

# Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges

Dominik Baumann\*, Fabian Mager\*, Ulf Wetzker, Lothar Thiele *Member, IEEE*, Marco Zimmerling, and Sebastian Trimpe *Member, IEEE*

**Abstract**—Smart manufacturing aims to overcome the limitations of today’s rigid assembly lines by making the material flow and manufacturing process more flexible, versatile, and scalable. The main economic drivers are higher resource and cost efficiency as the manufacturers can more quickly adapt to changing market needs and also increase the lifespan of their production sites. The ability to close feedback loops fast and reliably over long distances among mobile robots, remote sensors, and human operators is a key enabler for smart manufacturing. Thus, this article provides a perspective on control and coordination over wireless networks. Based on an analysis of real-world use cases, we identify the main technical challenges that need to be solved to close the large gap between the current state of the art in industry and the vision of smart manufacturing. We discuss to what extent existing control-over-wireless solutions in the literature address those challenges, including our own approach toward a tight integration of control and wireless communication. In addition to a theoretical analysis of closed-loop stability, practical experiments on a cyber-physical testbed demonstrate that our approach supports relevant smart manufacturing scenarios. The article concludes with a discussion of open challenges and future research directions.

MANUFACTURING and several other industrial sectors are increasingly caught between a rapidly growing demand for individualized, high-quality products, and the constant pressure to maximize profit margins. To successfully handle this balancing act, the manufacturing industry is in the early phases of a revolution: Driven by advances in digitalization and automation, *smart manufacturing* promises more flexible, versatile, and scalable material flows and manufacturing processes through plants that can be reconfigured based on the individual product and overall process requirements [1]. These plants will consist of *physical systems* (e.g., machines, storage systems, supply-chain entities) with sensing, computation,

communication, and actuation capabilities. Reconfigured and automated through edge- or cloud-based services, the physical systems will interact autonomously with one another and human operators. Smart manufacturing will thus give rise to a new breed of complex, large-scale *cyber-physical systems (CPS)* that are safety-critical and rely on networked and distributed control architectures.

Due to the complex interactions among the different entities, wired communication systems, such as fieldbuses and Industrial Ethernet, will reach their limits. Instead, *wireless communication* enables much higher flexibility, allowing for on-demand plant reconfigurability while mitigating cable breaks and faulty connections [2]. Wireless communication is also more robust to certain external influences, including heat, humidity, abrasive substances, and undamped vibrations. Completely untethered, mobile physical systems are possible through the use of batteries, which can be recharged using energy harvesting or wireless power transfer techniques [3]. Furthermore, to reach into tiny spaces and to cover large distances, physical systems built from *embedded hardware* and capable of *multi-hop communication* will be crucial [4], [5]. Taken together, wireless communication offers unprecedented flexibility, cost efficiency, and robustness and is, therefore, a key enabler for smart manufacturing.

Despite many advantages, using wireless instead of wired communication also poses significant challenges, for example, in terms of reliability and security. Moreover, because the manufacturing tasks are defined by algorithms, the software and algorithmic components are closely linked to the physical manufacturing processes. Hence, cyber and physical components are in *feedback* with each other and cannot be designed in isolation. Instead, joint designs are needed to leverage their full potential. In particular, the physical process will dictate timing and safety requirements, which pose challenges especially on the wireless communication and embedded systems, which drive the algorithms (e.g., control, learning, monitoring) that control the overall manufacturing process. The situation is exacerbated when many feedback loops are closed in a flexible and ad-hoc manner across the same network and possibly over large distances, while requiring fast and reliable updates, as is the case in many envisioned smart manufacturing use cases.

**Contributions and road map.** This article provides a perspective on control and coordination over wireless networks as a core technology for future smart manufacturing. Fig. 1 depicts the structure of the article. To ground the subsequent discussion, we start by presenting commonly discussed future use cases

This work was supported in part by the German Research Foundation (DFG) within SPP 1914 (grants ZI 1635/1-1 and TR 1433/1-1) and the Emmy Noether project NextIoT (grant ZI 1635/2-1), the German Federal Ministry of Education and Research (BMBF) project Veritas (grant 01IS18073A), the Cyber Valley Initiative, and the Max Planck Society.

\* Equal contribution.

D. Baumann is with the Intelligent Control Systems Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany; e-mail: dbaumann@tuebingen.mpg.de

F. Mager and M. Zimmerling are with the Networked Embedded Systems Lab, TU Dresden, Dresden, Germany; e-mail: fabian.mager@tu-dresden.de, marco.zimmerling@tu-dresden.de.

U. Wetzker is with the Industrial Wireless Communication group, Fraunhofer Institute for Integrated Circuits Division Engineering of Adaptive Systems, Dresden, Germany; e-mail: ulf.wetzker@eas.iis.fraunhofer.de

L. Thiele is with the Computer Engineering group, ETH Zurich, Zurich, Switzerland; e-mail: thiele@ethz.ch

S. Trimpe is with the Institute for Data Science in Mechanical Engineering, RWTH Aachen University, Aachen, Germany and with the Intelligent Control Systems Group, Max Planck Institute for Intelligent Systems, Stuttgart, Germany; e-mail: trimpe@dsme.rwth-aachen.de

**Accepted final version.** To appear in *Proceedings of the IEEE*.

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

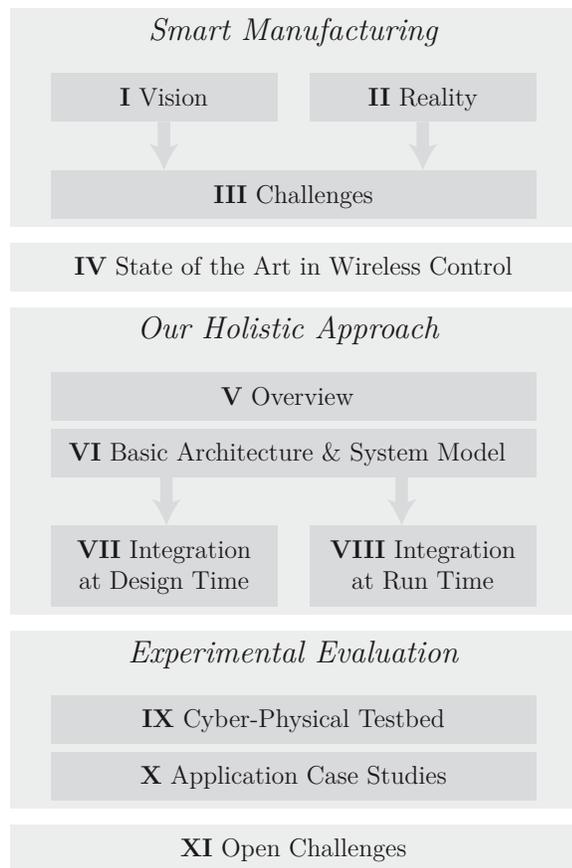


Figure 1. Structure of the article.

of smart manufacturing and derive a unified vision from these (Sec. I). We then contrast this vision with concrete examples of wireless technology in industrial practice today (Sec. II), and analyze the gap between vision and practice. Based on the found reality gap, we derive requirements and challenges of wireless control for smart manufacturing (Sec. III), and review the state of the art (Sec. IV). In particular, we discuss to what extent existing approaches in the literature address core desiderata, such as support for fast feedback, multi-hop communication, stability guarantees, and resource efficiency. We then focus on our own approach from recent work that jointly addresses these aspects through a tight integration of control and wireless communication, both at design time (i.e., prior to operation) [6], [7] and at run time (i.e., during operation) [8]. Sec. V introduces our co-design approach, followed by the presentation of the basic architecture and unified system model (Sec. VI). Through *integration at design time* (Sec. VII), we are able to achieve fast feedback control (on the order of tens of milliseconds) over multi-hop wireless networks with formal guarantees on closed-loop stability. To further increase adaptability and resource efficiency, we deepen the integration toward *run time* (Sec. VIII) by having the control system inform the communication system about its current needs for data transmission, and empowering the communication systems to react accordingly. The developed approaches are illustrated through several experiments on a cyber-physical testbed consisting of ten inverted pendulum systems as fast physical systems and 20 wireless embedded nodes (Sec. IX and Sec. X). The article ends in Sec. XI with a

discussion of open challenges that are not yet addressed by the state of the art in wireless control (including our own work), but are vital for realizing the full vision of smart manufacturing and thus provide ample opportunities for important future research.

**Difference to prior publications.** This article makes several novel contributions compared with our prior work [6], [7], [8], [9] by: (i) articulating a vision of smart manufacturing from the perspective of wireless control and putting this vision in contrast to the current reality in the manufacturing industry; (ii) working out the main challenges for smart manufacturing in general and wireless control in particular to close the gap between vision and reality; (iii) providing a unified, tutorial-style description of how our state-of-the-art approaches, which have appeared in specialized venues and separate communities, are able to address for the first time some of these challenges; (iv) reporting on new experiments on an extended cyber-physical testbed that demonstrate the capabilities of our approach; (v) and highlighting the key research questions yet to be answered.

**Relation to Industry 4.0.** Enabling wireless control constitutes one important building block toward realizing the vision of smart manufacturing. This vision is often described in the context of *Industry 4.0* [10], [11], [12]. Relating the concepts discussed herein with the reference architecture proposed in [10], we aim at enabling wireless control and coordination of entities *within* a smart factory. This can be further extended by also establishing communication between different factories, possibly worldwide, through the *Internet of Things*.

## I. SMART MANUFACTURING: POTENTIAL USE CASES AND VISION

To provide some context and concrete motivation for wireless control in future manufacturing, this section outlines envisioned use cases and the associated benefits (Sec. I-A to Sec. I-D). Abstracting from these use cases, we derive in Sec. I-E the main features and a unifying vision of smart manufacturing.

### A. Reconfigurable Conveyor Belts

In traditional factories, conveyor belts have two ends: one serves as an input, and the other one serves as an output. The machines alongside a conveyor belt are arranged such that the product can be manufactured in a step-by-step fashion. None of the machines in the chain is redundant. The whole setup is carefully designed in advance and follows the same procedure for its entire lifespan to manufacture the ever same product.

By contrast, conveyor belts in smart factories are capable of handling multiple types of products at once [13]. Each product has a unique ID, and the conveyor belt is equipped with multiple redundant machines. The process relies on machines taking information communicated by the product (i.e., the unique ID), extracting which processing steps are necessary, and distributing the workload across the machines such that every product is manufactured as fast as possible without sacrificing on product quality. For this, conveyor belts can be “closed,” e.g., arranged in a circular shape without input and output, enabling various production routes. Coordination is required between product and manufacturing machines as well as among the machines

distributed along the conveyor belt. While wired or mixed wired/wireless communication solutions may be possible in certain scenarios to realize this coordination, the use of wireless networks arguably provides the highest flexibility.

### B. Autonomous Drones

A predominant and widely-cited example of CPS in general is autonomous drones. Due to their agility, flexibility, and capability to operate in three-dimensional space, drones are also of interest for many use cases in future manufacturing [14]. Those use cases include visual inspection and monitoring of factory automation machinery, sensing tasks, transporting objects, or delivering goods. In laboratory contexts and as demonstrations, drones have also been used for cooperative manipulation and transportation [15] and for construction tasks [16].

Taking the example of inspection, drones can act in different ways. A drone can be fully autonomous, regularly monitoring a production plant. Alternatively, a human operator can define waypoints that a drone should follow while the drone transmits image or video data for visual inspection. Next to different levels of autonomy, drones also offer great flexibility as they can switch between different tasks. For instance, a drone monitoring a production plant can interrupt this task momentarily to carry a spare part across the factory hall to a location where it is urgently needed. While these application examples, performed by a single drone, already constitute useful tasks that can improve the efficiency and quality level of a smart factory, it is swarms or fleets of multiple drones that can leverage the full potential of autonomous flying vehicles. Fleets of drones can, for example, carry parts or goods that are too heavy for a single drone, jointly monitor larger plants, or directly participate in the manufacturing process. In all cases, coordination among drones is essential to ensure that they fulfill their tasks and to prevent drones from crashing into each other. Typically, each drone is equipped with an embedded microcontroller for local control tasks (e.g., stabilizing the flight), while distributed control tasks (e.g., coordinating the position of every drone in a fleet) require wireless communication among the drones [17].

### C. Remote Control of Mobile Robots

When multiple mobile robots act in the same workspace, be it in the air or on the ground, planning and coordination among them is often essential. In particular, when planning actions for one robot it can be beneficial to take information from other agents into account for improving safety, minimizing reaction time, and optimizing resource usage. When decision making happens locally on an agent, information from other agents must be wirelessly communicated. Alternatively, in many scenarios it is desirable to partly or entirely outsource the necessary computations to edge devices or cloud services executing inside large data centers [18], [19]. Outsourcing computations can bring many benefits. Edge devices and in particular cloud services offer more compute power, memory, and can exploit additional data sources. This makes the application of more sophisticated control and planning algorithms possible. Further, stored data can be used to improve the algorithms' decisions over time, for example, via machine learning techniques. The edge devices or cloud services then act as an "external brain."

### D. Mining Industry

While not belonging to smart manufacturing in a strict sense, the mining industry shares many design considerations and has an intrinsic need for autonomous and wireless technologies. One reason for this is that mining environments are typically extraordinarily hazardous and dangerous to humans; numerous accidents have been reported over the last decade [20]. A first step toward reducing the risks of human accidents in the mine is the deployment of wireless sensor networks that localize the workers in the mine [20] and monitor the environment [21]. This already comes with severe challenges because large spaces have to be covered by the wireless sensor network. However, it would be even more desirable to employ autonomous agents or to control tasks in the mine remotely. On the one hand, if autonomous or remotely controlled agents could fulfill all tasks in the mine, this would entirely eliminate the risk of human accidents in the mine. On the other hand, it would also enable substantial savings and thus also be economically viable, since pumping fresh air to the workers in the mine is extremely costly. While companies are already moving into this direction [22], the vision of fully autonomous agents working independently in the mine has not been realized to date.

### E. The Vision of Smart Manufacturing

Multiple heterogeneous sensing-computing-actuation units are at the heart of the above-mentioned use cases. As shown in Fig. 2, a unit may involve one main modality (e.g., sensing for a stationary sensor in a mine or computing for an edge device) or all modalities together (e.g., a drone that senses its environment, locally computes a path, and actuates its motors to move along that path). We refer to these sensing-computing-actuation units as *agents*. The agents are generally coupled with the physical world: sensors measure physical quantities and actuators act on their environment. The connection of the two happens through algorithms that execute on embedded computers. Using sensors and actuators, software and processing are thus linked to the physical world, which is the defining characteristic of CPS [23]. An integral aspect of future smart manufacturing systems are *multiple agents* (or multiple CPS) that, each on an individual level, connect the physical world and the cyber space, but are at the same time also interconnected with each other to jointly perform manufacturing tasks. At an abstract level, the overall smart manufacturing system is therefore given by multiple agents that *interconnect* in various ways, execute different types of *algorithms* (control, optimization, learning, planning, etc.), and can thus flexibly and jointly perform diverse tasks that contribute to the overall manufacturing process.

Interconnections occur in various forms and on different hierarchical levels. At the core of many smart manufacturing visions are teams of mobile agents that interconnect with each other. In all the above-mentioned use cases, inter-agent communication is wireless (at least partially) to support the required flexibility, mobility, ease of installation and maintenance, and so on. Mobile agents may additionally connect with stationary agents (e.g., fixed sensors), human operators, and the infrastructure (e.g., edge devices), as illustrated in Fig. 2. Depending on the use case, the multi-agent networks differ in the number of

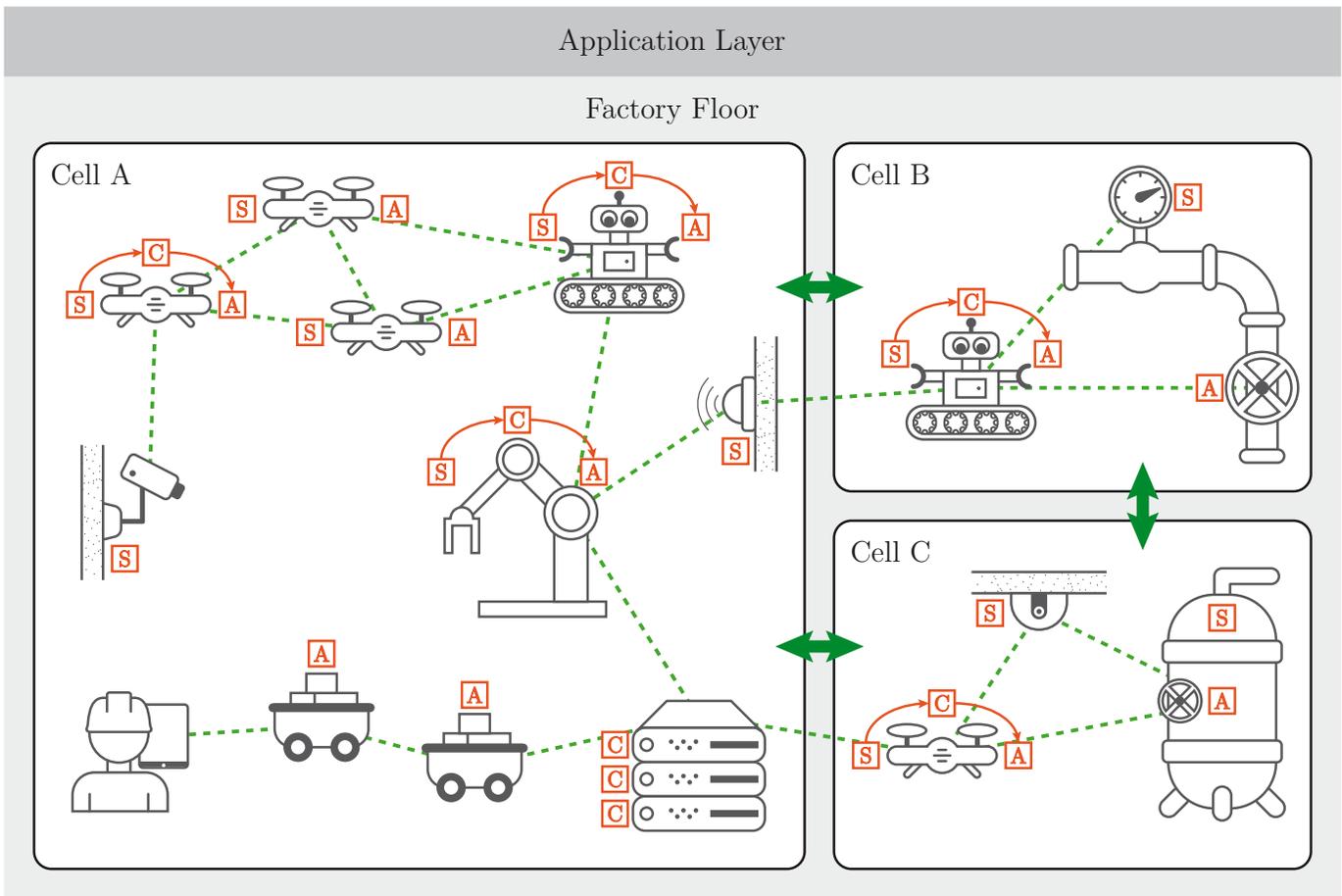


Figure 2. Abstract view of a future smart manufacturing system. Distributed agents are equipped with sensing (S), computing (C), and actuation (A) capabilities, or a subset thereof. Depending on the manufacturing task at hand, which can change dynamically at run time, the agents interconnect wirelessly (green dashed lines depict communication links) to form teams within an automation cell, among cells across the factory hall, and with global wired communication infrastructure to access, for example, cloud services.

agents supported (e.g., two for remote control of a mobile robot, or hundreds in the drone scenario) and the physical distance covered (e.g., direct peer-to-peer links on the conveyor belt, and long-distance multi-hop communication in the mining scenario). Further, one may distinguish different levels of communication (according to [24]), also indicated in Fig. 2: A *local automation cell* comprises multiple interconnected agents, a *factory hall* interconnects several cells, and factories interconnect through the *application layer communication* to global networks, such as the Internet or cloud services. The need for mobility typically decreases with increasing communication level. While higher-level communication between cells or with the cloud can often resort to cabled routing-based networking solutions, wireless communication is critical on the cell and factory hall levels. Depending on the envisioned scenarios, wireless communication may extend across several cells or the whole factory hall (e.g., coordinated inspection with drones).

A key prerequisite for the envisioned flexibility in smart manufacturing is the ability to perform various types of tasks that are enabled and defined through algorithms and software. For instance, the same drone may switch from an inspection task to bringing an urgently needed tool to a worker from one moment to the next, just by downloading a new algorithm from the cloud in a split second. Algorithms in smart manufacturing

can either use only local information and resources available to a single agent (localized) or use information and resources from multiple agents (distributed), which is supported through interconnections and the communication system.

Many different types of algorithms are relevant and jointly define the smart manufacturing process. Nevertheless, ultimately, all algorithms operate on data from physical sensors or received over the network, and decisions lead to actions in the physical world (the manufacturing). Hence, *control* algorithms are an essential component of any manufacturing system. Controllers compute actuator commands based on sensor information to achieve a desired control objective, including stability, precise positioning, or robust performance in the face of uncertainty.

For the design of a control algorithm, the question of what information is available as input is of key importance and leads to different control architectures. In a *centralized* architecture, the controller has global information; that is, it has access to all relevant sensor readings based on which it computes all actuator commands. In a *decentralized* architecture, the control computation is distributed among multiple controllers, each of which has access only to local sensors and computes commands only for local actuators; in a purely decentralized architecture, no information is shared between the controllers. Finally, *distributed* control resides in between the two extremes:

control computations are distributed, but some information is shared between the controllers. Typically, centralized control yields the best performance as compared with decentralized control [25] because it has maximal information available.

In smart manufacturing, the control architecture is not given but can be influenced and adapted, in particular through the communication system. For example, wireless links can be established between otherwise decoupled controllers, at the cost of using extra resources such as communication bandwidth and energy. Thus, depending on the communication support, control may be easier or harder, and the co-design of control and communication in smart manufacturing gives significant additional design freedom over traditional fixed, cable-based architectures [26]. In general, the smart manufacturing system comprises a multitude and various types of feedback loops and control algorithms across different hierarchies. Besides control, typical algorithmic components are *optimization*, *planning*, *data fusion*, and *machine learning*. Here, architectural aspects play a similar role as discussed for control [27], [28], [29].

A key characteristic of future smart manufacturing systems is the demand for unprecedented *flexibility* and *adaptability* of the overall system. This concerns both the interconnections and the algorithms. Agents need to flexibly interconnect to form heterogeneous teams that constitute an automation cell or may even span across multiple cells. Similarly, algorithms have to adapt to the task at hand. Adaptation may happen locally on a computing unit, through the interaction of many agents, or by downloading a new algorithm from the cloud, for example.

Fig. 2 abstracts envisioned smart manufacturing use cases and illustrates key system characteristics. Before developing requirements and challenges for realizing this vision, we discuss in the next section how this vision relates to the current reality in the manufacturing industry.

## II. SMART MANUFACTURING: INDUSTRY EXAMPLES

We now present three example applications, where wireless technology, as a key enabler for smart production, is actually adopted by the manufacturing industry today. The discussion is based on our experience at Fraunhofer IIS in troubleshooting real-world industrial communication systems. Afterward, we examine the gap between envisioned and existing use cases.

### A. Automated Guided Vehicles

The automotive industry progressively uses automated guided vehicles (AGVs) to drive car bodies to the respective assembly station. In the considered example, 50 AGVs are commissioned on the first factory floor and arrive at the assembly line on the second floor by use of an elevator. All AGVs are wirelessly connected to a central control system, which plans the routes for each of them according to the car model, the model's individual configuration, and the utilization of the different assembly stations. This way, the factory floor is used more effectively without being tied to a specific vehicle model.

Communication between the control system and the AGVs is based on IEEE 802.11 (i.e., Wi-Fi) in the unlicensed 2.4 GHz frequency band. After more than a year of productive operation, unpredictable communication faults occurred more and more

frequently, resulting in unplanned stops of the AGVs. This was caused by an ever-increasing use of the 2.4 GHz frequency band by additional Wi-Fi and Bluetooth-based applications, which led to temporary coexistence problems. Such coexistence problems are among the hardest to troubleshoot in real-world wireless networks [30]. Unfortunately, moving to a licensed frequency band (e.g., a 4G/LTE or 5G cellular network) is not a viable option for many applications as running costs are high; companies are unable to operate the network themselves, leading to long service deployment times, demanding reliability, latency, and real-time requirements cannot be met; and existing security mechanisms do not offer sufficient protection for sensitive data [31], [32]. So far, technological advances have only been able to contain the coexistence problem, but not to solve it. A co-design of control and communication is the most promising approach to ensure the availability of the system.

### B. Cordless Power Tools

In machine construction, cordless power tools are frequently used to give workers maximum freedom of movement and to access hard-to-reach places. To further improve the working conditions and facilitate continuous quality assessment, these tools can be connected to a central control system, which sets the torque for each screw individually and logs the measured values in a database. In the concrete use case, situated in a motor assembly line at Mercedes-Benz Ludwigsfelde GmbH in Germany (see Fig. 3), 18 cordless power tools are connected to a central programmable logic controller (PLC) via Bluetooth across a 100 m by 500 m factory floor. While this wireless solution provided the desired freedom of movement, the reliability was not satisfactory. Especially in hard-to-reach places, the shadowing effect of both the workers and the cordless power tools resulted in severe message loss. Such message loss may be prevented by introducing redundancy in the network, for example, by installing more nodes that form a mesh topology and forward messages on behalf of other nodes (i.e., multi-hop communication). The star topology of Bluetooth, however, proved non-ideal for dealing with shadowing issues.

### C. Parametrization of Automotive Electronics

The various electronic control units (ECUs) installed in a car during production are initially not customized for the specific vehicle model and equipment. A parametrization of the generic software components running on the ECUs is carried out while going through multiple test stations (e.g., chassis dynamometer, advanced driver-assistance systems, sensor calibration, and the like). During this calibration procedure, the worker needs to set up the control system by connecting the test station to the ECU through vehicle communication interfaces (VCIs). In the considered use case, the VCIs are based on Wi-Fi operating in the unlicensed 5 GHz frequency band. Up to 30 of these free-moving wireless VCIs are used on a multi-level all-metal factory floor of 100 m by 150 m. The coverage of this area is provided by 15 Wi-Fi access points. Since the 5 GHz frequency band is shared with weather radar stations, the access points need to perform "in service monitoring" for the presence of radar pulses. If radar pulses are detected, an access point must



(a) Factory floor.



(b) Cordless power tool.

Figure 3. Motor assembly line at Mercedes-Benz Ludwigsfelde GmbH in Germany.

move to another channel within ten seconds of “channel move time.” In the considered use case, unfounded detections of radar pulses by the access points occurred. This led to sudden channel changes and a reconnection of the VCIs, which caused delays or even interruptions of the calibration procedure.

#### D. The Reality of Smart Manufacturing

These real-world application examples are in stark contrast to the envisioned use cases of smart manufacturing presented in Sec. I. Especially the use of wireless communication today is restricted to applications that are neither time- nor safety-critical. Closing sensing-computing-actuation loops continuously over wireless communication networks, as put forward by the smart manufacturing vision, is extremely far from what industry is currently willing to take up, although the wish to wirelessly interconnect robots, rotating parts, and AGVs in automated high-rise warehouses is commonly expressed by many of our industry partners. However, already in the comparably simple real-world use cases discussed above, wireless communication causes several issues. These issues are mainly related to the reliability of wireless communication because, for example, shadowing effects and coexistence problems can cause severe, unpredictable message loss or complete disconnections. Thus, the main reason for the reluctance of industry to adopt wireless technology for more challenging and critical tasks appears to be a lack of trust in the reliability of wireless solutions.

### III. CHALLENGES FOR WIRELESS CONTROL IN SMART MANUFACTURING

The comparison between vision and reality of smart manufacturing revealed that there clearly exists a large gap. Our analysis also identified three main challenges that need to be solved to close this gap. We detail those challenges next.

#### A. Dependability

The real-world examples and envisioned use cases clearly demonstrate that smart manufacturing systems are large-scale, complex, and *safety-critical CPS*. For this reason, *dependability*

is of utmost importance, which entails that the systems must be provably secure and function as intended under realistic attacker and fault models. In addition to broken sensors, failing devices, software bugs, side-channel attacks, and so on, these models must account for message loss and disconnections due to the notorious unreliability of wireless communication. Based on these attacker and fault models, rigorous proofs of dependability properties along with an end-to-end validation of the systems on real-world smart manufacturing demonstrators are needed to promote industrial *and* societal acceptance of wireless technology in safety-critical applications—*there seems no alternative path to realizing the smart manufacturing vision*.

Concerning wireless control, the *stability* of the feedback loops in a smart manufacturing system is arguably the most basic dependability property that has to be provably satisfied. To guarantee closed-loop stability and also achieve the application-specific control performance, the ability to provide *fast* feedback over a wireless communication network is essential to keep up with the dynamics of the physical (often mechanical) systems in smart manufacturing. For example, controlling the motion of a remote robot and coordinating a fleet of drones, as mentioned in the envisioned use cases from Sec. I, require update intervals of tens of milliseconds [19], [33]. Such real-time requirements must also be met when wireless communication occurs over *multiple hops*, which is key, for example, when large distances in mining or across a whole factory floor are to be covered.

#### B. Adaptability

Smart manufacturing systems also have to be *highly adaptive* to support on-demand reconfigurability of conveyor belts, dynamic assignment of new tasks to individual drones, or the reformation of entire fleets of AGVs. This means in particular that the systems must adapt at run time to changes in application requirements, hardware and software components, and dynamics of the environment in which they operate. As a result of such adaptability, the number, types, and capabilities of agents, the algorithms they execute, and how they are interconnected can change in unforeseen and complex ways throughout the lifespan of a smart manufacturing system.

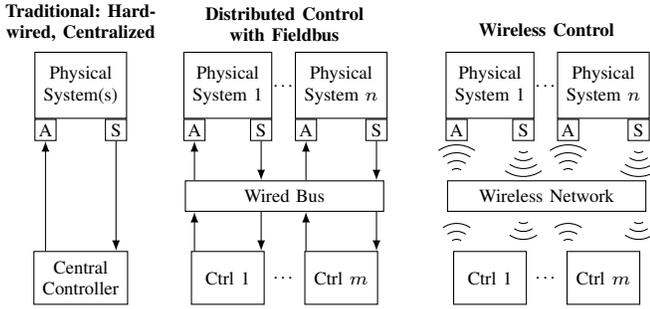


Figure 4. Evolution of control architectures. *Physical systems with sensors (S) and actuators (A) are connected to controllers (Ctrl) over point-to-point wires (left), a wired bus (center), or a wireless network (right).*

With regard to wireless control, a first step toward adaptability is to enable the system to switch at run time between a fixed set of *operating modes*, while ensuring closed-loop stability and control performance. A mode may correspond to a distinct manufacturing task or conveyor belt configuration, and *mode changes* may be triggered by the cloud-based automation logic or the arrival of a product at a conveyor belt. For an effective collaboration of multiple agents, a *distributed implementation*, where every agent locally uses information received over the network to solve a common control task, is highly beneficial. *Multi-hop communication* supports unforeseen interconnections; for example, no additional access points must be installed if suddenly the need for a larger wireless coverage arises.

### C. Efficiency

The algorithms (control, optimization, learning, etc.), communication protocols, and other software components need to operate *efficiently* because of limited compute power, memory capacity, and wireless bandwidth. This is particularly relevant on the agent level, where embedded hardware such as low-cost microcontrollers and wireless radios are desirable for economic, size, or weight considerations. Energy efficiency is not only a concern for battery-powered drones, remote energy-harvesting sensors, and the like, but also for cloud-based services [34].

With regard to wireless control, models and methods are needed that increase control performance (which may translate into shorter manufacturing cycles and higher product quality) with the least resources. To achieve this goal, controllers should, for example, *save resources* whenever possible and *reallocate* otherwise unused resources to avoid wasting them.

## IV. STATE OF THE ART IN CONTROL OVER WIRELESS

This section reviews efforts reported in the literature to solve the wireless control challenges outlined in the previous section.

Such efforts are required as closing feedback loops over a wireless network can make the control design significantly more difficult compared to traditional control architectures [48], [49], [50], [51]. Traditionally, sensors and actuators are connected to a centralized controller through *point-to-point wires* (Fig. 4, left) [51]. Although centralized control is beneficial because the controller has global information, it is often impractical for large-scale systems. An alternative is decentralized control,

where the system is split into a number of subsystems, each connected to a local controller, but without signal transfer between them [48], [49]. However, decentralized control can exhibit poor performance, and it may not even be possible to achieve closed-loop stability [48]. Hence, communication networks in the form of *wired busses* were introduced [51] (Fig. 4, center), which are still widely used in automation and control today [52], [53]. Replacing a wired bus with a *wireless network* (Fig. 4, right), as required for smart manufacturing to close the sensing-computing-actuation loops shown in Fig. 2, leads to communication imperfections including longer transmission delays, larger jitter, as well as higher and correlated message loss (see Sec. V for a detailed discussion).

Over the past two decades, a lot of research in the control community has considered these communication imperfections in the form of *theoretical* stability analyses for different control architectures, transmission delay distributions, message loss models, etc. Numerous *simulation* case studies have also been carried out, for instance, based on the WirelessHART industrial standard [54], [55]. This line of research has led to a principled understanding of the wireless control problem, and excellent surveys [56], [57] review the developed approaches and results.

However, as discussed in Sec. II, one of the primary concerns in industry today is a lack of trust in the reliability of wireless solutions. This trust can only be established by complementing rigorous theoretical analyses with real experiments on realistic cyber-physical testbeds [58]. Therefore, we focus our discussion in the following on control-over-wireless approaches that have been validated on physical platforms and real wireless networks.

Table I qualitatively compares such approaches currently found in the literature by assessing whether the wireless control challenges outlined in Sec. III have been addressed. Although these constitute basic challenges toward realizing the smart manufacturing vision and many challenges are yet to be solved, as discussed in Sec. XI, we can see that none of the existing approaches is capable of addressing all basic challenges.

Having a closer look at Table I, we observe that only [7], [8] support a *distributed implementation*<sup>1</sup>. Distributed implementation is understood in the sense that agents are equipped with local computing units and use their own sensor readings *and* information received over the communication network to locally compute new actuation commands. All other approaches support only a central computing unit that computes the new actuation commands for all agents. However, support for both architectures is needed to enable the use cases from Sec. I.

Further, motion control for mobile robots, for example, needs to happen at *fast update intervals* (i.e., tens of milliseconds) to keep up with the physical dynamics of the agents. At the same time, in large factories or for remote control in mines, large distances are to be covered, requiring *multi-hop communication*. We see that apart from [7], [8] no solution can cater for this crucial requirement. References [35], [36] control a double-tank system, where update intervals of around 1 s are sufficient, over a single-hop network. Several works consider different variants of inverted pendulum systems [37], [38], [39], [43], [44] or

<sup>1</sup>Another work that demonstrated distributed control over wireless is [6], a prior conference version of [7]. In general, [6] provides the same properties as [7], the main difference being that [6] does not support mode changes.

Table I  
 DESIGN SPACE OF WIRELESS CPS THAT HAVE BEEN EVALUATED ON PHYSICAL PLATFORMS AND REAL WIRELESS NETWORKS.

Work	Dependability			Adaptability		Efficiency	
	Stability analysis	Fast update intervals	Multi-hop	Mode changes	Distributed implementation	Reallocation	Resource savings
[35], [36]	✓	✗	✗	✗	✗	✓	✓
[37]	✓	✓	✗	✗	✗	✗	✗
[38]	✓	✓	✗	✗	✗	✗	✗
[39]	✓	✓	✗	✗	✗	✗	✗
[40]	✓	✓	✗	✗	✗	✗	✓
[41], [42]	✓	✓	✗	✗	✗	✗	✗
[43]	✗	✓	✗	✗	✗	✗	✗
[44]	✗	✓	✗	✗	✗	✗	✗
[45]	✗	✗	✓	✗	✗	✗	✗
[46]	✗	✗	✓	✗	✗	✓	✓
[47]	✗	✗	✓	✗	✗	✗	✓
[7]	✓	✓	✓	✓	✓	✗	✗
[8]	✗	✓	✓	✗	✓	✓	✓

mobile robots [40], [41], [42], demanding update intervals of at most tens of milliseconds. These approaches are all restricted to single-hop networks. Wireless control over multi-hop networks is demonstrated in [45], where the lightning in an operational road tunnel is controlled, in [46] which considers power capping management in a data center, and in [47] where Matlab simulations of physical plants are controlled. The update intervals in those setups are on the order of several seconds. While the majority of the works that consider control over single-hop networks provide some kind of stability analysis, only [6], [7] can do this for multi-hop networks. However, since a practical demonstration can never cover all potential situations that may be encountered during a real execution in a smart factory, a theoretical analysis is absolutely indispensable. Naturally, this becomes more challenging under multi-hop communication. Also, the demand for (basic) adaptation at run time through *mode changes* (e.g., changes between a fixed set of application tasks) are exclusively demonstrated in [7].

Another key characteristic of smart factories is that many agents need to transmit information over the wireless channel, such as their current positions, latest sensor values, or actuation commands. Moreover, at least some of the agents and infrastructure sensors attached to walls or moving equipment in a smart factory may be powered by batteries. As the bandwidth of a wireless channel is limited, and wireless radios draw considerable power, messages should only be sent if necessary.

Motivated by this, the control community has developed a variety of event-triggered and self-triggered methods [59], [60]. Both methods send information only upon the occurrence of certain events, such as an error growing too large. Event-triggered control algorithms take this decision instantaneously. Thus, the communication system cannot reallocate resources that are not needed by the control system. To enable this, the control system needs to inform the communication system in advance about its communication demands. This can be realized through self-triggered control, where at each communication instant the control system already decides when it needs to transmit information the next time. Table I therefore also lists integrations of self-triggered control with communication systems to enable *resource savings* and *reallocation*. Next to self-

triggered approaches, [46] considers a network manager which checks the communication demands of all agents and assigns slots accordingly. However, as the update intervals considered in [46] are 20s or longer, this solution is not applicable to the fast feedback loops in future smart manufacturing systems. In summary, [8] is the only work that can combine resource reallocation and resource savings with fast update intervals, multi-hop communication, and a distributed implementation.

Our analysis of the state of the art shows that [7], [8] are superior to other prior work. Thus, in the following sections, we illustrate the approach presented in [7], [8] in more detail.

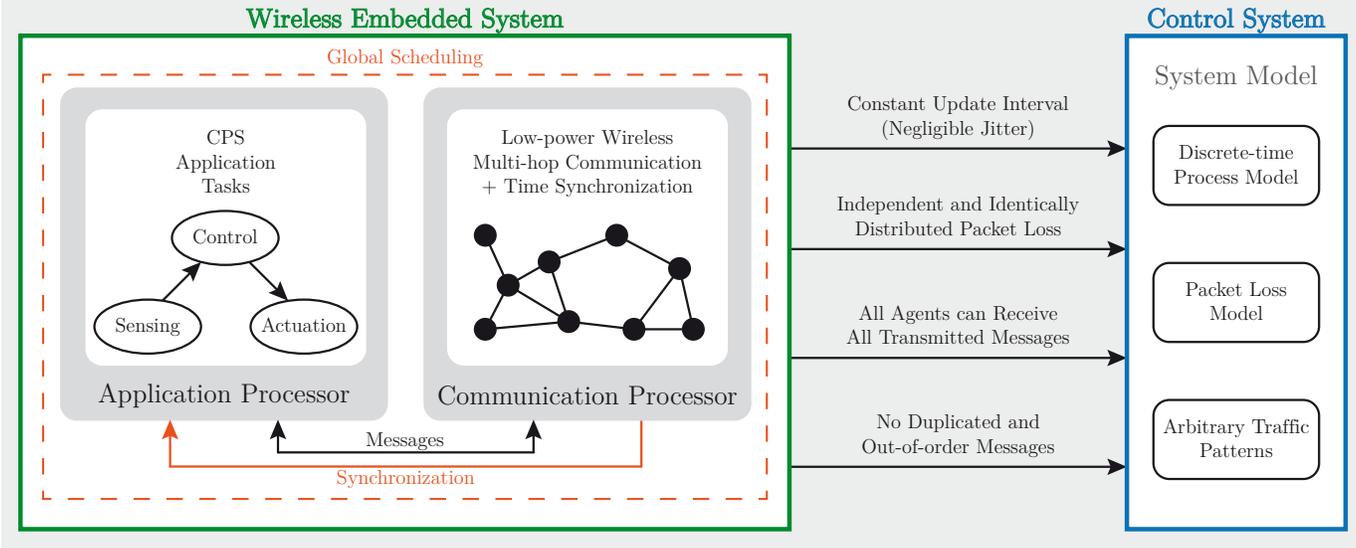
## V. HOLISTIC APPROACH: OVERVIEW

In Sec. III, we identified three main challenges that need to be addressed to close the gap between vision and reality of smart manufacturing, while in Sec. IV we discussed to which extent those challenges are addressed by the current state of the art in research. From this discussion, the approaches proposed in recent works [7], [8] resulted to be the only approaches that successfully address these challenges. Those works propose a *holistic approach* that tightly integrates the control and wireless communication systems with all their hardware and software components, both at design and at run time.

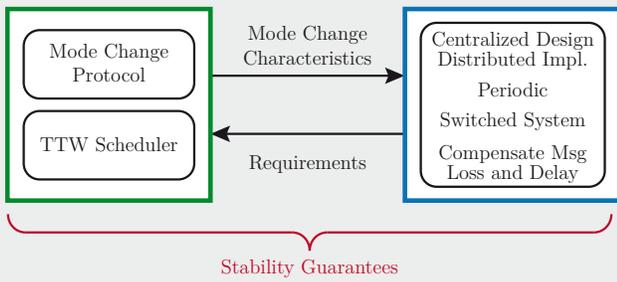
The reasons why prior solutions fail to address the above-mentioned challenges lie in the imperfections of the wireless communication medium as compared to cable-based solutions:

- C1 Limited throughput.** Depending on the wireless technology used, the number of messages carrying sensor readings and control signals that can be exchanged among distributed agents per unit of time can be significantly lower compared with a wired communication system. Also, the throughput in a multi-hop network is generally lower than in a single-hop network where all agents can directly communicate with one another. Fundamental trade-offs between achievable throughput, communication range, and energy cost require non-trivial system design decisions to be able to meet the dependability and efficiency requirements.
- C2 Unpredictable delays.** Communication delays hinder the system-wide coordination of distributed entities, can impair control performance, and can even make it impossible

Basic System Architecture (Section VI)



Integration at Design Time (Section VII)



Integration at Run Time (Section VIII)

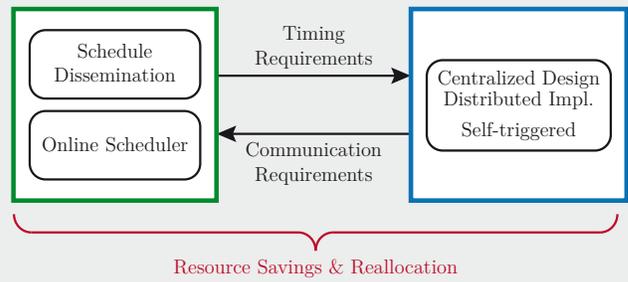


Figure 5. Overview of system architecture with co-design and integration approaches.

to stabilize a feedback system, especially when the delays vary unpredictably fashion (i.e., distribution of the delays is unknown). Such unpredictable, time-varying delays are the norm rather than an exception in wireless networks, caused by, for example, retransmissions of lost messages, varying dwell times in message queues, and dynamic changes in the routing paths or communication schedules.

- C3** *Constrained communication patterns.* Although wireless communication is broadcast in nature, allowing all devices in communication range of a sender to receive a message if they have their radio turned on, the need for reliable and efficient communication has led to many protocols that support only specific communication patterns. For instance, in WirelessHART, all messages need to pass through the central gateway, which makes an effective coordination of all agents in a cell highly inefficient. Moreover, constrained communication patterns make it more challenging to provide certain dependability properties: Well-known fault-tolerance mechanisms [61] rely on the ability to share information among a set of agents (e.g., three agents, where one acts as the primary controller and the other two act as back-up controllers that seamlessly take over

when the primary controller fails) and closed-loop control without visibility into the global system state can lead to poor performance and render stability infeasible [62].

- C4** *Correlated message loss.* It has been shown that communication in both wireless [63] and wired [64] networks is often subject to correlated message loss, that is, multiple consecutive messages can be lost in a “burst” and the maximum length of these bursts is extremely difficult to predict. As correlated message loss interrupts a feedback loop in an unpredictable way, it can be very hard if not intractable to perform a valid closed-loop stability analysis.

As indicated in Fig. 5, our holistic approach starts by taming those imperfections to the extent possible on the wireless embedded system side. The resulting key characteristics are then embedded into a common system model that also captures the dynamics of the physical systems to be controlled. Such a unified underlying model allows for the co-design of the control and communication systems based on relevant mutual properties. For instance, the control system needs to countervail inherent imperfections of wireless communication, such as message loss and delays, while the communication system needs to account for the traffic demands of the control system.

Since all of this can be done prior to execution, we call this *integration at design time*. Nevertheless, at run time, the control and communication systems must adapt their functioning in a timely and safe manner to changes in the mutual properties to keep satisfying the requirements. Such changes can be due to varying application tasks, hardware or software updates, or environment dynamics, including benign and malicious ones. For example, often interference-free operation of a wireless CPS cannot be guaranteed. To keep the smart manufacturing system in a safe state, the control system needs to adapt and put more focus on robustness to reduce communication demands, while possibly sacrificing control performance. In this way, dependability can be ensured without sacrificing adaptability and efficiency, yet it requires a tight integration of control and communication both at design time and *at run time*.

Over the past three years, we have made significant progress toward such a holistic approach. Fig. 5 illustrates the overall architecture of the wireless CPS along with our co-design and run time integration approaches. The basic system architecture, described in Sec. VI, embodies the fundamental idea underlying our approach: Address challenges **C1–C4** to the extent possible in the design of the wireless embedded system such that the system model can account for them in an accurate and tractable way. The integration of the control and wireless embedded systems at design time and run time can then be achieved using straightforward methods, as detailed in Secs. VII and VIII.

As a result, we were able to demonstrate for the first time reliable and efficient control and coordination of multiple physical systems over low-power wireless multi-hop networks at update intervals of 20-50 ms [6]. Specifically, through integration at design time, we obtained formal closed-loop stability guarantees even while the wireless CPS adapts at run time to varying application needs and network dynamics [7]. Through integration at run time, we have shown significant bandwidth and energy savings as well as the dynamic reallocation of otherwise unused resources [8]. With this, our holistic approach demonstrated the capability to meet certain dependability, adaptability, and efficiency requirements of future smart manufacturing that were impossible before.

**Road map.** The following three sections describe our holistic approach in more detail. Afterward, Sec. IX describes a full-fledged cyber-physical testbed we designed to systematically evaluate our approach on physical platforms and real wireless networks. Sec. X presents experimental results that illustrate the capabilities of our approach in a range of scenarios that are representative of future smart manufacturing systems. While we believe that our work represents an important step toward realizing the smart manufacturing vision, several important challenges still remain to be solved. Sec. XI discusses some of these challenges and opportunities for future research. Essential parts of our work are available as open source (see Table II).

## VI. BASIC ARCHITECTURE AND SYSTEM MODEL

The foundation of our approach toward building a dependable, adaptive, and efficient wireless control system is a wireless embedded system architecture that facilitates the co-design of communication and control at design time and at run time. To realize this co-design, we proceeded in two steps:

Table II  
AVAILABILITY OF THE DIFFERENT COMPONENTS OF THE PRESENTED APPROACH AND CYBER-PHYSICAL TESTBED

Component	Available
Dual processor platform	<a href="https://gitlab.ethz.ch/tec/public/dpp/-/wikis/home">https://gitlab.ethz.ch/tec/public/dpp/-/wikis/home</a>
Inverted pendulum systems	Quanser systems (proprietary): <a href="https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/">https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/</a> , Hardware details for the self-built pendulum systems and the interface between pendulums and the DPP are available on request.
Communication code	The communication code is based on Glossy, see <a href="https://github.com/ETHZ-TEC/LWB">https://github.com/ETHZ-TEC/LWB</a> .
TTW scheduling framework	<a href="https://github.com/romain-jacob/TTW-Artifacts">https://github.com/romain-jacob/TTW-Artifacts</a>
Control code	The details of the control algorithms can be found in [6], [7], [8]. Their implementation is highly specific to our hardware and physical systems and available on request.

- 1) mitigate imperfections (i.e., delay, jitter, packet loss) on the wireless embedded system side to the extent possible;
- 2) transfer the resulting properties of the wireless system into a system model that is used for the control design.

To this end, we designed a wireless embedded system consisting of a hardware platform, a wireless communication protocol, and a global scheduling policy at its heart. We provide a brief overview of these components in the following; more detailed descriptions can be found in [6], [7].

### A. Wireless Embedded System

**Hardware platform.** One of the key building blocks of our wireless embedded system is the hardware platform. It combines all necessary components while providing the mechanisms needed to achieve the goals of dependability, efficiency as well as predictability (see challenge **C2**). From a hardware perspective, we argue for low-power and low-cost commodity hardware as it can be deployed numerous and almost everywhere due to its small form factor. As discussed in Sec. I-E, we find multiple interconnected sensing-computing-actuation units in typical CPS applications. Thus, we have two kinds of tasks: the application task, which consists of sensing, computing, and/or actuation, and the communication task. Both tasks are essentially independent of each other, except that the application generates the data to be communicated, and may execute concurrently with vastly different workloads.

We thus leverage a custom-built heterogeneous dual processor platform (DPP). This DPP features a communication processor (CP) and a more powerful application processor (AP). Because application and communication tasks execute on different processors, we avoid any resource interference between them, which could otherwise lead to unpredictable execution delays. Furthermore, the data exchange between AP and CP is handled by Bolt [65], a processor interconnect with formally verified worst-case execution times, a critical property to enable real-time operation and to mitigate challenge **C2**.

**Wireless communication protocol.** In general, communication must be reliable enough to guarantee a safe execution of CPS,

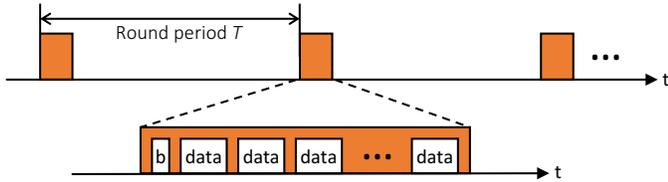


Figure 6. Time-triggered operation of the low-power wireless protocol. Communication occurs in rounds that are scheduled with a given round period  $T$ . In the time interval between rounds, there is no communication between nodes, i.e., communication sleeps for energy efficiency. Every beacon ( $b$ ) and data slot corresponds to an one-to-all Glossy flood [69].

as captured by challenges **C2** (unpredictable delays) and **C4** (correlated message loss). The available bandwidth should be sufficiently large in order to keep up with the dynamics of the physical processes to be controlled (see challenge **C1**). Those requirements also need to be met when large distances are covered, that is, when messages must travel multiple hops. Finally, the protocol needs to ensure consistent functioning also when agents are moving in the network, under changes in the environment, or under dynamically changing communication patterns (see challenge **C3**). In such cases, we are facing a dynamic network where the quality of communication links continuously changes, and links appear and disappear.

The research community and industry have developed numerous low-power wireless communication protocols over the past few years. Comparing them is non-trivial as they were evaluated in different environments, on different hardware platforms, and using different experimental use cases. Efforts have recently been made to benchmark some of these protocols [66]. In particular, the results of the annual EWSN Dependability Competition [67] clearly show that, under various considered benchmark scenarios, stateless flooding-based solutions have achieved the best performance in terms of latency, reliability, and energy efficiency, outperforming protocols based on stateful routing [68]. All top-three solutions in the competitions have in common that they are based on a variant of Glossy [69].

Glossy is a multi-hop communication primitive based on synchronous transmissions [70]. It inherently supports dynamic networks as it operates topology-agnostic, i.e., independent of the underlying network state. In a Glossy flood, one message is disseminated from one node to all other nodes in a (multi-hop) network with a reliability that has been found to be above 99.9% on various real testbeds with more than 200 nodes [71]. The time required for this is close to the theoretical minimum for half-duplex radios. In the rare event of message loss, it has been shown that these losses can be safely approximated by an i.i.d. Bernoulli process [72], [73], which is an essential property for the control design. Glossy can also time-synchronize all nodes in the network with an accuracy of a few microseconds.

Our wireless protocol is also based on Glossy. As illustrated in Fig. 6, nodes communicate in periodic rounds with period  $T$  and sleep in between to save energy. This is possible because of Glossy's time synchronization, which is used to schedule the start of the communication rounds at every node. Each round consists of a series of *slots*. In each slot, data are sent by a single

node using a Glossy flood. The *beacon slot* is always initiated by a dedicated node, called *host*, serving as synchronization point for all other nodes. All other communication slots after the beacon are *data slots*, which contain the actual application messages, one slot for each message. As a result, one round realizes the data exchange required for one control cycle within a few tens of milliseconds, which makes our protocol applicable for many control applications and especially for fast feedback control. Moreover, since every message can be received by all nodes, any communication pattern is inherently supported (see challenge **C2**). This significantly simplifies the control design as it enables individual nodes to work based on global knowledge. Although this kind of broadcast-only communication may seem wasteful, many solutions based on synchronous transmissions that have been developed over the past few years demonstrate that this approach can outperform traditional routing-based approaches by a significant margin in terms of efficiency and reliability across a wide range of real-world scenarios [70].

**Global scheduling.** In order to provide end-to-end guarantees for the overall CPS, a global scheduling of the different tasks executing on the two processors of the DPP platform and a time-predictable wireless data exchange is needed to cope with delays and jitter (see challenge **C2**). The time needed from sensing the state of the system until applying the corresponding control command is the end-to-end delay  $T_D$ . Intervals between consecutive sensing and actuation tasks define the update interval  $T_U$ . An appropriate task scheduling achieves two goals:

- 1) the most recent data is available for communication;
- 2) variations of  $T_U$  and  $T_D$  are minimized to the extent that both can be bounded by a small jitter, which considerably simplifies the control design and the further analysis of the overall CPS (e.g., in terms of closed-loop stability).

The basis of our scheduling is an accurate global time reference provided by the communication protocol, at every node. An exemplary schedule and further details are explained in [6].

**Essential properties.** With the presented wireless embedded system, the control design can build on the following properties:

- P1** The theoretical worst-case jitter on  $T_U$  and  $T_D$  is bounded by  $\pm 50 \mu\text{s}$  for update intervals of up to 100 ms [6].
- P2** Message losses across the wireless network can be safely approximated by an i.i.d. Bernoulli process [72], [73].
- P3** The application or control logic can make use of arbitrary communication patterns over a multi-hop network.
- P4** Message duplicates and out-of-order message deliveries are impossible by design.

## B. Control System

In this section, we present a mathematical model of the overall CPS capturing the key properties that result from the wireless embedded system design. We consider two settings, depicted in Fig. 7: remote control and distributed control. We first discuss the common underlying system model and then comment on differences between both settings.

The mathematical model shall describe the physical system and the key properties of the wireless embedded system. As stated in property **P1**, the jitter is orders of magnitude lower

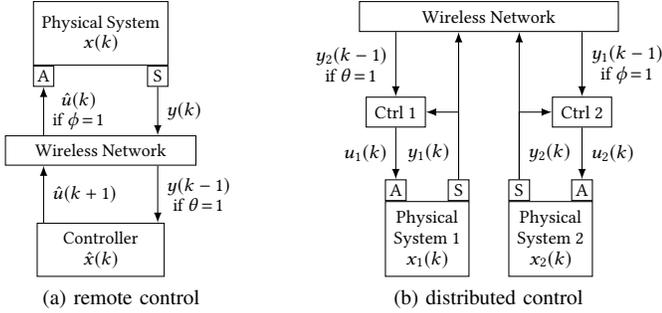


Figure 7. Block diagram of the overall control system. We consider two scenarios, namely remote control (Fig. 7a) and distributed control (Fig. 7b).

than the update intervals of tens of milliseconds we target with mechanical systems. Thus, the jitter can be safely neglected for control design [74, p. 48]. This allows us to model the system in discrete time, where one discrete time step corresponds to one communication round. For each controlled physical system  $i$ , we assume linear and time-invariant (LTI) dynamics,

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + v_i(k), \quad (1a)$$

with  $x_i(k) \in \mathbb{R}^n$  the state of the system,  $u_i(k) \in \mathbb{R}^m$  the control input,  $v_i(k) \in \mathbb{R}^n$  a normally distributed random variable (capturing, for instance, model uncertainty),  $k \in \mathbb{N}$  the discrete time index, and  $A_i$  and  $B_i$  matrices of appropriate dimensions. Further, we assume that we can measure the full state, but measurements are corrupted by noise,

$$y_i(k) = x_i(k) + w_i(k), \quad (1b)$$

where  $w_i(k)$  is again a normally distributed random variable.

Message loss is modeled through variables  $\theta(k)$  and  $\phi(k)$  (cf. Fig. 7), which are two independent Bernoulli random variables indicating successfully received ( $\theta(k) = 1$ ,  $\phi(k) = 1$ ) respectively lost ( $\theta(k) = 0$ ,  $\phi(k) = 0$ ) messages. While the Bernoulli assumption is oftentimes not satisfied for traditional wireless systems based on routing [72], it is indeed valid for our wireless embedded system design as per property **P2**. To ease the presentation and since both  $\theta$  and  $\phi$  are i.i.d., we omit the time dependence of the variables in the following.

For remote control (see Fig. 7a), we consider a remote computing unit, the controller, connected to the physical system over a wireless network. In this case, sensor measurements and control commands need to be sent over the wireless network. In this scenario, all computations are executed at a remote location, and no computational power is required directly at the physical system. Thus, in case a data packet containing the next control input is lost, we perform a zero-order hold (ZOH), i.e., we keep applying the previous control input:

$$u(k) = \phi \hat{u}(k) + (1 - \phi)u(k-1). \quad (2)$$

Here,  $\hat{u}(k)$  denotes the control input computed by the controller, to be made precise in the following section. We present the control strategy and analysis for remote control for the single-loop case, i.e., with only one system. Therefore, we drop the index  $i$  whenever discussing the remote control scenario.

In the distributed control scenario (see Fig. 7b), we assume systems to be collocated with a local controller. Communication over the wireless multi-hop network is then needed to solve the distributed control task. Fig. 7b illustrates a scenario where agents communicate their sensor measurements over the network. However, the general methodology developed in the next sections extends straightforwardly to other scenarios, and we will also discuss such examples throughout the text.

## VII. INTEGRATION AT DESIGN TIME

Building on the basic architecture from the previous section, we now present an integration of the wireless embedded system and the control system at *design time*. Prior to execution, we derive a global schedule that serves the requirements of the control system (see Fig. 5). On the control side, we take the key properties of the wireless embedded system design, come up with a suitable control strategy, and formally prove closed-loop stability of the overall CPS. As will become apparent in the remainder of this section, the nature of the wireless embedded system allows us to use fairly straightforward methods from control theory to design a suitable control strategy and prove stability. The stability proof yields a condition that is computationally cheap to evaluate and can be used to check stability for any LTI system using our wireless embedded system design and control strategy. Moreover, we show in two examples how the support for arbitrary communication patterns greatly simplifies solving distributed control tasks.

To additionally be able to influence the behavior during execution, we further extend the basic architecture and enable the CPS to change between a well-defined set of modes. A mode describes the operation of the overall CPS, including the scheduling of tasks, the control objectives, and the content and period of communication rounds. Since we calculate the modes a priori, we can verify closed-loop stability for each mode and also for switches between any pair of modes.

### A. Wireless Embedded System

We extend the basic wireless embedded system with an offline and an online component. We leverage Time-Triggered Wireless (TTW) [75] to compute different operating modes prior to execution (offline), which includes the order and time offsets of application and communication tasks. To switch between modes and thus adapt to changing application demands during execution (online), we design a mode change protocol.

**Time-Triggered Wireless.** With an increasing number of nodes and possibly multiple control loops being closed over the same wireless network, the scheduling problem, as described in Sec. VI-A, quickly gets more complex, which calls for an automated solution. To this end, we use TTW, a framework tailored to solve this type of scheduling problem [75]. TTW co-schedules task executions and message transmissions together with the communication rounds. It statically synthesizes the schedule of all tasks, messages, and rounds offline by formulating the corresponding mixed-integer linear program. This way, the application's real-time constraints can be guaranteed while minimizing the computational demand at runtime.

**Mode change protocol.** The computed modes are distributed to all nodes prior to execution. This allows us to realize a mode change at runtime by exchanging unique mode identifiers rather than all information that is relevant for a mode. As a result, the communication bandwidth required to realize a mode change becomes extremely small, which reduces end-to-end latencies and energy costs on the communication side.

Our mode change protocol ensures a timely and safe transition from one mode to another. The beacon slot (see Fig. 6), originally only used to signal the start of a communication round, is extended with information about the current mode ID, the next mode ID, and a counter that describes the number of rounds until the mode change becomes active. Nodes in the network can learn about the mode change until the counter reaches 0. In the unlikely event that a node misses all beacons during this time, it enters a resynchronization state where it does not participate in any communication until it receives again a beacon with the new mode ID. This prevents nodes from executing the wrong mode and disturbing the network, which ensures a safe operation. See [7] for more details.

## B. Control System

Next, we outline the control strategy for both remote control and distributed control tasks, which builds on the essential properties of the preceding wireless embedded system design.

**Control design.** We first discuss the remote control scenario illustrated in Fig. 7a. The controller aims to achieve desired (e.g., stable) closed-loop behavior of the physical system by sending appropriate input signals. To this end, we start by designing a controller for the nominal system, that is, without delays and message loss. This can be done using well-known techniques from control theory, such as the linear quadratic regulator (LQR) [76]. We then enhance the control design to cope with constant delays and rare i.i.d. message loss.

We know that messages that are sent over the wireless channel arrive with a delay of one time step (see Fig. 7a). Further, we know when a data packet should be sent in each communication round, and thus we also know when a packet has been lost. The controller can now compensate for the delay and message loss by performing state predictions based on a mathematical model of the physical system,

$$\begin{aligned}\hat{x}(k) &= \theta Ay(k-1) + (1-\theta)A\hat{x}(k-1) + B\hat{u}(k-1) \\ &= \theta Ax(k-1) + (1-\theta)A\hat{x}(k-1) + B\hat{u}(k-1) \quad (3) \\ &\quad + \theta Aw(k-1).\end{aligned}$$

Here,  $\hat{x}(k)$  denotes the predicted state and  $\hat{u}(k)$  the control input computed by the controller.

With  $\hat{x}(k)$ , the controller has an estimate of the current state of the system. However, the control input that is sent over the network will be delayed by another time step. The controller anticipates this delay by making a further prediction and computing the next control input based on this,

$$\hat{u}(k+1) = F(A\hat{x}(k) + B\hat{u}(k)), \quad (4)$$

where  $F$  has been designed to render the nominal system stable (e.g., an LQR design). The input  $\hat{u}(k+1)$  is then sent over the wireless network.

**Stability analysis.** The nature of the wireless embedded system and the control design presented above allow for providing theoretical stability guarantees. In the following, we will give an intuition of how these stability results can be obtained and refer to [6], [7] for detailed descriptions and the full proofs.

Two main ingredients for the stability analysis are properties **P1** and **P2** of the wireless embedded system. Since jitter is negligible, we can accurately describe the dynamics with the discrete-time model developed so far, assuming a fixed sampling interval. The i.i.d. property of message loss further makes it possible to apply well-known tools from the theory of linear matrix inequalities (LMIs) [77, Ch. 9]. In essence, the system is stable as long as the message loss probability is “low enough” and the update interval is “short enough.” For a certain message loss probability and update interval, stability of the overall CPS can be assessed by evaluating a single LMI. The concrete LMI and stability test are given in [7, Thm. 1].

This analysis is valid as long as we stay in a given mode. In the face of mode changes, the analysis needs to be enhanced. A mode change might, for instance, indicate a change of the length of a discrete time step and, thus, different  $A$  and  $B$  matrices. In control theory, this is referred to as a *switched system*. It is well known that a switched system can become unstable even if all subsystems are stable [78]. However, if we ensure that the system, on average, stays in each mode for a sufficient amount of time, called the *average dwell time*, the stability of the subsystems is still sufficient. Thus, as long as the mode changes respect the average dwell time, we can guarantee stability of the overall system. In [7], we discuss how to calculate the dwell times and show in examples that respecting those on average is not overly restrictive.

**Distributed control.** We now turn our attention to distributed control, as depicted in Fig. 7b. We reintroduce agent index  $i$  as we are dealing with multiple systems. While we here assume a local controller to be collocated with each system, information of other systems is needed to fulfill the distributed task. That is, the control input  $u_i(k)$  of each system now not only depends on its local state, but it also needs to take the states of other systems into account. Considering again static linear feedback, the control law can be written as

$$u_i(k) = F_{ii}x_i(k) + \sum_{j \in \Omega_i} F_{ij}\hat{x}_j(k-1), \quad (5)$$

where  $\Omega_i$  denotes the set of all agents whose state is relevant for agent  $i$ . In general, if we want to find the optimal solution for the distributed control problem,  $\Omega_i$  may comprise all other agents. Per property **P3** the wireless embedded systems supports arbitrary communication patterns, and thus we can also cater for the most extreme case where every agent needs information from all other agents (i.e., all-to-all communication).

Similar to the remote control case, the state  $\hat{x}_j(k-1)$  in (5) denotes the current estimate that agent  $i$  has of agent  $j$ 's state. As can be seen from (5), the delay is already incorporated since at time  $k$  the value  $\hat{x}_j(k-1)$  is used. In case of message loss, we use ZOH, i.e., for agent 2 from Fig. 7b we have

$$\hat{x}_2(k-1) = \theta y_2(k-1) + (1-\theta)\hat{x}_2(k-2). \quad (6)$$

In the following, we give two concrete examples of distributed control tasks. In both cases, we present the two-agent setting for illustration as per Fig. 7b. The general framework straightforwardly extends to a larger number of agents, as we demonstrate through testbed experiments in Sec. X.

**Example 1 (Consensus).** We consider a variant of the well-known average consensus problem [79]. Therein, the agents aim to reach the desired state  $x_{\text{des}} = \frac{1}{N} \sum_i x_i$ , which is the average of the individual states  $x_i$  of  $N$  agents. Here, we divide the problem into two parts. First, the agents should agree on a common  $x_{\text{des}}$  via average consensus. Then, this state should be tracked. Thus, communication between agents is needed only in the first part.

A straightforward way to achieve average consensus is to initialize the local  $x_{i,\text{des}}(0)$  of each agent with its current state. Then, in each round, all agents receive the  $x_{j,\text{des}}(k-1)$  from the agents they can communicate with and update their  $x_{i,\text{des}}(k)$  following a standard consensus update law [79]

$$x_{i,\text{des}}(k) = p_{ii}x_{i,\text{des}}(k-1) + \sum_{j \in \Omega_i^j} p_{ij}\hat{x}_{j,\text{des}}(k-1), \quad (7)$$

where  $\Omega_i^j$  denotes the set of all agents  $j$  from which agent  $i$  receives information, and  $p_{ii}$  and  $p_{ij}$  are nonnegative real numbers designed such that  $p_{ii} + \sum_{j \in \Omega_i^j} p_{ij} = 1$ . Consensus can be trivially achieved if all agents can communicate their states to all others. Then, choosing  $p_{ii} = p_{ij} = \frac{1}{N}$  for all  $i, j$  yields consensus after one round. In Sec. X-A, we make this advantage of many-to-all communication concrete by contrasting to an example with nearest-neighbor communication. As a second step, after consensus has been reached at some time  $k^*$ , all agents can implement the control law (analogous to (5))

$$u_i(k) = F_i(x_i(k) - x_{i,\text{des}}(k^*)) \quad (8)$$

in order to track the agreed-upon goal  $x_{i,\text{des}}(k^*) = x_{\text{des}}$ .

The consensus algorithm presented in Example 1 is an intuitive and basic example of distributed control. It solely focuses on agreeing on a common target state and tracking that target state. In real applications, however, systems typically need to find a trade-off between the distributed control task and, for instance, stabilizing themselves locally. One possibility to also account for this is the use of optimal control methods, which we present in the second example.

**Example 2 (Optimal Distributed Control: Synchronization).** As in Example 1, we seek to have  $x_1(k)$  and  $x_2(k)$  evolve as closely as possible, while now trading this goal off with further objectives. To distinguish it from Example 1, we will refer to this problem as synchronization.

As commonly done in optimal control [76], we formulate the different objectives as a quadratic cost function

$$J = \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E} \left[ \sum_{k=0}^{K-1} \sum_{i=0}^2 (x_i(k)^T Q_i x_i(k) + u_i(k)^T R_i u_i(k)) + (x_1(k) - x_2(k))^T Q_{\text{sync}} (x_1(k) - x_2(k)) \right]. \quad (9)$$

The positive definite matrices  $Q_{\text{sync}}$ ,  $Q_i$ , and  $R_i$  indicate our objectives of satisfying the synchronization objective (i.e., keeping the difference between states small by choosing large penalty  $Q_{\text{sync}}$ ), being stable (i.e., keeping the states close to the equilibrium state at  $x_i(k) = 0$ ), and keeping the inputs small, respectively. Using augmented states, this can be brought into standard LQR form and be solved using readily available tools [76]. As a solution, we obtain a static feedback matrix  $F$  and the control input (5) for agent 1 can be written as

$$u_1(k) = F_1 \hat{x}(k) = F_{11} x_1(k) + F_{12} \hat{x}_2(k). \quad (10)$$

In general, the feedback matrix  $F$  is dense; thus, for optimal control, every agent needs information from all other agents. Using our wireless embedded design, this can be straightforwardly supported since information sent by one agent can be received by all others in the wireless network (property P3).

## VIII. INTEGRATION AT RUN TIME

The presented integration of communication and control at design time already fulfills many of the requirements and provides means to solve the associated challenges of future CPS, as outlined in Sec. III. However, decisions about the usage of the wireless medium are taken in an uninformed way, i.e., sensor and control messages are transmitted at the highest rate supported by the wireless embedded system, regardless of whether or not those messages contain relevant information. In wireless CPS, this is undesirable for two main reasons. First, the bandwidth in a wireless network is severely limited, and with that also the number of systems that can participate in a communication round (see challenge C1, limited throughput). Second, the RF transceiver consumes power, which is a significant factor in the overall energy consumption of the hardware platform. For embedded sensors and mobile devices, which are untethered and usually powered by batteries, this is a key concern (see efficiency challenge described in Sec. III).

To mitigate both problems, the presented integration at design time needs to be complemented by an integration at *run time*. As illustrated in Fig. 5, during operation, the control system reasons about its future communication demands and informs the communication system accordingly. To achieve this, recent event-triggered and self-triggered control approaches [59], [60] can be used, where in contrast to traditional periodic or discrete-time control [80] sensor and controller updates occur only when needed. The communication system then uses the information about whether an update is necessary or not to allocate freed resources to lower priority traffic (e.g., status information or additional sensor measurements) or to shut down resources completely for saving energy. We call this integration at run time *control-guided communication*. Below, we outline the main ideas behind control-guided communication and refer the reader to [8] for further details.

### A. Control System

While the model of the wireless CPS derived in Sec. VI is still valid, we need to adapt the control strategy if we envision a control-guided communication. Apart from stabilizing a system or solving a distributed task, the new control strategy has two

additional goals: 1) only use the wireless channel if necessary, and 2) decide about communication demands in advance. To achieve both goals, we employ a self-triggered control strategy. In self-triggered control, the control algorithm decides at each communication instant when to communicate the next time [81]. This information can be included in the data packet that would have been sent anyways in the current communication round. Various self-triggered control algorithms have been proposed in the literature. Here, we present the approach from [8].

We start by defining an error  $e(k)$ , which can, for instance, be the deviation from the equilibrium for a remote control scenario or the deviation between system states for a synchronization task. Next, we set a triggering threshold  $\delta$ . The threshold  $\delta$  is basically the deviation from the control objective that we are willing to tolerate. A straightforward triggering design is then to trigger communication whenever  $e^T(k)e(k) > \delta$ . However, to enable the communication system to adapt to this demand, we need to decide about the next triggering instant in advance. Thus, at time  $k$ , we reason about when we expect the error to exceed the threshold the next time based on all data collected so far. Formally, we find the smallest  $M > 1$  such that

$$\mathbb{E}[e^T(k+M)e(k+M)|\mathcal{D}(k)] > \delta, \quad (11)$$

where  $\mathcal{D}(k)$  is the data available at time  $k$ . The value  $M$  is included in the data packet that is sent over the network.

In the following, we make the triggering design precise for the distributed control scenario from Example 2.

**Example 3** (Optimal Distributed Control: Synchronization (cont.)). *In Example 2, we assumed that agent 2 directly sends its sensor measurements to agent 1. Since each agent is aware of the full feedback matrix  $F$ , agent 2 can also, instead of sending the sensor measurements, compute and communicate the control input  $u_{12}(k) = F_{12}x_2(k)$ . Then, the control law for agent 1 reads*

$$u_1(k) = F_{11}x_1(k) + u_{12}(k). \quad (12)$$

*For this case, we define the error as  $e_{12}(k) := u_{12}(k) - u_{12}(k_\ell)$ , where  $k_\ell$  denotes the last time instant agent 2 transmitted  $u_{12}$  to agent 1. As shown in [8], we can find an analytical expression for (11), which an agent checks at each communication instant. That is, at each communication instant an agent estimates when it needs to communicate the next time and includes this information in the data packet.*

### B. Wireless Embedded System

With this approach, the number of agents participating in the next communication round depends on the current state of the control system. An offline derivation and distribution of communication schedules is not sufficient anymore. Instead, we extend the wireless embedded system design with an online scheduler that disseminates the schedule at the beginning of each communication round, as shown in Fig. 5. This allows us to adapt to the demands signaled by the control system dynamically and to either reallocate resources or to save energy in case the control system does not need all slots in a round.

During operation each agent computes the number of rounds  $M$  until it needs to transmit again control information. The

computed  $M$  is piggybacked onto the messages sent in the data slots (see Fig. 6). The owner of the beacon slot acts as the *network manager*: It collects all communication demands and computes an appropriate communication schedule according to an exchangeable policy. Depending on the application, an appropriate policy can be implemented; for example, an energy-saving policy may shut down all slots that are not needed to meet the communication demands. The communication schedule is sent by the network manager in the beacon slot, containing information about the number of data slots and the mapping of nodes to slots in the current communication round. Because the network manager knows the communication demands of all nodes, it can infer when messages are lost. Since messages contain the future communication demand, the network manager allocates a slot in the next communication round for every lost message. This potentially leads to allocated but unused resources. However, it does ensure that enough resources are allocated to fulfill the communication demands.

## IX. CYBER-PHYSICAL TESTBED

Next to a systematic co-design and a thorough theoretical analysis, it is essential to validate wireless control solutions on realistic cyber-physical testbeds [82]. We presented such a testbed in [9]. In this section, we elaborate on our design choices, the capabilities of the testbed, and the testbed extensions used in this paper compared to the original version.

Fig. 8 shows the layout of the testbed used for the experiments in Sec. X. It consists of 20 DPP wireless nodes, 6 real and heterogeneous physical systems (A–F), and 4 simulated physical systems (G–J) deployed in a robotic laboratory environment of approximately 20 m by 30 m. To analyze the network diameter during testbed deployment, we ran multiple link tests. These tests indicate that for the chosen radio transmit power of -12 dBm we obtain a 3-hop network. Communication is subject to interference from various electronic equipment and other experiments in the same and adjacent rooms.

A crucial design point for a cyber-physical testbed is the choice of a physical system. The physical system should be well understood and have fast dynamics that challenge the control, computing, and networking elements of CPS. For these reasons, we use cart-pole systems in our testbed, as depicted in Fig. 9. Cart-pole systems have been widely studied in control [83], and their dynamics match the timescales of (mechanical) systems envisioned for smart manufacturing, such as drones.

Moreover, the testbed should be versatile in the sense that it accommodates a variety of tasks to evaluate different aspects of the wireless control solution. The most widely studied control task for the cart-pole system is probably the *inverted pendulum*, stabilizing the pole in an upright position through appropriate motions of the cart. Due to the fast dynamics of the cart-pole system, this requires fast and reliable feedback.

For wireless control, scenarios with remote controllers are of particular interest. For example, sensor measurements need to be communicated over multiple hops to a remote controller, which computes and sends back the actuator commands. Our testbed supports such scenarios, as one or multiple pendulums may be stabilized via one or multiple remote controllers.

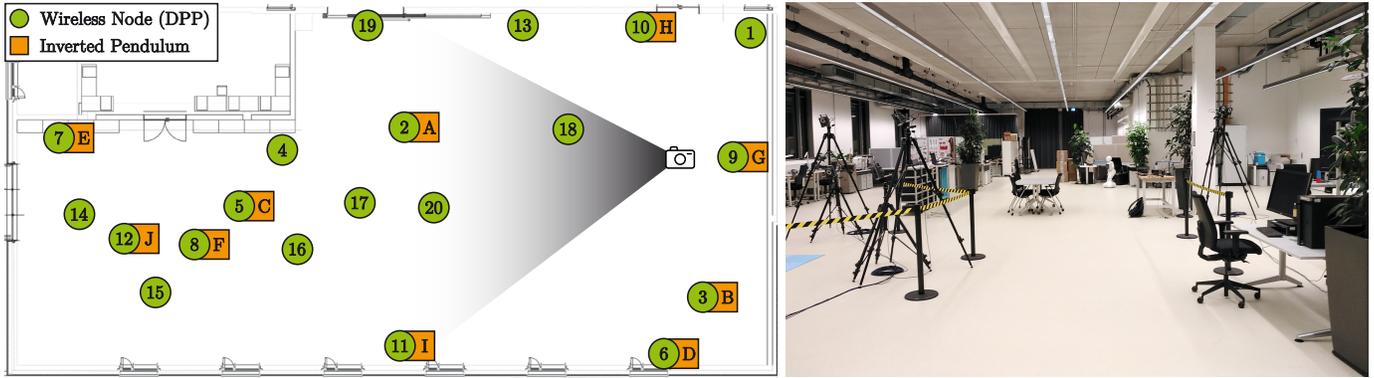


Figure 8. Layout of the cyber-physical testbed with 20 dual processor platform (DPP) wireless nodes, 6 real physical systems (A–F), and 4 simulated physical systems (G–J), deployed in a robotic laboratory of approximately 20 m by 30 m. The viewing angle of the photograph is indicated in the schematic on the left.

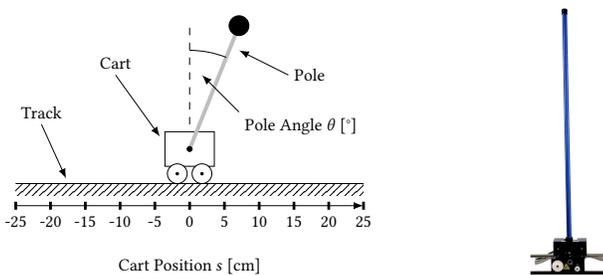


Figure 9. We use multiple heterogeneous cart-pole systems as physical systems in the cyber-physical testbed. In addition to two types of real cart-pole systems with different physical characteristics, the testbed also supports the deployment of simulation models to scale up the number of agents in a cost-efficient way.

Besides remote control, our testbed also supports distributed control tasks. For instance, we may attempt to synchronize the positions of the carts to let them move in concert. The difficulty of this distributed control task can also be varied. In the most straightforward case, we may try to synchronize the cart positions with unmounted poles and thus neglect the stabilization problem. Making it more difficult, we may add the poles but equip each pendulum with a local controller, so stabilization happens locally at faster update intervals than synchronization over the network. In this case, communication is only needed for synchronization and not critical for closed-loop stability. In the most challenging case, both synchronization and stabilization happen over the (same) wireless network.

With this set of supported control tasks, we can mimic various use cases from Fig. 2. A remotely controlled cart-pole system may represent an agent with only sensing and actuation units that is connected directly or over the wireless network with an edge computing unit or a cloud-based service that carries out the control computations. The distributed control scenarios relate to a fleet of agents that need to coordinate their actions, either using local or remote computing units.

Most of the envisioned applications in Sec. I require a large number of agents. It should therefore be possible to scale up the testbed at low cost to challenge the scalability of CPS designs. We accommodate for this by enabling an easy integration of simulated physical systems. A mathematical model of a cart-

pole system can be deployed on any of the DPP nodes in our testbed. One advantage of simulated systems is a better reproducibility because, unlike real physical systems, they always react in the same way given the same control inputs.

In general, distributed control with heterogeneous agents is more challenging than with homogeneous agents [84]. However, this is what is expected in smart manufacturing, for example, when different types of mobile robots need to work together on a given task. To this end, our testbed includes physical systems that can be bought off-the-shelf<sup>2</sup> as well as self-built cart-pole systems; only the former are used in the original testbed [9]. Due to different hardware configurations, also the dynamics of these systems are different. Instead of affording different types of real cart-pole systems, distributed control with heterogeneous agents can also be studied by altering parameters in the mathematical model of the simulated cart-pole systems.

## X. APPLICATION CASE STUDIES

In this section, we use our wireless cyber-physical testbed to illustrate the performance and capabilities of the proposed co-design methodology under realistic conditions. To this end, we consider a series of experimental scenarios that resemble envisioned application use cases of smart manufacturing:

- The AGV example from Sec. II-A includes multiple agents that need to coordinate their actions with each other but are stable on their own. Accordingly, in Sec. X-A, we show how to reliably coordinate multiple stable agents. In particular, we demonstrate the benefits of arbitrary communication patterns supported by our wireless protocol.
- Distributed control becomes more difficult if we consider unstable agents, for example, a team of drones that jointly inspect machines in a factory. Each drone needs to stabilize its flight, while also coordinating its actions with the other drones in the team. Moreover, drones may join or leave the team, or change between different tasks at run time. We address this use case in Sec. X-B, where we show distributed control of unstable agents with mode changes.
- Using the same distributed control scenario, we showcase in Sec. X-C how the integration at run time allows for

<sup>2</sup>We use Quanser's *Linear Servo Base Unit with Inverted Pendulum*.

scheduling additional traffic or saving energy based on the communication resources requested by the control system.

- Besides distributed control, remote control is also of great importance for future smart manufacturing, for instance, when an edge devices controls a mobile robot. Sec. X-D demonstrates remote control of multiple unstable agents.
- Because wireless communication is, in general, more susceptible to message loss than wired communication, we investigate in Sec. X-E the impact of message loss on the control performance of a remotely stabilized agent.

Most of the control strategies in the following experiments rely on a mathematical model of the physical system to be controlled. For the cart-pole systems D and E in Fig. 8 and all simulated systems, we use the same model as in [6], [7]. For the self-built systems A, B, C, and F, we identify the mathematical model using standard methods from system identification [85]. Specifically, we apply a chirp signal and estimate the  $A$  and  $B$  matrices using the method of least squares. This yields

$$A \approx \begin{pmatrix} 1 & -3 \times 10^{-3} & 8 \times 10^{-3} & 5 \times 10^{-4} \\ -2 \times 10^{-3} & 1.02 & -6 \times 10^{-3} & 1 \times 10^{-2} \\ 1 \times 10^{-2} & -0.11 & 0.94 & 2 \times 10^{-2} \\ 1 \times 10^{-2} & -2 \times 10^{-2} & -8 \times 10^{-2} & 1.03 \end{pmatrix}$$

$$B \approx \begin{pmatrix} 3 \times 10^{-4} \\ 9 \times 10^{-4} \\ 8 \times 10^{-3} \\ 1 \times 10^{-2} \end{pmatrix}$$

for a sampling interval of 10 ms. In the following experiments, we always let all 20 DPP nodes participate in the wireless communication, but we use the physical system only for a selected subset of the nodes, referred to as agents.

### A. Average Consensus

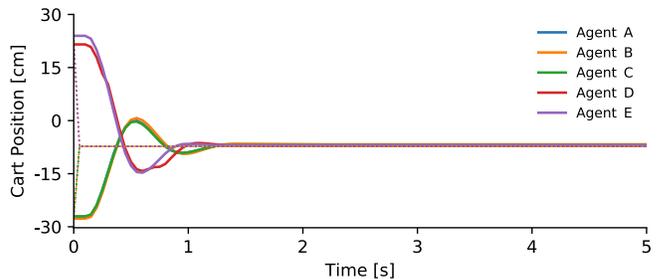
We begin with the average consensus scenario introduced in Sec. VII-B (Example 1). Our experiments show that:

**Finding.** *Thanks to the many-to-all communication support, our wireless control system reaches consensus within one time step using a straightforward and computationally cheap algorithm.*

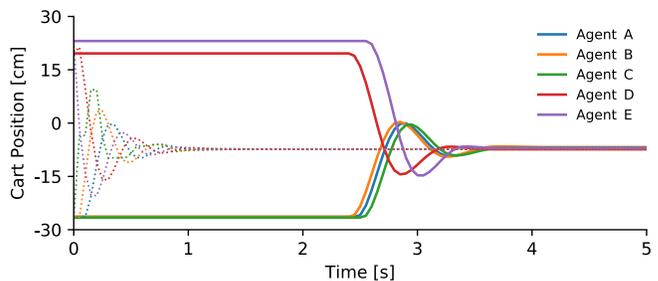
**Setup.** We use 5 agents with real cart-pole systems (A–E in Fig. 8) and different initial cart positions. Agents communicate their state information across the 3-hop wireless network to solve the average consensus problem explained in Sec. VII-B (Example 1). After agreeing on a common desired position, the agents apply control commands to track this position.

We compare two approaches. In the first one, every agent shares its state information with all other agents, referred to as *all-to-all communication*. As discussed in Example 1, we choose  $p_{ii} = p_{ij} = 1/N$  for all  $i, j$  with  $N = 5$ . For the second approach, we implement *nearest neighbor communication*; that is, agent A can only send information to agent B, which can only send information to agent C, and so on. We set all weights  $p$  to  $1/2$ . Using this approach, agents only have partial information, which makes it more difficult to determine when global consensus is reached. Therefore, we require that the desired position of neighbors stays the same for 3 consecutive communication rounds before agents start to track this position.

We set  $F_i = 50$  (see Example 1) and introduce an additional integrator with a gain of 5 to ensure that the desired position is reached. The update interval of the control loop for exchanging



(a) All-to-all communication.



(b) Nearest neighbor communication.

Figure 10. Five agents with real cart-pole systems first aim to reach consensus on their common desired position (dotted lines) by exchanging state information over the wireless network, and then track this cart position (solid lines) using a local control loop. It takes only 100 ms to reach consensus using all-to-all communication compared to 2.5 s steps with nearest neighbor communication.

state information over the wireless network is 100 ms. A second control loop for tracking the desired position becomes effective once consensus is reached. This control loop is independent of the first one and runs at a shorter update interval of 10 ms.

**Results.** Fig. 10 shows the results for all-to-all and nearest neighbor communication; solid lines are the cart positions and dotted lines are the calculated desired positions of the 5 agents over time. We see from the dotted lines in Fig. 10a that with all-to-all communication agents need only 100 ms (i.e., 1 time step) to agree on a common desired position. Using nearest neighbor communication, shown in Fig. 10b, this takes significantly longer: 2.5 s (i.e., 25 time steps). Because the tracking of the desired position is independent of the communication approach, we see a similar performance after consensus is reached. These results clearly showcase the benefits of the support for arbitrary communication patterns of our wireless control system.

### B. Optimal Distributed Control

Increasing the difficulty of the distributed control task, we now turn to a scenario where agents have two control objectives: stabilizing their mounted poles while also synchronizing their cart positions. Because of the self-stabilization task, it is not sufficient for the agents to initially agree on a common desired position. Instead, the agents need to continuously exchange their states and adapt their control input based on their own state and the states of the other agents. To showcase the adaptability of our co-design solution, we additionally let agents switch between different operating modes at run time. We find that:

**Finding.** *The support for many-to-all communication allows for solving distributed control tasks in a straightforward way*

by designing a centralized optimal controller and implementing this controller in a distributed fashion locally on each agent. We thus can successfully synchronize heterogeneous agents and safely switch between different operating modes at run time.

**Setup.** We use 10 agents (A–F with real and G–J with simulated physical systems), and adopt the optimal control design from Sec. VII-B (Example 2) with parameter settings from [6], [7]. Agents exchange state updates over the network every 100 ms for synchronization and run local control loops with an update interval of 10 ms for stabilization. All agents start in a local stabilization mode without synchronization. After 60 s they switch to a synchronization mode, while still stabilizing their poles locally. After 180 s agent A leaves the team (i.e., switches back to local stabilization), corresponding to a second mode change. A third and final mode change is after 300 s when agent B also leaves the team. All other agents keep synchronizing their cart positions until the end of the experiment.

**Results.** Fig. 11 shows the cart positions of all 10 agents over time. During the initial local stabilization mode, the 10 agents move independently of each other with different frequencies and amplitudes (i.e., varying usage of track length). After the first mode change, the agents synchronize their cart positions and start to move in concert with the same frequency and similar amplitudes. We further observe that agents A and B leave the team at the second and third mode change, respectively, without causing any disturbance to the other agents remaining in the team, which continue to move in concert. All 10 agents successfully stabilize their poles across all mode changes.

### C. Online Resource Allocation

In the previous experiment, agents transmit state information in every communication round. However, the actual communication demand varies during execution. Indeed, depending on an agent’s current state and the communication period, it is possible to transmit less often, allowing to dynamically reuse freed slots for other traffic or to shut slots down completely to save energy. Such dynamic resource allocation can be realized through the control-guided communication approach presented in Sec. VIII: When an agent communicates, it already decides about the next time it needs a communication slot and piggybacks this demand information onto its data packet. The network manager collects these communication demands and ensures that all demands are served. Our results show that:

**Finding.** *Through integration at run time, we can reliably synchronize 5 agents while serving additional traffic and saving energy. Using less bandwidth for control comes at the cost of reduced control performance. However, as long as about 25 % of the bandwidth is available for control traffic, the control performance is comparable to the periodic control baseline.*

**Setup.** We use 5 agents (D and E with real as well as G–I with simulated physical systems), whose cart positions are to be synchronized. The control design follows the architecture outlined in Sec. VIII-A (Example 3) with the same parameterization as in [8]. Each communication round consists of 5 data slots, which is sufficient to exchange control data among all 5 agents in every round. The communication period is set to 50 ms. As

before, the local control loops for stabilization run at a shorter update interval of 10 ms. We use an example scheduling policy where the network manager always assigns, if possible, one of the free slots to other traffic while the remaining free slots, if any, are shut down to save energy.

**Results.** Fig. 12 shows for a representative time interval from our experiments how the 5 data slots are used in each communication round, for a self-trigger threshold of  $\delta = 0.03$ . We see that only about one third of the bandwidth is needed for control traffic. The remaining bandwidth is used to serve other traffic or shut down to save energy. Fig. 13 shows the trade-off among control performance (in terms of the root mean squared synchronization error), associated energy costs (in terms of radio duty cycle for control traffic), and fraction of bandwidth available for other traffic. We vary the fraction of bandwidth used for control traffic through different choices of the self-trigger threshold  $\delta$ . For  $\delta = 0$  all agents communicate control data in every round, which we use as a periodic control baseline. We see that using less bandwidth for control leads to a drop in control performance. However, when using only about 25 % of the bandwidth for control, we can still obtain reasonable control performance while being able to serve around 75 % of additional traffic and to reduce energy costs by about 80 % compared with the periodic control baseline.

### D. Reliable Remote Control in the Face of Mobile Agents

Complementing the distributed control scenarios above, we now look at remote control with mobile agents, a highly relevant scenario for future smart manufacturing. We find that:

**Finding.** *We can reliably stabilize two agents with fast dynamics over a 3-hop wireless network despite agent mobility. Apart from the practical demonstration, our co-design methodology also allows us to provide theoretical stability guarantees.*

**Setup.** We use two agents with real cart-pole systems (A and B) and stabilize their poles via a remote controller running on agent 1. Thus, we have two agents that send their state information over the 3-hop wireless network to a remote controller. The controller computes control inputs and sends them back across the network to both the two agents. The control design follows the description in Sec. VII-B, concrete parameters are chosen as in [6], [7]. Communication over the wireless network happens at an update interval of 40 ms. Using a conservative estimate of 0.1 % for the message loss probability, we obtain a theoretical stability guarantee. During the experiment, we pick up the agent running the controller and move around the testbed area.

**Results.** Fig. 14 shows the evolution of the pole angle and the cart position of both agents throughout the experiment. We see that both agents stay in safe regimes over the entire duration of the experiment and never come close to reaching the limits of their tracks. In other words, we can safely stabilize them via the wireless multi-hop network. Moving the controller node during the experiment has no impact on control performance<sup>3</sup> as the operation of our wireless protocol is not affected by changes in the network as long as agents remain connected [69], [71].

<sup>3</sup>A video of such an experiment can be found at <https://youtu.be/19xPHjnobkY>.

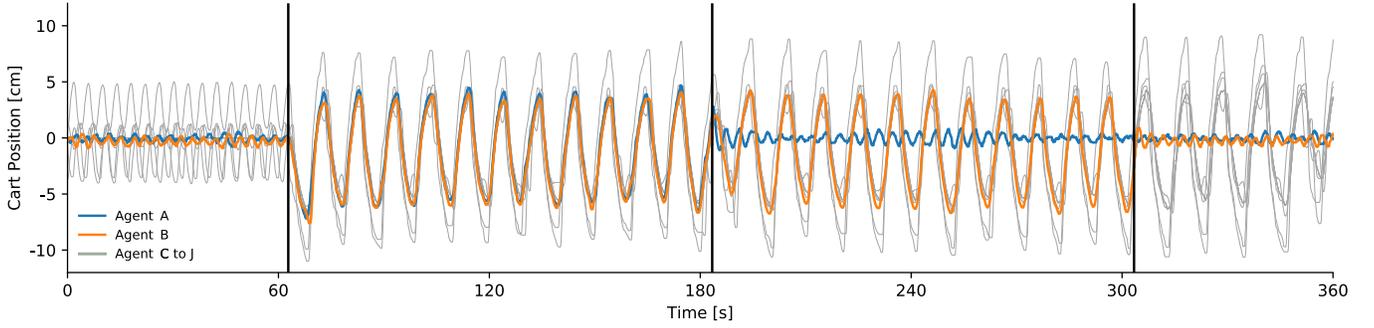


Figure 11. Cart positions over time as agents stabilize their poles locally. Black lines indicate mode changes. After the first mode change, all 10 agents start to synchronize their cart positions over a 3-hop wireless network. At the second and third mode change, one agent each drops out of the synchronization task, while the other agents keep synchronizing their cart positions. Agents successfully synchronize and smoothly switch from one operating mode to another.

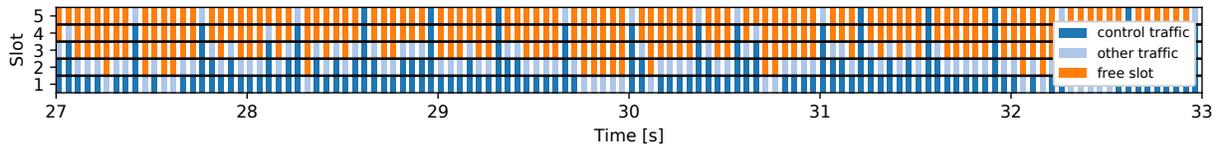


Figure 12. Bandwidth utilization with control-guided communication over time, recorded during an experiment with 5 agents that synchronize their cart positions over a 3-hop wireless network. Because of the self-triggered control design, the bandwidth used for control traffic in each communication round varies between 0 and 5 slots. In almost every communication round one slot is used for other traffic, while the remaining free slots are shut down to save energy.

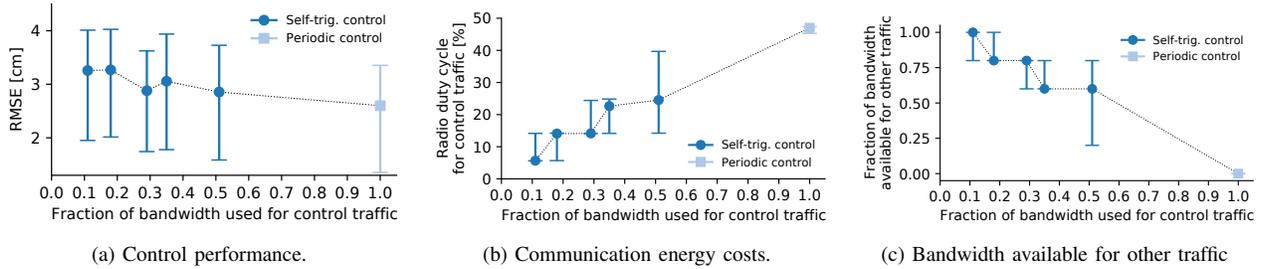


Figure 13. Trade-off between control performance, the associated communication energy costs, and the ability to serve other traffic for different fractions of bandwidth available for control traffic. Graphs show the median and the 25th and 75th percentiles. Control performance decreases when less bandwidth is used for control traffic. On the other hand, bandwidth that is not used for control can be used to save significant communication energy or to serve other traffic.

### E. Reliable Wireless Control Despite Node Failures

Great efforts are being made toward “ultra-reliable” wireless communication, targeting message loss rates of  $10^{-9}$  or even lower. We demonstrate that, given a proper co-design of the communication and control systems, the resulting CPS can deal with much higher loss rates. In particular, we find that:

**Finding.** We can stabilize two agents via a remote controller over a multi-hop wireless network despite 10% message loss.

**Setup.** We consider the setup from the previous experiment, but artificially introduce message loss. That is, in addition to the possible message loss over the wireless network, we let agent B drop 10% of control commands according to an i.i.d. Bernoulli process. Even at such high message loss we expect the agent to remain stable according to our theoretical analysis.

**Results.** The resulting position and angle trajectories are nearly indistinguishable from the ones shown in Fig. 14 without artificial message loss. Thus, we do not plot them here, and conclude that we can safely stabilize the agent despite 10%

of message loss. Differences in the control performance can be seen, for example, by looking at the distances traveled by the carts during the experiment. Fig. 15a shows the traveled distances from the previous experiment without artificial message loss. The results in Fig. 15b with artificial message loss show that agent B compensates the message loss by moving slightly more. We draw three conclusions from these results: (i) If the control and communication systems are carefully designed in tandem, message loss rates well above typically targeted rates can be tolerated. (ii) Instead of focusing on “ultra-reliable” wireless communication, focus should be put on designing wireless solutions for which the assumption of i.i.d. message loss is a valid approximation. (iii) Message loss has an impact on control performance; however, even for loss rates of up to 10%, the decrease in control performance is not severe, as long as the losses are i.i.d. The i.i.d. assumption is highly valid for our wireless solution, as per property P2.

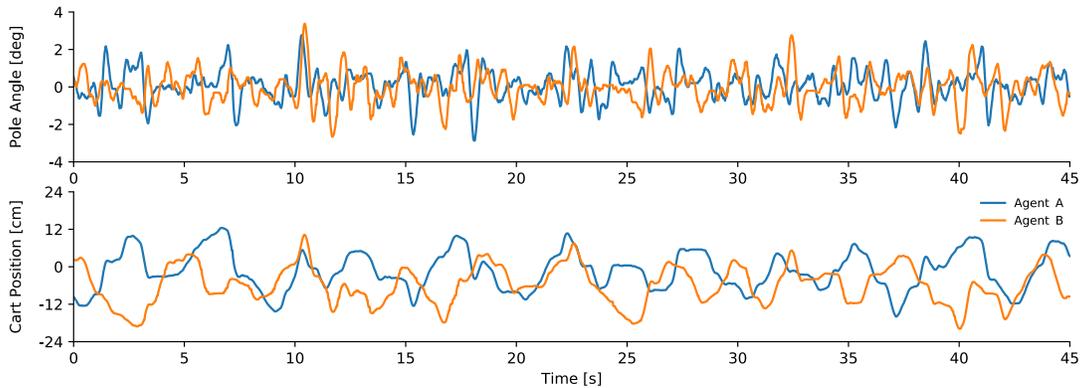


Figure 14. Remote stabilization of two agents with real cart-pole systems over a 3-hop wireless network. During the experiment, the agent running the remote controller is moved around the testbed area. *The poles of both agents are safely stabilized, and the control performance is not affected by agent mobility.*

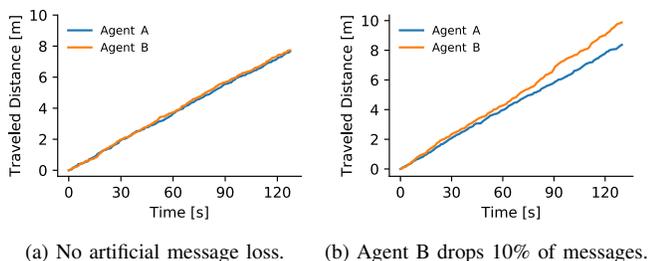


Figure 15. Traveled distance of two agents while stabilizing their poles in remote control experiments. *Agent B travels slightly farther when it artificially drops messages, which indicates a lower control performance. However, the drop in control performance is not severe, and stability is still guaranteed.*

## XI. OPEN CHALLENGES AND OPPORTUNITIES

Starting from a perspective on future smart manufacturing, we derived several requirements and associated challenges, such as predictability of operation in terms of functional correctness and timeliness, as well as the efficiency of operation. Constructing systems that satisfy these requirements is particularly difficult in systems that have low and non-deterministically changing resource availabilities and require a highly dynamic operation. We presented a design methodology based on a holistic approach that integrates distributed feedback control and multi-hop wireless communication, both at design and at run time (cf. Fig. 5). In particular, through integration at design time, we were able to achieve fast distributed feedback control on the order of tens of milliseconds over multi-hop networks with stability guarantees also in the face of mode changes. Further savings and flexible reallocation of resources were achieved by having controllers inform the communication system at run time about their current need for communication. Taken together, integration at design time and run time allows for high-performance distributed control while making efficient use of the available resources, as well as for adaptive behavior such as switching seamlessly between modes of operation.

Through this holistic approach, we were able to address many challenges that currently prevent wireless technology from being adopted in the manufacturing industry. However, to fully realize the smart manufacturing vision, there are many more

challenges to be overcome. Below, we outline some important directions for future work, structured along the main challenges of dependability, adaptability, and efficiency (see Sec. III).

### A. Dependability

The presented solution allows for reliable wireless communication (fast updates, highly uncorrelated packet drops, negligible jitter). Further, through the integration of control and communication at design time, we can provide a formal proof of stability, which is a base requirement for any feedback control system. The theoretical guarantee encompasses all components of the CPS (computation, communication, control, LTI physical process), mode changes, and dynamic network topologies.

**Formal guarantees.** The type of theoretical guarantee given herein concerns the dynamical stability of the closed-loop control system with all main components of the CPS. It is desirable to complement this with further guarantees and certification. For example, *formal verification* of protocols, algorithms, or entire CPS for certain safety and liveness properties is highly relevant. While approaches like model checking [86] hinge on the derivation of an accurate model and can be challenging to apply for complex systems, the automatic design of software components (e.g., scheduler, controller, protocols) that are *correct by construction* is an orthogonal approach. Also in the realm of feedback control, some aspects have not been addressed yet. For instance, extending our current stability guarantees to incorporate further uncertain components, such as (partly) unknown or data-driven physical system models and learning-based controllers, is highly relevant, especially in the context of the envisioned adaptability in smart manufacturing.

**Fault detection and identification.** The larger scale of future smart manufacturing systems in terms of number of components and their interconnection complexity makes the prediction, early detection, and identification of faults highly demanding. It is necessary to have means to observe the operation of the various components by collecting key performance indicators and combine the data toward information and knowledge concerning the overall system state, causes of failures, as well as predictive maintenance. Whereas many results are available

using advanced methods for data analysis and machine learning, it is unclear how to employ these methods and the associated data collection in low-resource environments. As the occurrence of failures in a complex system is inevitable, the careful design of contingency and fault handling strategies is another core task. Here, the flexibility and redundancy envisioned in smart manufacturing (e.g., in terms of agents) can be of help.

**Distributed architectures.** With the help of smart manufacturing, production processes are becoming significantly more flexible and versatile while minimizing its operators' management effort. In traditional manufacturing systems, the collaboration between machines and other CPS devices is mainly controlled by a central management system. Such a concept has a high risk of failure since the overall system's availability and reliability largely depend on the underlying communication network and the centralized management unit's availability. At the same time, this centralized approach is no longer manageable with the increasing number of collaborating CPS devices and the vast complexity managing a large number of interconnected devices. The same observation holds for centralized scheduling and coordination to synchronize and schedule the computational as well as communication tasks. Better scalability and fault tolerance concepts are needed toward fully distributed approaches without a central scheduler or controller while still being able to provide formal guarantees for the correct operation. As an attractive middle ground, such distributed operation may also be temporary or limited to certain parts of the hierarchy (e.g., a specific automation cell).

**Security and privacy.** Deliberately, the discussion did not touch on any security- or privacy-related challenges in the design of future smart manufacturing systems. One may argue that well-known security concepts from the conventional computing domain can be simply applied and are sufficient. Obviously, this is not the case when considering problems like side-channel attacks via the physical embedding of sensors and actuators, key-management over the whole lifetime of the large number of distributed components, novel attack scenarios exploring the low-power wireless connectivity, as well as attacking the overall system functionality by manipulating sensor readings as available to the control algorithms. Likewise, when sharing data and methods across users, factories, and possibly companies, important privacy concerns must be answered. Novel security and privacy methods are thus needed to address the unique challenges of future smart manufacturing systems.

### *B. Adaptability*

Because of the underlying synchronous transmissions, the presented wireless control solution is ignorant of any graph structure of the network and requires no routing. Hence, the solution is inherently adaptive to dynamic nodes, which is key to provide the mobility needed in many smart manufacturing scenarios. Furthermore, the current design and stability guarantees provide for some adaptability when switching between different operation modes. The set of possible modes, however, must be known in advance.

**Enhanced adaptability at run time.** While in some scenarios such as highly safety-critical ones, it is plausible that the set of possible application requirements and thus operation modes is fixed and known beforehand, it is vital for the vision of smart manufacturing to enhance adaptability also to situations that are previously unknown at design time. For example, when robots with new manufacturing capabilities are added or for highly individualized tasks, the smart manufacturing system must adapt at run time. The goals of instantaneous adaptation to unknowns while giving guarantees throughout operation are obviously conflicting in general. Providing meaningful solutions and trade-offs tailored to domain and problem classes is thus a formidable task for basic research. A promising direction for enhanced adaptability is the combination of recent advances in machine learning and data science with control and communication, as discussed next.

**Learning-based control.** In the complete vision of smart manufacturing, properties of the computation and communication environment, as well as of the controlled physical processes, are constantly changing. In addition, when scaling the complexity and size of smart manufacturing systems, it is increasingly unfeasible to manually derive appropriate plant and control algorithms that integrate computation and (wireless) communication, even if they are parameterizable. Therefore, recent approaches attempt to combine classical control-theoretic approaches with data-driven concepts, e.g., machine learning. Again, it is unclear how these novel design approaches that are partially based on massive data collection can be viably implemented in low-resource environments, and how dependability requirements can be met alongside.

### *C. Efficiency*

The presented solution builds on low-power wireless communication technology, which is in contrast, e.g., to envisioned 5G communication. While there are potential and huge expectations associated with future 5G solutions, the industry is also facing significant issues such as high running costs, network infrastructure not being owned by the manufacturer, insufficient data protection, and current inability to meet certain requirements on latency, jitter, and reliability. Here, we presented an alternative solution that addresses similar use cases as being discussed in the context of the 5G vision, yet is based on existing commodity hardware.

In addition to making use of low-power hardware, we further improved efficiency utilizing resource-aware control algorithms and the integration of control and communication at run time. Through the concept of control-guided communication, communication bandwidth can be flexibly allocated at run time to the control processes that are in need or be saved otherwise. While constituting only a first approach toward such integration at run time, we were already able to demonstrate significant resource savings and flexible reallocation.

**Truly event-triggered wireless control.** The presented solution for control-guided communication is based on a self-triggered control approach. With self-triggered control, the control algorithm decides at the time of the current control

computation, when the next control update needs to happen. Including this information in the current data packet, the network manager can plan resource allocation in advance and, in this way, ensure efficient use of communication bandwidth. The downside of the self-triggered approach, however, is that the need for communication must be planned in advance. For better efficiency and also adaptability, it would be desirable to instead decide *instantaneously* about the need for communication. While in the control community, many so-called event-triggered estimation and control approaches have been developed in the last two decades [59], [60], it remains largely unclear whether and how these can be *integrated* with the communication system and indeed result in demonstrable resource reallocation, savings, or other advantages for wireless systems in practice.

**Advances in wireless communication.** Scaling up the network diameter, number of agents, or supporting higher update rates may require faster low-power wireless physical layers (e.g., Bluetooth Low Energy (BLE 5), ultra-wide band communication). Further, protocol innovations such as network coding have the potential of increasing the effective capacity of the network and thus enhancing efficiency. These and other advances on the physical and protocol layers will directly benefit wireless control approaches as the one presented herein.

#### D. Real-world Deployment

All core aspects of our presented wireless control solution have been deployed and demonstrated on the CPS testbed (Sec. IX), which we developed for this research. We deliberately chose the components of the testbed to be representative of many smart manufacturing use cases, as well as to pose a real challenge for the state of the art in wireless control. For instance, we opted for inverted pendulums as physical processes because they represent prototypical control tasks and *require* fast feedback for their operation at their unstable equilibrium. In addition to many hours of testing in the lab, we have also deployed our CPS testbed on multiple occasions outside the laboratory such as during a demonstration at the *2019 Cyber-Physical Systems and Internet-of-Things Week* in Montreal, Canada.<sup>4</sup> Despite being in a room with hundreds of conference attendees and much other wireless equipment operating in parallel, we were able to run the coordination and remote stabilization demos similar to those presented in Sec. X-B and Sec. X-D for about three hours without a single failure.

At the same time, we are convinced that more real-world demonstrators and deployments will be essential on the way of making wireless control a reliable and accepted technology in smart manufacturing. In addition to theoretical proofs based on rigorous analysis, it is key to systematically evaluate the end-to-end system with all hardware and software components on physical platforms and real-world wireless networks in scenarios that resemble as close as possible the targeted smart manufacturing use cases. We as a community thus need more of such testbeds with robots, flying drones, conveyor belts, etc. Ideally, these testbeds are open and extensible, and researchers around the world can use them. Besides practical validation,

such testbeds are also needed to enable comparability and reproducibility of research claims and results, and they may serve as a common benchmarking platform. To this end, efforts on testbeds in control and communication should no longer run separately, but be united and work toward a common goal. Ultimately, the final step is then, of course, real-world trials in actual manufacturing and production plants. But without prior demonstrators, the companies will be unlikely to allow this.

#### ACKNOWLEDGEMENTS

We would like to thank Romain Jacob for his contributions to the research detailed in this paper. We would also like to thank Harsoveet Singh, Joel Bessekon Akpo, and Felix Grimminger for their help with the testbed, and the TEC group at ETH Zurich for making the design of the DPP platform available to the public.

#### REFERENCES

- [1] S. Jeschke, C. Brecher, H. Song, and D. B. Rawat, *Industrial Internet of Things: Cybermanufacturing Systems*, ser. Springer Series in Wireless Technology. Springer, 2017.
- [2] G. McMillan, "Is wireless process control ready for prime time?" *Control*, May 2009, <http://modelingandcontrol.com/repository/WirelessPrimeTime.pdf>.
- [3] N. A. Bhatti, M. H. Alizai, A. A. Syed, and L. Mottola, "Energy harvesting and wireless transfer in sensor network applications: Concepts and experiences," *ACM Transactions on Sensor Networks*, vol. 12, no. 3, 2016.
- [4] A. Ahlen, J. Akerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, "Toward wireless control in industrial process automation: A case study at a paper mill," *IEEE Control Systems Magazine*, vol. 39, no. 5, pp. 36–57, 2019.
- [5] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *IEEE International Conference on Industrial Informatics (INDIN)*, 2011.
- [6] F. Mager, D. Baumann, R. Jacob, L. Thiele, S. Trimpe, and M. Zimmerling, "Feedback control goes wireless: Guaranteed stability over low-power multi-hop networks," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 97–108.
- [7] D. Baumann, F. Mager, R. Jacob, L. Thiele, M. Zimmerling, and S. Trimpe, "Fast feedback control over multi-hop wireless networks with mode changes and stability guarantees," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 2, p. 18, 2019.
- [8] D. Baumann, F. Mager, M. Zimmerling, and S. Trimpe, "Control-guided communication: Efficient resource arbitration and allocation in multi-hop wireless control systems," *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 127–132, 2020.
- [9] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe, "Evaluating low-power wireless cyber-physical systems," in *IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, 2018, pp. 13–18.
- [10] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, B. Kärcher *et al.*, "Reference architecture model Industrie 4.0 (RAMI4.0)," *ZVEI und VDI, Status report*, 2015.
- [11] S.-W. Lin, B. Miller, J. Durand, G. Bleakley, A. Chigani, R. Martin, R. Murphy, and M. Crawford, "The Industrial Internet of Things volume G1: reference architecture," *Industrial Internet Consortium*, pp. 10–46, 2017.
- [12] Y. Lu, K. C. Morris, and S. Frechette, "Current standards landscape for smart manufacturing systems," *National Institute of Standards and Technology, NISTIR*, vol. 8107, p. 39, 2016.
- [13] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, p. 3159805, 2016.
- [14] O. Maghazei and T. Netland, "Drones in manufacturing: exploring opportunities for research and practice," *Journal of Manufacturing Technology Management*, 2019.

<sup>4</sup>This video gives an impression: <https://youtu.be/AtULmfGkVCE>.

- [15] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.
- [16] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, 2014.
- [17] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, 2016.
- [18] R. Sabella, A. Thuelig, M. Carrozza, and M. Ippolito, "Industrial automation enabled by robotics, machine intelligence and 5G," *Ericsson Technology Review on 5G and Industrial Automation*, pp. 2–12, 2018.
- [19] J. Abbenseth, F. G. Lopez, C. Henkel, and S. Dörr, "Cloud-based cooperative navigation for mobile service robots in dynamic industrial environments," in *Proceedings of the Symposium on Applied Computing*, 2017, pp. 283–288.
- [20] Z. Pei, Z. Deng, S. Xu, and X. Xu, "Anchor-free localization method for mobile targets in coal mine wireless sensor networks," *Sensors*, vol. 9, no. 4, pp. 2836–2850, 2009.
- [21] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 2, p. 10, 2009.
- [22] J. Morton, "Underground drilling advances improve productivity, safety while cutting costs," *Engineering and Mining Journal*, vol. 219, no. 1, pp. 40–44, 2018.
- [23] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, June 2010, pp. 731–736.
- [24] J. Rüth, F. Schmidt, M. Serror, K. Wehrle, and T. Zimmermann, "Communication and networking for the industrial internet of things," in *Industrial Internet of Things*. Springer, 2017, pp. 317–346.
- [25] J. M. Fowler and R. D'Andrea, "A formation flight experiment," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 35–43, 2003.
- [26] J. Lunze, *Feedback control of large-scale systems*. Prentice Hall New York, 1992.
- [27] J. Manyika and H. Durrant-Whyte, *Data fusion and sensor management: a decentralized information-theoretic approach*. Prentice Hall PTR, 1995.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [29] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *arXiv preprint arXiv:1912.09789*, 2019.
- [30] U. Wetzker, I. Splitt, M. Zimmerling, C. A. Boano, and K. Römer, "Troubleshooting wireless coexistence problems in the industrial internet of things," in *IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, 2016.
- [31] Siemens AG, "5G communication networks: Vertical industry requirements," 2016.
- [32] Z. A. G. Electrical and E. M. Association, "5G for connected industries and automation (white paper - second edition)," 2019.
- [33] J. A. Preiss, W. Hönig, G. S. Sukhatme, and N. Ayanian, "CrazySwarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [34] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.
- [35] J. Araújo, A. Anta, M. Mazo, J. Faria, A. Hernandez, P. Tabuada, and K. H. Johansson, "Self-triggered control over wireless sensor and actuator networks," in *Proc. of DCOSS*, 2011.
- [36] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, 2014.
- [37] J. Eker, A. Cervin, and A. Hörjel, "Distributed wireless control using bluetooth," in *IFAC Conference on New Technologies for Computer Control (NTCC)*, 2001.
- [38] N. J. Ploplys, P. A. Kawka, and A. G. Alleyne, "Closed-loop control over wireless networks," *IEEE Control Systems Magazine*, vol. 24, no. 3, 2004.
- [39] N. W. Bauer, S. B. Van Loon, N. Van De Wouw, and W. M. Heemels, "Exploring the boundaries of robust stability under uncertain communication: An ncs toolbox applied to a wireless control setup," *IEEE Control Systems Magazine*, vol. 34, no. 4, 2014.
- [40] S. Heshmati-Alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, and K. J. Kyriakopoulos, "A self-triggered visual servoing model predictive control scheme for under-actuated underwater robotic vehicles," in *IEEE Int. Conf. on Robotics and Automation*, 2014.
- [41] C. Santos, M. Mazo Jr, and F. Espinosa, "Adaptive self-triggered control of a remotely operated p3-dx robot: Simulation and experimentation," *Robotics and Autonomous Systems*, vol. 62, no. 6, 2014.
- [42] C. Santos, F. Espinosa, E. Santiso, and M. Mazo, "Aperiodic linear networked control considering variable channel delays: Application to robots coordination," *Sensors*, vol. 15, no. 6, 2015.
- [43] A. Stanoev, A. Ajiz, A. Portelli, and M. Baddeley, "Closed-loop control over wireless – remotely balancing an inverted pendulum on wheels," in *International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2020.
- [44] A. Hernandez, J. Faria, J. Araújo, P. Park, H. Sandberg, and K. H. Johansson, "Inverted pendulum control over an IEEE 802.15.4 wireless sensor and actuator network," in *European Conference on Wireless Sensor Networks (EWSN)*, 2011.
- [45] M. Ceriotti *et al.*, "Is there light at the ends of the tunnel? wireless sensor networks for adaptive lighting in road tunnels," in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011.
- [46] A. Saifullah, S. Sankar, J. Liu, C. Lu, R. Chandra, and B. Priyantha, "Capnet: A real-time wireless management network for data center power capping," in *IEEE Real-Time Systems Symposium (RTSS)*, 2014.
- [47] Y. Ma and C. Lu, "Efficient holistic control over industrial wireless sensor-actuator networks," in *Proc. of the IEEE International Conference on Industrial Internet (ICII)*, 2018.
- [48] T. Yang, H. Yu, M. Fei, and L. Li, "Networked control systems: a historical review and current research topics," *Measurement and Control*, vol. 38, no. 1, pp. 12–16, 2005.
- [49] J. Lunze and L. Grüne, "Introduction to networked control systems," in *Control theory of digitally networked dynamic systems*. Springer, 2014, pp. 1–30.
- [50] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
- [51] J. Baillieul and P. J. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.
- [52] J.-P. Thomesse, "Fieldbus technology in industrial automation," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1073–1101, June 2005.
- [53] S. Trimpe and R. D'Andrea, "The balancing cube: A dynamic sculpture as test bed for distributed estimation and control," *IEEE Control Systems Magazine*, vol. 32, no. 6, pp. 48–75, Dec. 2012.
- [54] B. Li, Y. Ma, T. Westenbroek, C. Wu, H. Gonzalez, and C. Lu, "Wireless routing and control: A cyber-physical case study," in *ACM/IEEE International Conference on Cyber-Physical Systems (ICCP)*, 2016.
- [55] Y. Ma, D. Gunatilaka, B. Li, H. Gonzalez, and C. Lu, "Holistic cyber-physical management for dependable wireless control systems," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 1, 2018.
- [56] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, 2007.
- [57] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems - a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, 2013.
- [58] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, 2016.
- [59] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 3270–3285.
- [60] M. Miskowicz, *Event-based control and signal processing*. CRC press, 2018.
- [61] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Comput. Surv.*, vol. 22, no. 4, p. 299?319, Dec. 1990. [Online]. Available: <https://doi.org/10.1145/98163.98167>
- [62] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, 2000.
- [63] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sens. Netw.*, vol. 6, no. 2, 2010.

- [64] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, 1999.
- [65] F. Sutton, M. Zimmerling, R. Da Forno, R. Lim, T. Gsell, G. Giannopoulou, F. Ferrari, J. Beutel, and L. Thiele, "Bolt: A stateful processor interconnect," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2015.
- [66] C. A. Boano, S. Duquennoy, A. Förster, O. Gnawali, R. Jacob, H. Kim, O. Landsiedel, R. Marfievici, L. Mottola, G. P. Picco, X. Vilajosana, T. Watteyne, and M. Zimmerling, "Iotbench: Towards a benchmark for low-power wireless networking," in *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, 2018, pp. 36–41.
- [67] M. Schuß, C. A. Boano, M. Weber, and K. Römer, "A competition to push the dependability of low-power wireless protocols to the edge," in *Proceedings of the 14<sup>th</sup> International Conference on Embedded Wireless Systems and Networks (EWSN)*. Junction Publishing, Feb. 2017, pp. 54–65.
- [68] C. A. Boano, M. Schuß, and K. Römer. (2017) EWSN dependability competition: Experiences and lessons learned. [Online]. Available: <https://iot.ieee.org/newsletter/march-2017/ewsn-dependability-competition-experiences-and-lessons-learned>
- [69] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with glossy," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011.
- [70] M. Zimmerling, L. Mottola, and S. Santini, "Synchronous transmissions in low-power wireless: A survey of communication protocols and network services," *ACM Computing Surveys*, 2020, accepted for publication.
- [71] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1714. [Online]. Available: <https://doi.org/10.1145/2426656.2426658>
- [72] M. Zimmerling, F. Ferrari, L. Mottola, and L. Thiele, "On modeling low-power wireless protocols based on synchronous packet transmissions," in *Proceedings of the 21st IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2013.
- [73] J. Karschau, M. Zimmerling, and B. M. Friedrich, "Renormalization group theory for percolation in time-varying networks," *Scientific Reports*, vol. 8, 2018.
- [74] A. Cervin, "Integrated control and real-time scheduling," Ph.D. dissertation, Lund University, 2003.
- [75] R. Jacob, L. Zhang, M. Zimmerling, J. Beutel, S. Chakraborty, and L. Thiele, "The time-triggered wireless architecture," in *32nd Euromicro Conference on Real-Time Systems (ECRTS)*, 2020.
- [76] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Prentice Hall, 2007.
- [77] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Siam, 1994, vol. 15.
- [78] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.
- [79] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [80] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*. Courier Corporation, 2013.
- [81] A. Anta and P. Tabuada, "To sample or not to sample: Self-triggered control for nonlinear systems," *IEEE Transactions on automatic control*, vol. 55, no. 9, pp. 2030–2042, 2010.
- [82] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2015.
- [83] O. Boubaker, "The inverted pendulum: A fundamental benchmark in control theory and robotics," in *International Conference on Education and e-Learning Innovations*, 2012, pp. 1–6.
- [84] J. Lunze, "Synchronization of heterogeneous agents," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2885–2890, 2012.
- [85] L. Ljung, *System Identification: Theory for the User*. Prentice Hall PTR, 1999.
- [86] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.



**Dominik Baumann** received the Dipl.-Ing. degree in electrical engineering from TU Dresden, Germany, in 2016. He is currently a joint PhD student with the Intelligent Control Systems Group at the Max Planck Institute for Intelligent Systems, Tübingen, Germany and the Division of Decision and Control Systems, KTH Stockholm, Sweden.

His research interests include the control of cyber-physical systems and the interconnection of control theory and machine learning. More info at <https://is.mpg.de/~dbaumann>.



**Fabian Mager** received the Dipl.-Medieninf. degree in media computer science from TU Dresden, Germany, in 2015. He is currently a PhD student in the Networked Embedded Systems Lab within the Center for Advancing Electronics at TU Dresden.

His research interests include cyber-physical systems and wireless communication, especially in the embedded low-power domain. More info at [https://nes-lab.org/wordpress/?team\\_member=fabian-mager](https://nes-lab.org/wordpress/?team_member=fabian-mager).



**Ulf Wetzker** holds a Diploma degree in computer science and systems engineering (2008) from TU Ilmenau, Germany. Since October 2008 he works as a research scientist at the Fraunhofer Institute for Integrated Circuits IIS, Division Engineering of Adaptive Systems EAS where he joined the Industrial Wireless Communication group in 2012.

His research interests are in the area of wireless communication systems, data analytics and machine learning, with a special focus on anomaly detection and root cause analysis in wireless networks. More info at [https://www.eas.iis.fraunhofer.de/en/research\\_topics/wireless-networked\\_automation.html](https://www.eas.iis.fraunhofer.de/en/research_topics/wireless-networked_automation.html).



**Lothar Thiele** is professor for Computer Engineering at ETH Zurich, Switzerland. His research interests include models, methods, and software tools for the design of embedded systems, Internet of Things, cyber-physical systems, sensor networks, embedded software, and bioinspired optimization techniques. Lothar Thiele is associate editor of INTEGRATION – the VLSI Journal, Journal of Signal Processing Systems, IEEE Transaction on Industrial Informatics, Journal of Systems Architecture, IEEE Transactions on Evolutionary Computation, Journal of Real-Time

Systems, ACM Transactions on Sensor Networks, ACM Transactions on Cyber-Physical Systems, and ACM Transaction on Internet of Things.

In 1986, he received the Dissertation Award of the Technical University of Munich, in 1987, the Outstanding Young Author Award of the IEEE Circuits and Systems Society, in 1988, the Browder J. Thompson Memorial Award of the IEEE, and in 2000–2001, the IBM Faculty Partnership Award. In 2004, he joined the German Academy of Sciences Leopoldina. In 2005, he was the recipient of the Honorary Blaise Pascal Chair of University Leiden, The Netherlands. Since 2009, he is a member of the Foundation Board of Hasler Foundation, Switzerland. Since 2010, he is a member of the Academia Europaea. In 2013, he joined the National Research Council of the Swiss National Science Foundation (SNF). Lothar Thiele received the EDAA Lifetime Achievement Award in 2015. Since 2017, he is Associate Vice President of ETH for Digital Transformation. In 2019, he joined the Foundation Board of Technopark Zurich as president. More info at <https://people.ee.ethz.ch/~thiele/>.



**Marco Zimmerling** holds a Diploma degree in computer science (2009) from TU Dresden, Germany, and a Ph.D. degree in computer engineering (2015) from ETH Zurich, Switzerland. Since November 2015 he leads the Networked Embedded Systems Lab within the Center for Advancing Electronics at TU Dresden. His research interests are in the general area of cyber-physical systems, with a special focus on dependable and sustainable wireless embedded systems. Dr. Zimmerling received the 2015 ACM SIGBED Paul Caspi Memorial Dissertation Award,

the 2016 EDAA Outstanding Dissertation Award, and Best Paper Awards at ACM/IEEE ICCPS 2019, ACM SenSys 2013, and ACM/IEEE IPSN 2011. More info at <https://wwwpub.zih.tu-dresden.de/~zimmerl/>.



**Sebastian Trimpe** (M'12) received the B. Sc. degree in general engineering science and the M.Sc. degree (Dipl.-Ing.) in electrical engineering from Hamburg University of Technology, Hamburg, Germany, in 2005 and 2007, respectively, and the Ph.D. degree (Dr. sc.) in mechanical engineering from ETH Zurich, Zurich, Switzerland, in 2013. Since 2020, he is a full professor at RWTH Aachen University, Germany, where he heads the Institute for Data Science in Mechanical Engineering. Before, he was an independent Research Group Leader at the Max Planck Institute

for Intelligent Systems in Stuttgart and Tübingen, Germany. His main research interests are in systems and control theory, machine learning, networked and autonomous systems.

Dr. Trimpe is recipient of several awards, among others, the triennial IFAC World Congress Interactive Paper Prize (2011), the Klaus Tschira Award for achievements in public understanding of science (2014), the Best Demo Award of the International Conference on Information Processing in Sensor Networks (2019), and the Best Paper Award of the International Conference on Cyber-Physical Systems (2019). More info at <https://is.mpg.de/~strimpe>.