

Control-guided Communication: Efficient Resource Arbitration and Allocation in Multi-hop Wireless Control Systems

Dominik Baumann^{1,*}, Fabian Mager^{2,*}, Marco Zimmerling², and Sebastian Trimpe¹

Abstract—In future autonomous systems, wireless multi-hop communication is key to enable collaboration among distributed agents at low cost and high flexibility. When many agents need to transmit information over the same wireless network, communication becomes a shared and contested resource. Event-triggered and self-triggered control account for this by transmitting data only when needed, enabling significant energy savings. However, a solution that brings those benefits to multi-hop networks and can reallocate freed up bandwidth to additional agents or data sources is still missing. To fill this gap, we propose *control-guided communication*, a novel co-design approach for distributed self-triggered control over wireless multi-hop networks. The control system informs the communication system of its transmission demands ahead of time, and the communication system allocates resources accordingly. Experiments on a cyber-physical testbed show that multiple cart-poles can be synchronized over wireless, while serving other traffic when resources are available, or saving energy. These experiments are the first to demonstrate and evaluate distributed self-triggered control over low-power multi-hop wireless networks at update rates of tens of milliseconds.

Index Terms—Wireless control systems, self-triggered control.

I. INTRODUCTION

THE unparalleled flexibility and cost efficiency when closing feedback loops over wireless networks enables many cyber-physical applications. For instance, in a smart factory, plants are controlled via remote controllers, mobile robots interact with the plants, and distributed sensors provide additional measurements. Another example is drones regularly exchanging data to fly in formation. These and other applications demand wireless *multi-hop* communication to cover large distances and *fast* update intervals of tens of milliseconds to keep up with the dynamics of the systems to be controlled [1]. **Challenges.** Fast feedback control over wireless multi-hop networks is challenging owing to the inherent imperfections of wireless networks, such as transmission delays and message loss. Moreover, the limited network bandwidth can lead to congestion when many agents need to communicate at the same time, and wireless radios draw considerable power, which

*Equal contribution.

¹Intelligent Control Systems Group, Max Planck Institute for Intelligent Systems, Stuttgart/Tübingen, Germany. Email: dbaumann@tuebingen.mpg.de, trimpe@is.mpg.de.

²Networked Embedded Systems Lab, TU Dresden, Dresden, Germany. Email: {fabian.mager, marco.zimmerling}@tu-dresden.de

This work was supported in part by the German Research Foundation within the Cluster of Excellence cfaed (grant EXC 1056), SPP 1914 (grants ZI 1635/1-1 and TR 1433/1-1), and the Emmy Noether project NextIoT (grant ZI 1635/2-1); the Cyber Valley Initiative; and the Max Planck Society.

Table I: Qualitative comparison of prior and our work on integrating STC with wireless communication, evaluated through real-world experiments.

Work	Fast update intervals	Multi-hop	Energy savings	Reallocation	Distributed implementation
[8]	✗	✗	✓	✓	✗
[9]	✗	✗	✓	✓	✗
[10]	✗	✓	✓	✗	✗
[11]	✓	✗	✓	✗	✗
[12]	✓	✗	✓	✗	✗
This	✓	✓	✓	✓	✓

is a major concern for embedded sensors and mobile devices that must be untethered and thus powered by batteries. For these reasons, adaptive schemes are needed where agents use the network only when necessary to *save energy*, and available resources are *reallocated* at run time to serve those in need.

To use the limited bandwidth and energy more efficiently, event-triggered control (ETC) and self-triggered control (STC) methods have been developed [2], [3]. Unlike periodic control, in ETC and STC the decision whether to communicate or not is based on events, such as an error exceeding a threshold. ETC *instantaneously* decides whether to communicate, leaving no time to save energy or reallocate bandwidth in case of a negative triggering decision. STC, instead, decides *ahead of time* about the next triggering instant. However, to utilize freed resources (*e.g.*, to serve traffic from additional remote sensors), an integration of STC designs and wireless communication protocols is required. Moreover, such co-design approaches must be evaluated on real cyber-physical testbeds to establish trust in feedback control over wireless [4]. While a large body of work on STC exists (see [2], [3], [5]–[7] and the references therein), the integration of STC designs with wireless protocols including an experimental evaluation has rarely been considered. The few exceptions are listed in Table I and discussed next.

Prior work. Existing approaches integrating STC and wireless communication target remote control, for example, of a double-tank process [8], [9], a simulated load-positioning system [10], or a mobile robot [11]. Coordination in multi-robot systems has been studied in [12], but the control commands are computed by a central entity, so the implementation is not distributed. All works show that STC allows for solving the control task with less communication than periodic control, enabling significant energy savings. However, reallocation of freed resources has only been demonstrated in [8], [9], for single-hop networks and update intervals of a few seconds. In fact, STC over a wireless multi-hop network has only been shown in [10], with an update interval of 1 s.

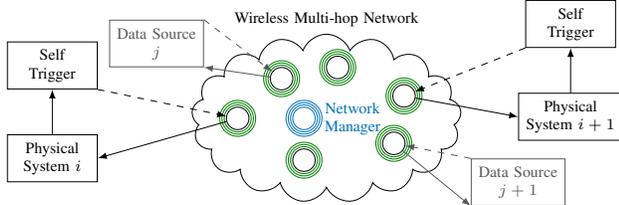


Figure 1: We consider multiple physical systems connected over a wireless multi-hop network. Each system is associated with a self trigger that computes at the current communication instant when it needs to communicate next. This information is piggybacked onto the message it sends. The network manager uses this information to compute a communication schedule respecting these demands and, if possible, reallocating bandwidth to additional data sources.

In summary, no solution exists that provides energy savings *and* reallocation of freed resources for the control of systems at fast update intervals over multi-hop networks. Moreover, no work has shown a distributed implementation of a STC law, where agents locally use information obtained over the network to solve a common control task. However, a complete solution is needed to enable novel applications, such as collaborative multi-robot swarms for future smart production systems.

Contribution. We present a co-design of control and communication for multi-hop wireless networks that fills this gap. Our approach arbitrates the available communication bandwidth among different types of traffic from any entity in the network, while simultaneously shutting down resources completely to save energy when neither the control system nor any other entity needs the full bandwidth. We evaluate the approach on a three-hop cyber-physical testbed with multiple physical systems [13], demonstrating improved resource efficiency at high control performance for update intervals below 100 ms.

At the heart of our solution is the novel concept of *control-guided communication*: The control system informs the communication system *at run time* about its resource requirements, and the communication system leverages this information to dynamically allocate or shut down resources. Concretely, we consider the setup depicted in Fig. 1. Each agent uses STC to decide at the current communication instant when it will communicate next. The agent piggybacks the decision of its self trigger onto the messages it sends. The network manager uses this information as input when dynamically computing the communication schedule at run time. For example, when some agents do not need to communicate, their share of the bandwidth can be reallocated to serve other traffic (*e.g.*, from remote sensors) or can be shut down to conserve energy. The concrete scheduling policy is an exchangeable component of our design and can be adapted to the application requirements.

In essence, we make the following two main contributions:

- We propose control-guided communication, a tight integration of STC and wireless multi-hop communication in which the control system informs the network at run time about future communication demands to enable both energy savings *and* reallocation of network bandwidth.
- Using experiments on a real cyber-physical testbed with five inverted pendulums, we are the first to demonstrate distributed STC over wireless multi-hop networks with update intervals below 100 ms, while showing energy savings of up to 87 % compared to the periodic baseline.

II. PROBLEM SETTING

We consider N physical systems connected over a wireless multi-hop network, as shown in Fig. 1. Each agent is modeled as a stochastic, linear, and time-invariant system

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

with state $x_i(k) \in \mathbb{R}^n$, input $u_i(k) \in \mathbb{R}^m$, and $v_i(k) \in \mathbb{R}^n$ a Gaussian random variable with zero mean and variance Σ_i , capturing process noise. We assume each agent has a local controller that receives local observations directly, but also needs information from other agents for distributed control.

There are various methods to design distributed controllers (see, for example, [14]). In this work, we adopt an approach based on the linear quadratic regulator (LQR) [15]. Using augmented states $\tilde{x}(k) = (x_1(k), \dots, x_N(k))^T$ and inputs $\tilde{u}(k) = (u_1(k), \dots, u_N(k))^T$, we define the cost function

$$J = \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E}[\tilde{x}^T(k) Q \tilde{x}(k) + \tilde{u}^T(k) R \tilde{u}(k)], \quad (2)$$

with positive definite weight matrices Q and R . The optimal stabilizing controller that minimizes (2) is of the form $u_i(k) = \sum_j F_{ij} x_j(k)$, where F_{ij} denotes entry (i, j) of the feedback matrix F . That is, to implement this controller, each agent needs information from all other agents, which is sent over the wireless multi-hop network. To provide high-performance control while efficiently using limited network bandwidth and energy resources, the system must meet several requirements:

- For coordination, the agents need to exchange data; in particular, for optimal control according to (2), all agents need to communicate with one another (all-to-all).
- Wireless multi-hop communication must be reliable and fast to support feedback control of physical systems with fast dynamics; we target mechanical systems requiring update intervals on the order of tens of milliseconds [1].
- The network must arbitrate among multiple types of data traffic as determined by the communication schedule, while always giving highest priority to control traffic.
- If some fraction of the bandwidth is not allocated to any entity, this resource should be shut down to save energy.

III. CO-DESIGN APPROACH

The main goal of this paper is to facilitate high-performance distributed control across multi-hop wireless networks with highly adaptive resource arbitration and allocation to support multiple traffic types and save unused resources. Prior work failed to reach this goal because the many imperfections of wireless systems, such as time-varying end-to-end delays and limited throughput, complicate the control design and make it difficult to quickly coordinate the system-wide operation and resource usage based on the current control-traffic demands.

To tackle this issue, we propose a novel co-design approach that integrates the control and communication systems in two ways. First, the *design* of the communication system tames network imperfections as much as possible, and the control system accounts for the emerging key properties and remaining imperfections. Second, during *operation*, the control system reasons about its future communication demands and informs

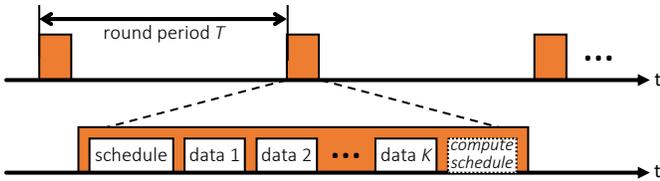


Figure 2: Time-triggered operation of the multi-hop low-power wireless protocol. Communication occurs in rounds with a constant period T . Each round consists of a schedule slot and up to K data slots. The schedule slot serves to inform all nodes of the number of subsequent data slots in the round and the allocation of control or other messages to the scheduled data slots.

the communication system accordingly. The communication system, on the other hand, adapts to these demands by arbitrating the available bandwidth among different types of traffic and by shutting down resources completely to save energy when neither the control system nor any other participant needs the full bandwidth. We call this concept *control-guided communication*, which we detail in the following two sections.

In addition, our wireless communication system provides fast and reliable many-to-all communication among any set of agents, even when the agents are mobile and thereby causing the network topology to change continuously. This feature is a key difference to traditional wireless communication systems, such as WirelessHART, and makes our co-design approach directly applicable to solve various kinds of distributed control problems that may be stated in the form of a cost function (2).

IV. WIRELESS COMMUNICATION SYSTEM DESIGN

We first describe the design of the wireless communication system, and detail the control design based on the emerging properties in the next section. The wireless system builds on the periodic design in [16] and consists of three elements, where 2) is significantly modified and 3) is a new component:

- 1) a *hardware platform* enabling a predictable and efficient execution of all control tasks and message transfers;
- 2) a *multi-hop wireless protocol* that provides many-to-all communication with minimal, bounded end-to-end delay;
- 3) an *online scheduler* that dynamically assigns bandwidth to each agent based on its communication requirements.

Hardware platform. We use a dual-processor platform (DPP) where sensing, actuation, and control execute on an application processor (MSP432P401R, 32 bit, 48 MHz) and the wireless multi-hop protocol executes on a communication processor (CC430F5147, 16 bit, 13 MHz). The processors communicate through the Bolt interconnect [17], which provides bounded worst-case execution times for the bidirectional exchange of messages between both processors. In this way, control and communication can efficiently execute in parallel and never interfere with each other, providing timing predictability.

Multi-hop wireless protocol. The communication processor of every DPP in the network runs a multi-hop protocol, whose design is inspired by a new breed of protocols that exploit *synchronous transmission* based flooding for highly reliable and efficient communication. As shown in Fig. 2, using our protocol, communication occurs in *rounds* of equal duration that repeat with a constant period T . Each round consists of a sequence of non-overlapping *slots*. In each slot, one node is allowed to initiate a Glossy flood [18] to send a message

to all other nodes. Glossy achieves the theoretical minimum latency for flooding a message in a multi-hop network using half-duplex radios, and provides a reliability above 99.9% in real-world scenarios [18], [19]. In fact, Glossy’s reliability can be pushed beyond 99.9999% by letting nodes transmit more often during a flood, and it time-synchronizes all nodes to within sub-microsecond accuracy at no additional cost [18].

Any node in the network can serve as the designated *network manager* that uses the first slot in a round to flood the *schedule*. The schedule informs all other nodes about the number of data slots in the round (up to K) and the allocation of nodes to these data slots. The transmitted messages carry, for example, high-priority control information from agents or lower-priority data from other nodes, such as measurements from a remote sensor or information about a node’s health status (*e.g.*, its battery’s state of charge). When sending a message, a node also piggybacks information about its future communication demands; if the network manager does not receive a message, it assumes that the respective node needs to transmit in the next round. Based on all demands, the network manager computes the schedule for the next round after the last data slot.

Online scheduler. To this end, the network manager maintains a list of unserved communication demands, and allocates up to K nodes to the data slots in the next round according to a *scheduling policy*. The scheduling policy can be adjusted to meet different application requirements. As an illustrative example, we design in this paper a new policy that aims to strike a balance between resource efficiency and accommodating lower-priority messages next to control traffic. Specifically, if there are free data slots after assigning all nodes with pending control messages in the next round, we allocate one of the free data slots to a node for sending some other message (sensor, status, etc.). The next node to send such message is chosen in a round-robin fashion. Any other free slot is left empty. Since nodes have their radios only on during allocated slots and off otherwise, this example policy illustrates that our wireless communication system allows for both arbitrating bandwidth among different traffic types and not allocating resources at all to save energy, as demonstrated in Sec. VI.

Key properties. Our wireless system design provides highly reliable, efficient many-to-all communication, system-wide time synchronization, and adapts at run time to the nodes’ communication demands. Due to the time synchronization, we can schedule control and communication tasks such that the jitter on the update interval and end-to-end delay is less than $\pm 50 \mu\text{s}$, as formally and experimentally validated in [16].

V. SELF-TRIGGERED CONTROL DESIGN

We now detail the control design, first our approach to distributed control and then our self-triggered design.

A. Distributed Control

The wireless communication system provides a constant update interval T as the jitter is negligible for the considered scenarios. We thus set one discrete time step in (1) to T and data that is sent over the network is delayed by one time step.

Moreover, the many-to-all communication scheme ensures that information can be received by all agents in the network. This greatly facilitates control design as essentially arbitrary information patterns can be implemented. For example, this allows for implementing a (centralized) optimal controller in a distributed fashion as we show in this paper. Given the high reliability of the wireless embedded system, we assume that data that are sent over the network are received by all agents.

As an example for distributed control, we consider synchronization of multiple agents through an LQR design as in (2). For ease of presentation, we outline the approach for the two-agent case, but it also extends to multiple agents as shown in Sec. VI. We choose the quadratic cost function

$$J = \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=0}^{K-1} \sum_{i=1}^2 \left(x_i^T(k) Q_i x_i(k) + u_i^T(k) R_i u_i(k) \right) + (x_1(k) - x_2(k))^T Q_{\text{sync}} (x_1(k) - x_2(k)) \right], \quad (3)$$

that is, we penalize deviations between $x_1(k)$ and $x_2(k)$ through the positive definite weight matrix Q_{sync} , as well as deviations from the equilibrium ($Q_i > 0$) and high control inputs ($R_i > 0$). Using augmented states as in (2), the term in the summation over k becomes

$$\tilde{x}^T(k) \begin{pmatrix} Q_1 + Q_{\text{sync}} & -Q_{\text{sync}} \\ -Q_{\text{sync}} & Q_2 + Q_{\text{sync}} \end{pmatrix} \tilde{x}(k) + \tilde{u}^T(k) \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix} \tilde{u}(k).$$

As discussed in Sec. II, solving the optimal control problem then leads to a feedback controller that has the form $u_1(k) = F_{11}x_1(k) + F_{12}x_2(k)$, that is, agent 1 needs information from agent 2. We account for this by letting agent 2 send $u_{12}(k) = F_{12}x_2(k)$ over the network. Thus, agent 1's control input consists of $u_{11}(k) = F_{11}x_1(k)$, which it can compute using its local observations, and $u_{12}(k)$, which it receives over the network. We can thus define the closed-loop matrix $\tilde{A}_1 = A_1 + B_1 F_{11}$ and (1) then reads as follows

$$x_1(k+1) = \tilde{A}_1 x_1(k) + B_1 u_{12}(k) + v_1(k). \quad (4)$$

B. Self-triggered Approach

Different STC designs have been proposed and are conceivable to realize control-guided communication. We use a design that exploits ideas from previous work on state estimation [20]. Instead of sending states as in [20], we consider the communication of control inputs. Specifically, rather than sending its entire state, agent 2 only sends the input $u_{12}(k)$ that is needed by agent 1. In case of no communication, agent 1 keeps applying $u_{12}(k_\ell)$, where k_ℓ is the last time step at which the input $u_{12}(k)$ was sent. We trigger communication based on the error $e_{12}(k) := u_{12}(k) - u_{12}(k_\ell)$ as follows

$$\gamma_2(k) = 1 \iff (e_{12}(k))^T e_{12}(k) > \delta. \quad (5)$$

Here, $\gamma_2(k)$ is a binary variable, denoting whether agent 2 communicates $u_{12}(k)$ ($\gamma_2(k)=1$) or not ($\gamma_2(k)=0$), while δ defines the designer's trade-off between saving communication (large δ) and keeping the error to a minimum (small δ).

If we directly implement (5), agent 2 instantaneously decides on whether to transmit $u_{12}(k)$ to agent 1. In case

of a negative triggering decision, there is no possibility to reallocate bandwidth and hence freed resources remain unused. To overcome this problem, we use a self-triggered strategy. Whenever an agent communicates, it already decides when to communicate next. To this end, we predict the evolution of the error and look for the smallest $M > 1$ such that

$$\mathbb{E}[(e_{12}(k+M))^T e_{12}(k+M) | \mathcal{D}_2(k)] > \delta \quad (6)$$

and set $\gamma(k+M-1) = 1$. Here, $\mathcal{D}_2(k)$ describes the data agent 2 collected until time step k , that is, its local states x_2 and the inputs u_2 and u_{12} that it has applied and sent so far, respectively. The rationale behind this triggering rule is as follows: Information that is sent over the network is delayed by one discrete time step. The inequality in (6) tells us that the error exceeds, in expectation, the threshold δ in M time steps. We thus seek to communicate next in $M-1$ time steps such that the new input arrives in M time steps, which is exactly when we expect the error to exceed the threshold.

The exact computation of (6) is complicated by the fact that the input $u_{21}(k)$ is not available at all times at agent 2. To derive the triggering law, we assume $u_{21}(k)$ is known and then comment on how we approximate it to yield a tractable implementation. Based on this, we get the error distribution

$$f(e_{12}(k+M) | \mathcal{D}_2(k)) = \mathcal{N}(\hat{e}_{12}(k+M|k), P_2(k+M|k)),$$

with mean \hat{e}_{12} and variance P_2 given as

$$\hat{e}_{12}(k+M|k) = F_{12}(\tilde{A}_2^M x_2(k) + \sum_{i=0}^{M-1} \tilde{A}_2^{M-i} B_2 u_{21}(k+i)) - u_{12}(k) \quad (7a)$$

$$P_2(k+1|k) = F_{12}^T (\tilde{A}_2^T P_2(k|k) \tilde{A}_2 + \Sigma_2) F_{12}. \quad (7b)$$

Equations (7) are standard open-loop state and covariance predictions of the system in (4), so the derivations follow from Kalman filter theory [21, p. 111].

Given this error distribution, we can now, using $\mathbb{E}[e^T e] = \|\mathbb{E}[e]\|^2 + \text{Tr}(\text{Var}[e])$, solve for the triggering rule (5): At every communication instant, find the smallest $M > 1$ such that

$$\|F_{12}(\tilde{A}_2^M x_2(k) + \sum_{i=0}^{M-1} \tilde{A}_2^{M-i} B_2 u_{21}(k+i)) - u_{12}(k)\|^2 + \text{Tr}(F_{12}^T (\tilde{A}_2^T P_2(k+M|k) \tilde{A}_2 + \Sigma_2) F_{12}) > \delta, \quad (8)$$

with Tr the trace of a matrix.

So far, we assumed that agent 2 has knowledge about the future development of $u_{21}(k+i)$, which does not hold in practice. Because agent 2 has no information about the current state of agent 1 and hence cannot infer the future development of $u_{21}(k+i)$, it approximates $u_{21}(k+i)$ as $u_{21}(k+i) = u_{21}(k) \forall i \in [0, M)$. With this, the input $u_{21}(k+i)$ in (7a) and (8) effectively becomes a constant.

We note that one way to let agent 1 reason about agent 2's state would be to send the entire state $x_2(k)$ instead of the control input $u_{12}(k)$. Agent 1 could use this state to compute $u_{12}(k)$ and to predict the evolution of agent 2's state. This, however, incurs higher communication demands at each instant as the state is typically of higher dimension than the input.

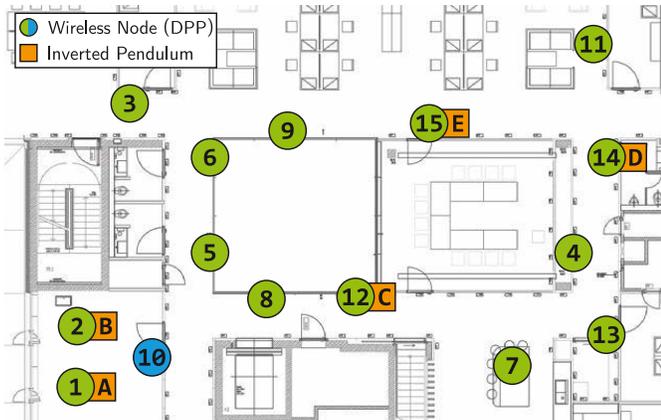


Figure 3: Cyber-physical testbed with 15 wireless DPP nodes and five cart-pole systems (A and B are real systems; C, D, and E are simulated systems). The network has a diameter of three hops. Node 10 is the network manager.

VI. EXPERIMENTAL EVALUATION

We evaluate our approach using experiments on a real cyber-physical testbed [13], [16] shown in Fig. 3. It consists of 15 wireless DPP nodes and five cart-pole systems (or pendulums), where A and B are real systems and C, D, and E are simulated systems. The nodes are distributed in an office space of about 15 m by 20 m, and transmit at -6 dBm and 250 kbit/s in the 868 MHz band, forming a three-hop wireless network.

A. Scenario and Metrics

Scenario. The control task of each pendulum is to locally stabilize itself and to synchronize its cart position with all others. Since each system has access to its local state $x_i(k)$, we can run the local feedback loop at a faster update interval than communication over the network occurs. Here, we choose an update interval of 10 ms for the local loop. Control inputs $u_{ij}(k)$ of the other agents are communicated over the wireless multi-hop network, where the exchange of all control inputs takes 50 ms (*i.e.*, one communication round with up to $K = 5$ data slots and 4 byte per agent). We use the scheduling policy outlined in Sec. IV. To challenge the synchronization of the cart positions, we apply a sine distortion signal (3.6 s period with an amplitude of ± 5 V) to the control input of pendulum B.

The controllers are designed as described in Sec. V. We use the same model for the cart-pole system as in [16] and also adopt the Q_i matrices used for periodic synchronization. For Q_{sync} , we set the first diagonal entry to 20 and all other entries to zero to express our desire to synchronize the cart positions. Further, we choose $R_i = 0.01$ for all systems.

Metrics. Our evaluation uses the following metrics:

- *root mean square of the synchronization error (RMSE)* computed based on the cart positions of all pendulums in an experiment as a measure of control performance;
- *utilization* of the available data slots during each round, broken down into free slots (radio off), slots used for control traffic, and slots used for additional (other) traffic;
- *radio duty cycle*, the fraction of time a node has its radio on, which is a widely used metric in the low-power wireless networking literature (see, *e.g.*, [19], [22]) for quantifying communication energy cost.

In the following, we first illustrate the run time operation of our co-designed wireless control system in a real experiment, and then evaluate the trade-off among control performance, communication energy cost, and serving additional traffic as a function of the triggering threshold.

B. Efficient Resource Arbitration and Allocation

Fig. 4 shows a real trace of the control performance (top) and the slot utilization in each communication round (bottom) over time for a triggering threshold of $\delta = 0.03$. Looking at the utilization, we see that, on average, less than one third of the available bandwidth is needed for control traffic. Our co-design approach effectively uses the freed bandwidth to schedule additional traffic (here at most one slot per round according to the example scheduling policy from Sec. IV) and to shut down the remaining bandwidth completely. During the many free slots all nodes have their radios turned off, which saves significant amounts of energy. Due to the sine distortion signal, the RMSE at the top exhibits a similar shape.

C. Control Performance vs. Efficiency vs. Flexibility

The triggering threshold δ allows a user to trade control performance for communication energy efficiency and flexibility in serving other traffic. To evaluate this trade-off, we consider six different thresholds and perform for each threshold three 2-minute experiments. In addition, we perform experiments with $\delta = 0$ to obtain results for periodic control, where all agents communicate in every time step requiring all bandwidth for control traffic. For each threshold, we report the median and 25th/75th percentiles across the three experiments.

Fig. 5 shows RMSE, radio duty cycle for control traffic, and fraction of bandwidth available for other traffic against the fraction of bandwidth used for control traffic. We use this intuitive unit for the x-axis instead of the triggering threshold δ because our measurements reveal that each δ corresponds to a certain fraction of bandwidth used for control traffic with negligible variance across experiments with the same δ .

Looking at Fig. 5, we observe that the more bandwidth is used for control traffic, the better the control performance and the less bandwidth is available for other traffic. As expected, higher bandwidth demands result in a higher radio duty cycle. Using 25 % of the available bandwidth for control traffic, the control performance is still comparable to the periodic baseline. Further bandwidth reductions lead to a noticeable decrease in control performance compared with the periodic baseline of up to 22 % when only 11 % of the available bandwidth is used for control traffic. At the same time, up to 87 % of communication energy can be saved, while the vast majority of the bandwidth is available for other traffic. Overall, these experimental results demonstrate that our control-guided communication approach allows for exploiting this trade-off to meet a wide range of requirements of emerging cyber-physical applications.

VII. CONCLUSIONS

We have demonstrated for the first time distributed, self-triggered control over wireless multi-hop networks with energy savings *and* reallocation of resources at fast update intervals

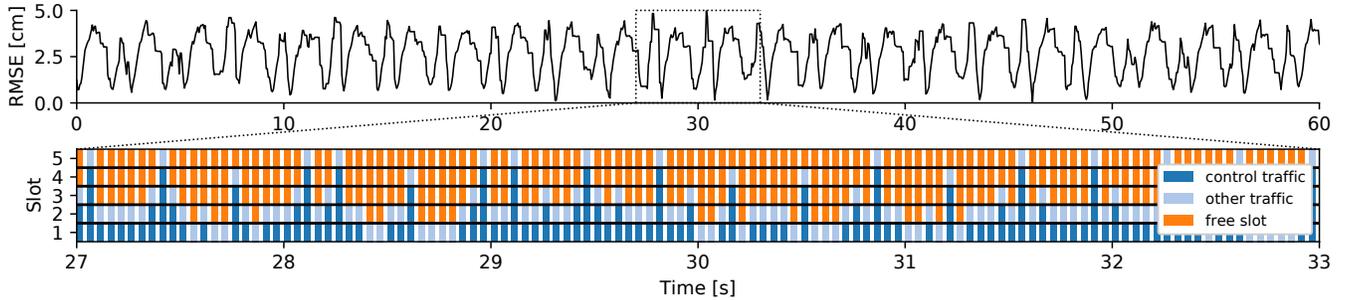


Figure 4: Control performance and bandwidth utilization over time, recorded during one of our experiments. The scheduling policy described in Sec. IV is used but applications can also specify any other policy. Each vertical line in the lower figure represents a communication round. The control traffic demands vary over time between 0 and 4 slots. One slot is always used for other traffic and the remaining free slots are shut down to save communication energy.

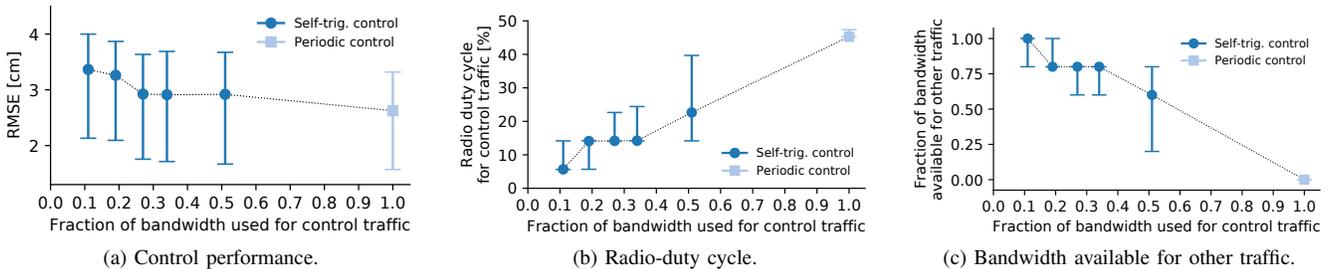


Figure 5: Trade-off between control performance, communication energy efficiency, and flexibility in serving other traffic for different fractions of control traffic, reported in terms of the median and 25th/75th percentiles. Control performance decreases when less bandwidth is used for control traffic. Conversely, freed resources that are not needed for control traffic result in considerable communication energy savings or allow to serve other traffic (e.g., status, sensors).

of tens of milliseconds. At the heart of our solution is control-guided communication, a new co-design approach where the control system predicts and informs the communication system about future resource demands. Using this information, bandwidth and energy are either saved or used efficiently for different kinds of traffic. Experiments on a real cyber-physical testbed show the effectiveness of our approach. As part of our future work, we focus on a variety of theoretical questions, for example, regarding the closed-loop stability of the overall system, especially in the presence of message loss.

ACKNOWLEDGEMENTS

We thank Harsoveet Singh and Felix Grimminger for 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] J. Åkerberg, M. Gidlund, and M. Björkman, “Future research challenges in wireless sensor and actuator networks targeting industrial automation,” in *9th IEEE Int. Conf. on Industrial Informatics*, 2011.
- [2] W. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *Proc. of IEEE CDC*, 2012.
- [3] M. Miskowicz, *Event-Based Control and Signal Processing*. CRC Press, 2016.
- [4] 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,” *Proc. IEEE*, vol. 104, no. 5, 2016.
- [5] M. Velasco, J. Fuertes, and P. Marti, “The self triggered task model for real-time control systems,” in *Proc. of IEEE RTSS*, 2003.
- [6] X. Wang and M. Lemmon, “Self-triggered feedback control systems with finite-gain \mathcal{L}_2 stability,” *IEEE Trans. Automat. Contr.*, vol. 54, no. 3, 2009.
- [7] M. Mazo, A. Anta, and P. Tabuada, “An ISS self-triggered implementation of linear controllers,” *Automatica*, vol. 46, no. 8, 2010.
- [8] J. Araujo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, “System architectures, protocols and algorithms for aperiodic wireless control systems,” *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, 2014.
- [9] 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.
- [10] Y. Ma and C. Lu, “Efficient holistic control over industrial wireless sensor-actuator networks,” in *Proc. of IEEE ICII*, 2018.
- [11] 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.
- [12] 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.
- [13] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe, “Evaluating low-power wireless cyber-physical systems,” in *Proc. of IEEE CPSBench*, 2018.
- [14] J. Lunze, *Feedback Control of Large-Scale Systems*. Prentice Hall, 1992.
- [15] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Prentice Hall, 2007.
- [16] 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 *Proc. of ACM/IEEE ICCPS*, 2019.
- [17] 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 *Proc. of ACM SenSys*, 2015.
- [18] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, “Efficient network flooding and time synchronization with Glossy,” in *Proc. of ACM/IEEE IPSN*, 2011.
- [19] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, “Low-power wireless bus,” in *Proc. of ACM SenSys*, 2012.
- [20] S. Trimpe and D. Baumann, “Resource-aware IoT control: Saving communication through predictive triggering,” *IEEE Internet Things J.*, 2019.
- [21] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.
- [22] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *Proc. of ACM SenSys*, 2009.