robert I. Davis, and Mateo Valero. IA^3: An interference aware allocation algorithm for multicore hard real-time systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 280–290, 2011.

[57] Tarikul Islam Papon, Ju Hyoung Mun, Shahin Roozkhosh, Denis Hoornaert, Ahmed Sanaullah, Ulrich Drepper, Renato Mancuso, and Manos Athanassoulis. Relational fabric: Transparent data transformation. In *IEEE International Conference on Data Engineering (ICDE)*, pages 3688–3698, 2023.

[58] Xen Project. Xen Project Public Repository. http://xenbits.xen.org/gitweb/?p=xen.git.

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

[60] Charles Reis, Adam Barth, and Carlos Pizano. Browser security: lessons from google chrome. *Communications of the ACM*, 52(8):45–49, 2009.

[61] Ahsan Saeed, Denis Hoornaert, Dakshina Dasari, Dirk Ziegenbein, Daniel Mueller-Gritschneder, Ulf Schlichtmann, Andreas Gerstlauer, and Renato Mancuso. Memory Latency Distribution-Driven Regulation for Temporal Isolation in MPSoCs. In *35th Euromicro Conference on Real-Time Systems (ECRTS)*, volume 262, pages 4:1–4:23, Dagstuhl, Germany, 2023.

[62] H. Sandberg, S. Amin, and K. Johansson. Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems Magazine*, 35(1):20–23, 2015.

[63] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint*, 2017.

[64] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[65] 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.

[66] M. Shekhar, A. Sarkar, H. Ramaprasad, and F. Mueller. Semi-partitioned hard-real-time scheduling under locked cache migration in multicore systems. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 331–340, 2012.

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

[68] Binqi Sun, Tomasz Kloda, Sergio Arribas Garcia, Giovani Gracioli, and Marco Caccamo. Minimizing cache usage for real-time systems. In *Proceedings of the 31st International Conference on Real-Time Networks and Systems (RTNS)*, pages 200–211, 2023.

[69] Binqi Sun, Tomasz Kloda, Jiyang Chen, Cen Lu, and Marco Caccamo. Schedulability analysis of non-preemptive sporadic gang tasks on hardware accelerators. In *IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 147–160, 2023.

[70] Binqi Sun, Debayan Roy, Tomasz Kloda, Andrea Bastoni, Rodolfo Pellizzoni, and Marco Caccamo. Co-optimizing cache partitioning and multi-core task scheduling: Exploit cache sensitivity or not? In *IEEE Real-Time Systems Symposium (RTSS)*, 2023.

[71] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.

[72] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[73] R. Tabish, R. Mancuso, S. Wasly, R. Pellizzoni, and M. Caccamo. A real-time scratchpad-centric OS with predictable inter/intra-core communication for multi-core embedded systems. *Real-Time Systems*, 55:850–888, 2019.

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

[75] Raphael Trumpp, Martin Büchner, Abhinav Valada, and Marco Caccamo. Efficient learning of urban driving policies using bird's-eye-view state representations. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023.

[76] Raphael Trumpp, Denis Hoornaert, and Marco Caccamo. Residual policy learning for vehicle control of autonomous racing cars. *IEEE Intelligent Vehicles Symposium (IV)*, 2023.

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

[78] Sergi Vilardell, Isabel Serra, Enrico Mezzetti, Jaume Abella, and Francisco J. Cazorla. Much: Exploiting pairwise hardware event monitor correlations for improved timing analysis of complex mpsocs. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, SAC '21, page 511–520, New York, NY, USA, 2021. Association for Computing Machinery.

[79] Saud Wasly and Rodolfo Pellizzoni. Bundled scheduling of parallel real-time tasks. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 130–142, 2019.

[80] M. Xu, L. T. X. Phan, H. Choi, Y. Lin, H. Li, C. Lu, and I. Lee. Holistic resource allocation for multicore real-time systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 345–356, 2019.

[81] 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.

[82] 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.

[83] Bingzhuo Zhong, Abolfazl Lavaei, Hongpeng Cao, Majid Zamani, and Marco Caccamo. Safe-visor architecture for sandboxing (ai-based) unverified controllers in stochastic cyber–physical systems. *Nonlinear Analysis: Hybrid Systems*, 43:101110, 2021.

[84] Bingzhuo Zhong, Siyuan Liu, Marco Caccamo, and Majid Zamani. Secure-by-construction controller synthesis via control barrier functions. *IFAC-PapersOnLine*, 56(2):239–245, 2023.

[85] Bingzhuo Zhong, Siyuan Liu, Marco Caccamo, and Majid Zamani. Towards trustworthy ai: Sandboxing ai-based unverified controllers for safe and secure cyber-physical systems. In *Proceedings of 62nd IEEE Conference on Decision and Control(CDC)*, 2023.

[86] Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. Sandboxing controllers for stochastic cyber-physical systems. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 247–264. Springer, 2019.

[87] Bingzhuo Zhong, Majid Zamani, and Marco Caccamo. Formal synthesis of controllers for uncertain linear systems against omega-regular properties: A set-based approach. *IEEE Transactions on Automatic Control*, 2023.

[88] 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)*, 2023.