MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Optimal Dynamic Transmission Scheduling for Wireless Networked Control Systems

Ma, Yehan; Guo, Jianlin; Wang, Yebin; Chakrabarty, Ankush; Ahn, Heejin; Orlik, Philip V.; Guan, Xinping; Lu, Chenyang

TR2022-043 May 19, 2022

Abstract

Wireless networked control systems (WNCS) have the potential to revolutionize industrial automation in smart factories. Optimizing closed-loop performance while maintaining stability is a fundamental challenge in WNCS due to limited bandwidth and non-deterministic link quality of wireless networks. In order to bridge the gap between network design and control system performance, we propose an optimal dynamic transmission scheduling strategy that optimizes performance of multi-loop control systems by allocating network resources based on predictions of both link quality and control performance at run-time. We formulate the optimal dynamic scheduling problem as a nonlinear integer programming problem, which is relaxed to a linear programming problem. We further extend the optimization problem to balance control performance and communication cost. The proposed optimal dynamic scheduling strategy renders the closed-loop system mean-square stable under mild assumptions. Its efficacy is demonstrated by simulating a four-loop control system over an IEEE 802.15.4 wireless network simulator – TOSSIM. The run-time network reconfiguration protocol tailored for optimal scheduling is designed and implemented on a real wireless network consisting of IEEE 802.15.4 devices. Hybrid simulations integrating a real wireless network and simulated physical plant control are performed. Simulation and experimental results show that the optimal dynamic scheduling can enhance control system performance and adapt to both constant and variable wireless interference and physical disturbance to the plant.

IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Optimal Dynamic Transmission Scheduling for Wireless Networked Control Systems

Yehan Ma, *Member, IEEE*, Jianlin Guo, *Senior Member, IEEE*, Yebin Wang, *Senior Member, IEEE*, Ankush Chakrabarty, *Senior Member, IEEE*, Heejin Ahn, *Member, IEEE*, Philip Orlik, *Senior Member, IEEE*, Xinping Guan, *Fellow, IEEE*, Chenyang Lu, *Fellow, IEEE*

***Abstract*—Wireless networked control systems (WNCS) have the potential to revolutionize industrial automation in smart factories. Optimizing closed-loop performance while maintaining stability is a fundamental challenge in WNCS due to limited bandwidth and non-deterministic link quality of wireless networks. In order to bridge the gap between network design and control system performance, we propose an optimal dynamic transmission scheduling strategy that optimizes performance of multi-loop control systems by allocating network resources based on predictions of both link quality and control performance at run-time. We formulate the optimal dynamic scheduling problem as a nonlinear integer programming problem, which is relaxed to a linear programming problem. We further extend the optimization problem to balance control performance and communication cost. The proposed optimal dynamic scheduling strategy renders the closed-loop system mean-square stable under mild assumptions. Its efficacy is demonstrated by simulating a four-loop control system over an IEEE 802.15.4 wireless network simulator – TOSSIM. The run-time network reconfiguration protocol tailored for optimal scheduling is designed and implemented on a real wireless network consisting of IEEE 802.15.4 devices. Hybrid simulations integrating a real wireless network and simulated physical plant control are performed. Simulation and experimental results show that the optimal dynamic scheduling can enhance control system performance and adapt to both constant and variable wireless interference and physical disturbance to the plant.**

***Index Terms*—Industrial internet of things, cyber-physical system, wireless networked control system, dynamic scheduling.**

I. Introduction

WIRELESS sensor-actuator networks (WSANs) are gaining rapid adoption in industrial automation for lowering deployment and maintenance costs in harsh industrial environments. Industrial standard organizations, such as ISA100 [2], WirelessHART [3], and ZigBee [4], have been actively pushing a wide range of applications of IEEE 802.15.4 [5] based wireless technologies in industrial automation, which emphasize very low-cost communication [6], [7]. However, in wireless networked control systems (WNCS), while advances have been made in sensing and monitoring, there are multiple challenges in closing the loop over wireless networks for control and actuation.

First, the networks with cable, power cord, and base station, such as Ethernet/Wi-Fi/5G, offering high data rates of over tens of Mbps. However, in industrial settings, a wireless device that requires a power cord is often impractical due to harsh industrial environments, flexibility requirements, and deployment cost according to ABB and Emerson. Therefore, one of the keys to the design of industrial field devices and wireless standards is to guarantee they could be battery powered for 4 to 10 years [8], [9]. Low-rate wireless personal area networks (LR-WPAN) have limited throughput (e.g., IEEE 802.15.4 physical layer supports data rates of only up to 250 kbit/s) but very low energy consumption and cost. As a result, LR-WPAN can be battery powered and completely wireless under recourse constraints. The control performance of WNCS is strongly correlated with the amount of network resource they can obtain. Besides, in the unlicensed 2.4 GHz band, multiple systems including Wi-Fi and WNCS are competing for the limited spectrum resource. Therefore, it is crucial to make use of the limited network resources while improving control performance in a WNCS.

Second, the physical isolation of wired networks ensures supreme link quality and resilience to external environment changes. In contrast, link qualities of wireless networks are prone to environmental factors such as obstacles, noise, interference, extreme weather, as well as human interference in the form of cyber attacks. Poor link quality can cause significant data packet loss, resulting in severe degradation of the control performance. Finally, most wireless network designs focus on network performance, overlooking control performance, which directly determine the profits and the safety of a factory. Therefore, a practical wireless network design for industrial control must optimize control performance while taking limited network resources and the impact of link quality and plant states into consideration.

In this paper, we bridge the gaps between control performance and network design by exploring the direct impact of network link quality and network resources allocation on physical control systems. We design an optimal dynamic

Yehan Ma is with John Hopcroft Center, Shanghai Jiao Tong University, Shanghai, 200240 China. She was a Ph.D. student at Washington University in St. Louis and an intern at Mitsubishi Electric Research Laboratories. E-mail: yehanma@sjtu.edu.cn.

Jianlin Guo, Yebin Wang, Ankush Chakrabarty, and Philip Orlik are with Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139 USA. E-mail: {guo, yebinwang, chakrabarty, porlik}@merl.com.

Heejin Ahn was a visiting research scientist at Mitsubishi Electric Research Laboratories and is now a postdoctoral research fellow at the University of British Columbia, Vancouver, BC V6T 1Z4, Canada. Email: hjahn@cs.ubc.ca.

Xinping Guan is with the Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240 China. E-mail: xpguan@sjtu.edu.cn.

Chengyang Lu is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO, 63130 USA. E-mail: lu@wustl.edu.

This submission is an extended version of the conference paper published at ACM/IEEE ICCPS, 2019. [1].

transmission scheduling strategy to optimize the control performance based on run-time predictions of link quality and physical control performance.

Our major **contributions** in this paper include:

1) incorporating link quality prediction of wireless networks and control performance prediction of physical processes into transmission scheduling;
2) providing a computationally tractable method for optimal network scheduling based on predictions of both link quality and the control performance;
3) establishing stability guarantees for the closed-loop control system with optimal scheduling;
4) illustrating the efficacy of our strategy on the high-fidelity TOSSIM simulation environment in spite of constant and variable channel noise in the wireless networks and disturbance to the physical plants;
5) designing and implementing the run-time network reconfiguration protocol tailored for optimal scheduling on IEEE 802.15.4 devices; performing hybrid simulations integrating the real wireless network and simulated physical plants; evaluating the dynamic scheduling algorithm over a real IEEE 802.15.4 wireless network.

The rest of the paper is organized as follows. We present related work in Section II, and provide an overview of wireless networked control systems in Section III. Section IV presents the optimal scheduling problem and its linear programming relaxation, and a stability analysis is provided in Section V. Section VI evaluates the simulation and experimental results, and we present our conclusions in Section VII.

II. Related work

The past decade has witnessed sustained interest in exploring WNCS and expanding their applications over industrial automation [6], [7], in the views of network, control system design, and recently, network and control co-design.

From a network design perspective, several approaches are presented to address resources allocation. For example, Huang et al. [10] propose an adaptive time slot allocation scheme for IEEE 802.15.4, which considers low latency and fairness of packet waiting time; Zhan et al. [11] allocate network resources by adjusting the slot length adaptively in accordance with the data size of the end device. Given link quality, end-to-end packet delivery ratio (PDR) can be effectively improved by retransmission [12], channel selection [13], routing [14], and reachability-aware scheduling [15], etc. However, few are targeting optimizing control performance.

On the control side, many control designs based on the physical plant models as well as on network parameters are performed to maintain the performance. To name a few, Sinopoli et al. [16] discuss Kalman filtering with intermittent measurement; Gao et al. [17] investigate robust output tracking control subject to time delay between controllers and actuators; Ma et al. [18] explore the design freedom of system architectures and propose a smart actuation architecture; Wang et al. [19], [20], [21] model packet loss as a Bernoulli or Markov-type process and establish stochastic stability of the resultant WNCS. However, most control designs consider only application-level network parameters, such as latency and PDR, instead of lower-level parameters, such as link quality and signal-to-noise ratio (SNR). However, with only application level information, it is hard to fully utilize and manage network resources for control performance.

More recently, network and control co-designs aim to jointly design the control and network to eliminate the effects of limited throughput and poor link quality of wireless networks, among which there are network resources allocation designs tailored for control performance of WNCS. Saifullah et al. determine [22], [23] sampling rates to optimize control performance. Gatsis et al. [24] propose distributed control-aware random network access policies for each sensor so that all control loops are stabilizable. Antunes et al. [25], [26] decide which node accesses the network and the control policy at each network time instant so as to optimize a quadratic performance objective. Lješnjanin et al. [27] allocate network resources by finding optimal nodes, which minimize cost function of model predictive control (MPC) in terms of network packet loss. Ma et al. [28], [29] propose the concept of holistic control that cojoins network reconfiguration and physical control over multi-hop mesh network. However, [22], [24], [25], [26] assume perfect link quality, and none of [23], [27], [28], [29] models the effects of link quality on control performance.

Furthermore, network protocols play a key role in the problem formulation and approach. Our work adopts IEEE 802.15.4 network as a protocol suitable for industrial control, which distinguishes our work from others based on pure mathematical modeling [25], [26], Wi-Fi [30], and cellular networks [31]. We further contribute to network reconfiguration protocols. Peters et al. [32] co-design scheduler and controller to derive optimal control and schedule actuation commands by incorporating both contention-free and contention-based medium access of IEEE 802.15.4 network. However, they assume that PDR is constant and do not consider retransmission in scheduling, which is a key factor of improving PDR and control performance [12].

In this paper, we explore the direct impact of wireless network link quality and resources allocation on the physical control system performance, and formulate an optimal dynamic scheduling strategy to optimize the control performance by dynamically allocating the number of transmissions among multiple control loops.

It is challenging to conduct experiments on industrial control systems, especially under cyber and physical disturbance. Therefore, simulation tools are of vital importance to evaluating WNCS. Truetime [33] is a MATLAB/Simulink-based tool including simulations of CPU scheduling and communication in addition to control algorithms and physical plants. NCSWT [34] integrates MATLAB/Simulink and NS-2 for simulations of WNCS. However, the wireless models adopted in these simulators do not accurately model the probabilistic and irregular packet receptions of wireless sensor networks [35], [36]. WCPS [37] integrates MATLAB/Simulink and TOSSIM [38] specifically designed to emulate complex temporal link dynamics of wireless networks. More recently, real wireless sensor networks have been used in WNCS experiments with lab-scale control testbeds[39], [40], [41]. To com-

bine the versatility of simulations of physical control systems and real wireless sensor networks, we develop a network-in-the-loop simulator, which integrates MATLAB/Simulink and a wireless network consisting of real IEEE 802.15.4 devices, which allows us to empirically evaluate the transmission scheduling mechanisms and their impacts on control systems.

III. Overview of WNCS

We provide an overview of the WNCS architecture used in this work (as shown in Fig. 1). The controllers are typically located separately from the physical plants. One reason is that plants operate in environments which may not be conductive to hardware implementation of control algorithms. Another reason is one control algorithm may be responsible for multiple plants, and therefore, a larger centralized unit of computation may be required to implement such an algorithm. The physical plants and the controllers communicate with each other via wireless networks. We consider N control loops that share the same wireless network with limited network resources, and we aim at optimizing the overall control performance of all control loops under both wireless interference and physical disturbance.

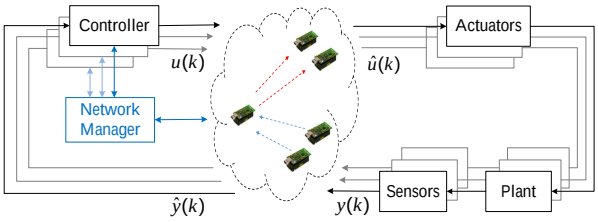


Fig. 1: Architecture of WNCS (red and blue dashed arrows indicate actuation and sensing flows, respectively).

A. Physical plant and controller

We assume that each control loop is associated with an individual plant. For the i-th loop, $i \in \{1, 2, \ldots, N\}$, the corresponding plant is modeled as a nonlinear discrete-time system of the form:

$$x_i(k+1) = f_i(x_i(k), u_i(k)), \tag{1}$$

where k is the sampling period index, $x_i(k) \in \mathbb{R}^{n_i}$ is the state vector, and $u_i(k) \in \mathbb{R}^{m_i}$ is the actuation vector that renders the closed-loop system asymptotically stable when there is no packet loss in network. For simplicity, we state all definitions and theorems for the case when the equilibrium point is at the origin of $\mathbb{R}^{n_i}$. There is no loss of generality because any equilibrium point can be shifted to the origin via a transformation of variables [42].

At time index k, a sensor sends measurements $y_i(k)$ to a controller over the wireless network. At the controller side, a state observer [16] estimates the states of the plant. Based on the estimated state $\hat{x}_i(k)$, the controller generates the actuation command $u_i(k)$ and sends it to the actuator over the wireless network. The actuator then applies $\hat{u}_i(k)$ to the plant. If $u_i(k)$ fails to be delivered by the deadline, the actuator reuses the control input of last period, $\hat{u}_i(k-1)$.

B. Wireless network

1) Wireless Sensor-Actuator Network (WSAN): We assume a WSAN comprising a coordinator and devices each equipped with IEEE 802.15.4 radio. The WSAN has a star topology in which every device can directly communicate with the coordinator. A transmission schedule is created and updated for the sensing and actuation flows of the control loops. In IEEE 802.15.4 MAC, a transmission schedule is defined by a *superframe* comprising a collection of timeslots that are repeated over time. In the beacon-enabled mode, a superframe starts with beacons sent by the coordinator. As shown in Fig. 2, the beacon frame transmission starts at the beginning of the first slot of each superframe. The beacons are used to synchronize the devices, to identify the network (including collecting link quality), and to describe the transmission schedule of the superframes. During the inactive period, the coordinator and devices may enter a low-power (sleep) mode. The active period is composed of contention-access period (CAP) and contention-free period (CFP). During CAP, devices compete for media access using carrier sense multiple access/ collision avoidance (CSMA/CA). Flows with real-time performance requirements are usually assigned guaranteed time slots (GTSs) during CFP. As specified by IEEE 802.15.4 MAC protocol [5], CFP may include up to 7 GTSs. The limitations of IEEE 802.15.4 MAC protocol was discussed and modified by [40], [43], such that the number of slots assigned to CAP and CFP becomes a tunable parameter. As we target control systems with real-time performance requirements, we focus on the scheduling of CFP, whereas CAP can be used for other flows.

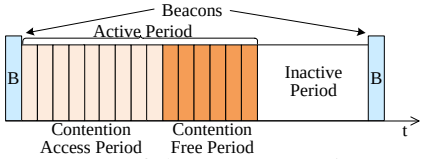


Fig. 2: Structure of the IEEE 802.15.4 superframe

2) Network reconfiguration protocol: The network manager (NM) manages the network and its devices. The NM, the plant controller, and the coordinator of the network are usually co-located or connected via a reliable wired network with negligible packet drop and latency [44]. We propose an NM that dynamically updates the schedule based on the information of predicted link quality and the knowledge of predicted control performance from the controllers to obtain optimal scheduling. As previous work found that control performance can be particularly sensitive to data loss of actuation flows [45], we focus on dynamically scheduling the *actuation* flows, while sensing flows follow static schedules.

After receiving the updated schedule from the NM, and the coordinator broadcasts the schedule in the beacon at the beginning of the next superframe. In this way, devices that receive the beacon update their schedules accordingly. In case a device misses the beacon, it stays awake and keeps listening to receive its actuation command in this superframe. This strategy improves resilience at additional energy cost when the beacon is lost.

Remark III.1. *In a multi-loop WNCS, the NM dynamically updates the schedule based on the predicted link quality and*

control performance of each loop. When the scheduled number of transmissions of loop i, denoted by η_i, is zero, the actuation event of loop i is not triggered. By determining η_i as 0 or $\mathbb{Z}^+$, actuation events of control loops are either skipped or triggered by the NM. Thus, a time-triggered controller, modulated by the proposed network resources allocation, can be viewed as a special kind of event-triggered control. The trigger condition depends on the combination of network and plant control states. □

IV. Optimal Scheduling

In this section, we propose an optimal dynamic scheduling strategy that optimizes control performance by allocating limited network resources based on predictions of both link quality and control performance at run-time. Following the architecture illustrated in Fig. 3, first, the controllers operate the control algorithm and generate control commands (red ①). And the controllers predict closed and open-loop plant performance for the next period (red ②), separately. At the same time, the NM evaluates the current status of the network (blue ①) and predicts the link quality of the next period (blue ②). Then the NM (optimal scheduler) solves the optimal scheduling problem given predicted plant performances and link qualities (blue ③). After solving the problem, the NM informs the network of the resultant optimal scheduling in the beacon message and sends the corresponding slots to the controller (blue ④). Finally, all nodes in the network adjust their schedules (blue ⑤), and the control inputs are sent in the assigned slots in the next period (red ⑤).

We will present (1) how the link quality is predicted by NM, (2) how the plant performance is predicted by the controller, and (3) how the optimal scheduler generates a schedule based on predicted link quality and plant performance. We formulate the optimal scheduling problem as a nonlinear integer program, which is relaxed into a linear programming (LP) problem. Additionally, we present a heuristic to sort control loops by the descending order of their control costs in each superframe for shortening the latency of control loops in need of communication. Finally, we address the trade-off between control performance and communication cost by formulating a new multi-objective optimization problem.

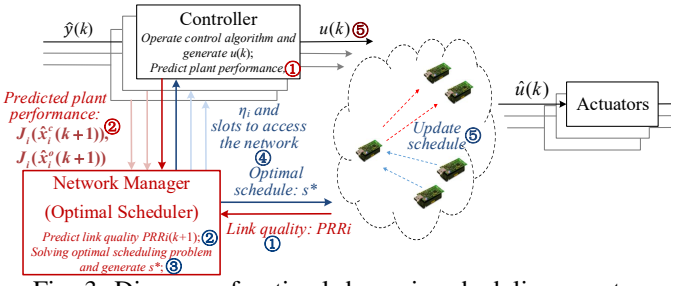


Fig. 3: Diagram of optimal dynamic scheduling events.

A. Multi-loop control system modeling

We use s to represent the network schedule of the next superframe. The number of transmission is at the center of the trade-offs between reliability and network resources, i.e., more transmissions lead to a higher packet delivery ratio (PDR) at a cost of network resources [28]. Denote η_i the number of transmission of *loop* i in schedule s. Additional $\eta_i - 1$ slots assigned to loop i are re-transmitting the same control signal in case the packet is lost in previous slots. For example, $\eta_i = 2$ indicates that loop i is assigned 2 transmission slots. Our scheduling problem is to determine and balance η_i among control loops by predicting link quality and physical system performance.

We focus on the actuation (downstream) packet scheduling problem. This is because the state observer provides robust and theoretically sound protection against loss of sensing information [16], [46], [47], and the control system performance can be more sensitive to packet loss to the actuators in certain WNCS [45]. Interested readers are referred to [48] and references therein for details about sensing packet scheduling. We will establish optimal schedule for both sensing and actuation packets in our future work.

In Sections IV-B– IV-D and V, we focus on modeling packet loss and schedule the actuation packets for the control loops in the ascending order of the loop number in each superframe. Since we focus on scheduling CFP of a IEEE 802.15.4 superframe, which employs time division multiple access (TDMA), transmission latency is bounded by the superframe length. For ease of analysis, we assume strict periodicity of actuation packets. This restriction is lifted in our simulation to allow realistic packet timing.

Remark IV.1. *For simplicity, all loops presumably have the same sampling period, which is equal to the superframe length. This is without loss of generality, and can be relaxed by applying well-established lifting technique in sampled-data control [49], i.e., representing a multi-rate system as a single-rate system with a single period being the least common multiple of all sampling periods.* □

B. Link quality

We adopt a general metric – packet reception ratio (PRR) – to represent the link quality since maximization of the successfully transmitted packets is the basic objective to most networks [50]. NM dynamically generates schedules for the WSAN based on predicted PRRs of all links. Besides, physical layer characteristics such as received signal strength indicator (RSSI), SNR, and link layer characteristics such as link quality indicator and expected transmission count also indicate the quality of wireless link [50].

1) Link quality prediction: We predict the link quality at the granularity of a superframe (sampling period) since in IEEE 802.15.4 networks, scheduling is done on a superframe basis. Because a superframe is short, typically in unit of hundreds of millisecond, the predicted link quality is assumed to remain constant within a same superframe. Holt's additive trend prediction method [51], [52] is employed to predict PRR of next m superframes,

$$\begin{aligned} S(k) &= \alpha PRR(k) + (1-\alpha)\big(S(k-1) + T(k-1)\big) \\ T(k) &= \gamma\big(S(k) - S(k-1)\big) + (1-\gamma)T(k-1) \\ \widehat{PRR}(k+m|k) &= S(k) + mT(k), \end{aligned} \tag{2}$$

where $PRR(k)$ is the current measured PRR of a specific link, $S(k)$ denotes an estimate of the current level of the series,

$T(k)$ represents an estimate of current trend (slope), m is a positive integer representing the superframes ahead, $\widehat{PRR}(k+m|k)$ is the predicted PRR m superframes ahead, α and γ ($0 < \alpha, \gamma < 1$) are the level and slope smoothing parameter, respectively.

2) Packet delivery modeling: Let a binary variable $\phi_i(k)$ denote end-to-end packet reception $\big(\phi_i(k) = 1\big)$ or loss $\big(\phi_i(k) = 0\big)$ within a superframe. PDR of actuation packets for loop i under schedule s is denoted as $\mu_{\phi_i}(s) = \mathbb{P}\big(\phi_i(k) = 1\big)$. Note that $\mu_{\phi_i}(s)$ depends on PRR of the link and the number of transmissions in schedule s. Particularly, given link failure ratio of *link* i (*loop* i) as $\beta_i = 1 - PRR_i$, we have PDR

$$\mu_{\phi_i}(s) = 1 - \beta_i^{\eta_i}. \tag{3}$$

C. Optimal scheduling formulation

At time k, controllers determine control $u(k)$ based on states $x(k)$ and physical system models (1). NM should come up with schedule $s(k)$ that optimizes overall control performance, based on states $x(k)$ and actuation commands $u(k)$ of the physical plants, wireless link quality PRR, and physical system models (1) at run-time. In fact, optimal scheduling solves for $s(k)$ based on the predicted state $x(k+1)$ which implicitly depends on schedule $s(k)$ through PDR. Specifically, state $\hat{x}_i(k+1)$ for loop i can be inferred from $x_i(k), u_i(k)$, for the cases of actuation packet reception or loss (i.e., ϕ_i),

1) actuation command packet of *loop* i at time index k arrives, and $\hat{u}_i(k)$ is actuated (closed loop):
$$\begin{aligned} \hat{u}_i^c(k) &= u_i(k), \\ x_i(k+1) = \hat{x}_i^c(k+1) &= f_i\big(x_i(k), \hat{u}_i^c(k)\big), \end{aligned} \tag{4}$$
2) actuation command packet $u_i(k)$ is lost, and $\hat{u}_i(k-1)$ is actuated (open loop):
$$\begin{aligned} \hat{u}_i^o(k) &= \hat{u}_i(k-1) \\ x_i(k+1) = \hat{x}_i^o(k+1) &= f_i\big(x_i(k), \hat{u}_i^o(k)\big). \end{aligned} \tag{5}$$

The objective for the optimal scheduling strategy is optimizing control system performance. For illustration purpose, we define a quadratic cost function of loop i as follows:

$$\mathcal{J}_i\big(x_i(k)\big) = x_i^T(k) W_i x_i(k), \tag{6}$$

where $W_i \succ 0$ is a positive definite matrix. Define the overall cost function as follows:

$$\mathcal{J}\big(x(k)\big) = \sum_{i=1}^{N} \mathcal{J}_i\big(x_i(k)\big) = x^T(k) W x(k), \tag{7}$$

where $x(k) = \big[x_1^T(k) \;\; x_2^T(k) \;\; \dots \;\; x_N^T(k)\big]^T$, and $W = \text{blkdiag}(W_1, W_2, ..., W_N)$. Here, blkdiag is the block-diagonalize operator that constructs a block diagonal matrix from input matrices. We will see in Sec. V that this objective function provides some benefits in terms of the guarantee of mean-square stability for LTI systems.

Given schedule $s(k)$, the expectation of predicted control cost for loop i, $\mathcal{J}_i\big(x_i(k+1)\big)$ is:

$$\begin{aligned} &\mathbb{E}\Big(\mathcal{J}_i\big(x_i(k+1)\big)\Big) = \\ &\sum_{\phi_i = \{0,1\}} \mathcal{J}_i\big(x_i(k+1)\big) \mathbb{P}\big(x_i(k+1) | \phi_i(k+1)\big) \mathbb{P}\big(\phi_i(k+1)\big) \\ &= \mathcal{J}_i\big(\hat{x}_i^c(\cdot)\big) \mathbb{P}\big(x_i(\cdot) = \hat{x}_i^c(\cdot) | \phi_i(\cdot) = 1\big) \mathbb{P}\big(\phi_i(\cdot) = 1\big) \\ &+ \mathcal{J}_i\big(\hat{x}_i^o(\cdot)\big) \mathbb{P}\big(x_i(\cdot) = \hat{x}_i^o(\cdot) | \phi_i(\cdot) = 0\big) \mathbb{P}\big(\phi_i(\cdot) = 0\big) \end{aligned}$$

According to (4) and (5), we have

$$\begin{aligned} \mathbb{P}\big(x_i(\cdot) = \hat{x}_i^c(\cdot) | \phi_i(\cdot) = 1\big) = 1, \\ \mathbb{P}\big(x_i(\cdot) = \hat{x}_i^o(\cdot) | \phi_i(\cdot) = 0\big) = 1. \end{aligned} \tag{8}$$

Therefore,

$$\begin{aligned} \mathbb{E}\Big(\mathcal{J}_i\big(x_i(k+1)\big)\Big) &= \mathcal{J}_i\big(\hat{x}_i^c(k+1)\big) \mu_{\phi_i}\big(s(k)\big) \\ &+ \mathcal{J}_i\big(\hat{x}_i^o(k+1)\big)\Big(1 - \mu_{\phi_i}\big(s(k)\big)\Big), \end{aligned} \tag{9}$$

where $\mathbb{E}$ is the expectation operator, with respect to the probabilities of link failure/success and the corresponding cost functions. Substituting (3) into (9) gives

$$\begin{aligned} \mathbb{E}\Big(\mathcal{J}_i\big(x_i(k+1)\big)\Big) &= \mathcal{J}_i\big(\hat{x}_i^c(k+1)\big) \\ &+ \Big(\mathcal{J}_i\big(\hat{x}_i^o(k+1)\big) - \mathcal{J}_i\big(\hat{x}_i^c(k+1)\big)\Big) \beta_i^{\eta_i}. \end{aligned} \tag{10}$$

We then calculate the expectation of overall control cost for N loops, and get (11) by simplifying derivation by ignoring time index. See Appendix A for details.

$$\mathbb{E}\big(\mathcal{J}(x)\big) = \sum_{i=1}^{N} \Big(\mathcal{J}_i(\hat{x}_i^c) \mu_{\phi_i}(s) + \mathcal{J}_i(\hat{x}_i^o)\big(1 - \mu_{\phi_i}(s)\big)\Big). \tag{11}$$

Given the objective function (11), the optimal scheduling problem is formulated as:

$$\underset{\eta_i}{\text{minimize}} \quad \mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big) \tag{12a}$$

$$\text{subject to} \quad \sum_{i=1}^{N} \eta_i \le L \tag{12b}$$

$$\eta_i \in \{0, 1, \dots, L\}, \forall i \in \{1, 2, ..., N\}, \tag{12c}$$

where L is the total number of slots assigned for all actuation flows in each superframe, which is usually customized based on the superframe length and transmission workloads. Constraint (12b) indicates the requirement of schedulability, i.e., the total number of transmissions of all actuation flows should be less than L. Constraint (12c) means that the transmission number should be a non-negative integer. Problem (12) is an integer programming problem. Furthermore, the objective function is nonlinear in η_i as can be seen from (10). Problem (12) is NP-hard [53].

D. Run-time optimal scheduling

Since the optimal scheduling problem must be solved for every superframe, its tractability is of vital importance.

1) Binary linear programming: We propose a transformation of variables to recast Problem (12) into a binary linear programming (BLP) problem. The resultant BLP problem is equivalent to Problem (12) by introducing the binary variable $\tilde{T}_{ij} \in \{0, 1\}$ that flags the magnitude of η_i, which implies the change of decision space from $\{0, 1, \dots, L\}^N$ to $\{0, 1\}^{N(L+1)}$:

$$\tilde{T}_{ij} = \begin{cases} 1, \;\; \eta_i = j, \;\; j \in \{0, 1, \dots, L\} \\ 0, \;\; \text{otherwise.} \end{cases} \tag{13}$$

We define

$$q_{ij} = \mathcal{J}_i\big(\hat{x}_i^c(k+1)\big) + \Big(\mathcal{J}_i\big(\hat{x}_i^o(k+1)\big) - \mathcal{J}_i\big(\hat{x}_i^c(k+1)\big)\Big) \beta_i^j.$$

Then we can represent $\mathbb{E}\Big(\mathcal{J}_i\big(x_i(k+1)\big)\Big)$ in (10) as

$$\mathbb{E}\Big(\mathcal{J}_i\big(x_i(k+1)\big)\Big) = \sum_{j=0}^{L} q_{ij} \tilde{T}_{ij}. \tag{14}$$

By defining
$\tilde{T} = \begin{bmatrix}\tilde{T}_{10} & \tilde{T}_{11} & \dots & \tilde{T}_{1L} & \tilde{T}_{20} & \tilde{T}_{21} & \dots & \tilde{T}_{2L} & \dots & \tilde{T}_{NL}\end{bmatrix}^T$,
we can see that the objective function (11) is linear in $\tilde{T}$,

$$\mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big) = Q\tilde{T}, \tag{15}$$

where $Q = \big[\begin{bmatrix}q_{10} & q_{11} & \dots & q_{1L}\end{bmatrix}, ..., \begin{bmatrix}q_{N0} & \dots & q_{NL}\end{bmatrix}\big]$. Problem (12) is reduced to a BLP problem as follows

$$\underset{\tilde{T}_{ij}}{\text{minimize}} \quad Q\tilde{T} \tag{16a}$$

$$\text{subject to} \quad \sum_{i=1}^{N}\sum_{j=0}^{L} j\tilde{T}_{ij} \le L \tag{16b}$$

$$\sum_{j=0}^{L} \tilde{T}_{ij} = 1 \tag{16c}$$

$$\tilde{T}_{ij} \in \{0,1\}, \forall i \in \{1,2,...,N\}, \forall j \in \{0,1,2,...,L\} \tag{16d}$$

Note that we rewrite the constraint (12b) as (16b). In order to ensure each loop i has unique η_i, we impose constraints (16c)-(16d). The transmission numbers can be recovered from $\tilde{T}$ using

$$\eta_i = \begin{bmatrix}0 & 1 & 2 & \dots & L\end{bmatrix}\begin{bmatrix}\tilde{T}_{i0} & \tilde{T}_{i1} & \tilde{T}_{i2} & \dots & \tilde{T}_{iL}\end{bmatrix}^T. \tag{17}$$

Commercially available integer programming solvers such as Gurobi®, CPLEX®, and *intlinprog* in MATLAB®, can be readily used to solve (16).

2) Linear programming relaxation: By relaxing binary constraint (16d) to $\tilde{T}_{ij} \in [0,1]$, we have a typical LP problem, which can be solved efficiently using *linprog* in MATLAB® or other LP solvers. We then convert the resultant relaxed solution to integral form by rounding η_i of (17). The complexity of LP is $O(\frac{m^3}{ln(m)}D)$ [54], where m is the dimension of decision space, i.e. $N(L+1)$, D denotes the bit length of the input data. When we set $N = 4$ and $W = I_4$, among 57,600 results, 99.98% of cases yield the optimal solutions (found by brute-force search in the feasible set). We illuminate the complexity of optimal scheduling problem when the number of control loops goes up to 200 ($N = 200$) in Fig. 4. LP relaxation shows its overwhelming advantage in reducing complexity over brute force search.

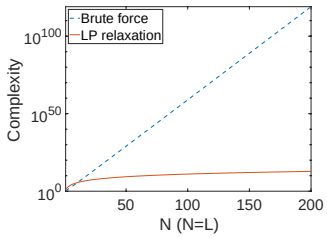


Fig. 4: Complexity of optimal scheduling problem

Remark IV.2. *The resultant η_i might be infeasible ($\sum_{i=1}^{N}\eta_i > L$) due to relaxation and rounding. Since there is a diminishing return in PDR improvement as η_i increases [28], we propose a heuristic method to achieve a feasible solution by iteratively reducing the largest element $\max_{1\le i\le N}\eta_i$ by one, until $\sum_{i=1}^{N}\eta_i \le L$.* □

Algorithm 1: Algorithm of sorting loops in each superframe

input : Transmission numbers returned by optimal scheduling: η_i, predicted costs: $Cost_i = w_i x_i(k)$, w_i, $i \in \{1,2,\dots,N\}$ are customized weight of each loop, number of slots for actuation packets: L.
output: The schedule of actuation packets in next superframe: *Schedule*

Schedule ← *zeros*(L); *Slot* ← 1;
for all i **do**
  η_i*_left* ← η_i;
  //η_i*_left* `represents unscheduled transmissions;`
Cost_matrix ← $\begin{bmatrix}1 & 2 & 3 & \dots & N\\ Cost_1 & Cost_2 & & \dots & Cost_N\end{bmatrix}$;
Sorted_loop_number ← sort loops (first row of *Cost_matrix*) in descending order of $Cost_i$ (second row of *Cost_matrix*);
while *slot*$\le L$ **and** $\sum_{i=1}^{N}\eta_i > 0$ **do**
  for i in *Sorted_loop_number* **do**
    if η_i*_left* > 0 **then**
      Schedule(Slot) ← i; *Slot* ← *Slot* + 1;
      η_i*_left* ← η_i*_left* − 1;
return *Schedule*;

E. Heuristics of sorting loops in superframes

In previous sections, we unnecessarily assume that the actuation packet of each loop is scheduled in the ascending order of the loop number. In this section, we lift that restriction and provide an algorithm to leverage the order of loops in each superframe given the solution η_i of optimal scheduling problem (16). As shown in Alg. 1, we propose to sort the actuation packet of each loop in the descending order of their costs ($Cost_i$), i.e., the loops with larger costs will be scheduled earlier so that the actuation packets of those loops will obtain shorter latency. In addition, we spread the retransmissions of same loop to shorten the latency of other loops.

F. Trade-offs between control performance and communication cost

The aforementioned optimal scheduling algorithm assigns all L slots to actuation flows. It does not consider trade-offs between control system performance and communication cost, which is defined by the total number of slots assigned to actuation commands. In practice, a smaller communication cost means lower bandwidth usage by actuation flows, which allows the network to accommodate more applications under its bandwidth constraint. Henceforth, we formulate a new optimal dynamic scheduling problem, aiming to minimize a cost function that incorporates both control performance and communication cost.

By adding a weighted term of communication cost to the cost function of (12), we have a new cost function to balance control performance and communication cost:

$$\mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big) + \epsilon\sum_{i=1}^{N}\eta_i, \tag{18}$$

where $\mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big)$ is the control performance cost as described in (7), and $\sum_{i=1}^{N}\eta_i$, the number of transmissions assigned for all actuation flows, is communication cost. Constant ϵ is used to control the weight of communication cost

relative to control performance. When ϵ approaches 0, which indicates that the communication cost in the objective function is ignored, the optimization problem is the same as the original optimization problem (12).

V. Stability analysis

The aforementioned optimal scheduling strategy (12) can improve the control performance of the multi-loop WNCS without loss of stability. In this section, we establish a sufficient condition of stability in the mean-square sense. According to [55], a discrete-time stochastic system is mean-square stable (MSS) if for any initial state $x(0)$,

$$\limsup_{k\to\infty} \mathbb{E}\big(\|x(k)x^T(k)\|\big) = 0.$$

A closed-loop system is MSS if there exists a stochastic Lyapunov function $V(x)$, such that

1) $V(0) = 0$ and $V(x) > 0, \forall x \neq 0$;
2) $\|x\| \to \infty \Rightarrow V(x) \to \infty$;
3) $\mathbb{E}\big(V(x)\big)$ decreases along system trajectories. That is,

$$\mathbb{E}\Big(V\big(x(k+1)\big)\Big) - \mathbb{E}\Big(V\big(x(k)\big)\Big) \leq 0. \tag{19}$$

Next we show that our optimal dynamic scheduling strategy can ensure mean-square stability of the closed-loop system under mild assumption: the existence of any fixed schedule such that the resultant system is MSS. A fixed schedule can be a typical periodic schedule or any static schedule that are calculated offline. We first need to determine whether there is a fixed schedule that makes the closed-loop system MSS. Here, we provide a condition to check whether systems resulted from a fixed schedule are MSS for discrete-time LTI (DT-LTI) systems as an example.

A. MSS analysis of LTI systems with fixed schedule

Consider a multi-loop DT-LTI system, with system dynamics of the loop i being given by

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k),\ u_i(k) = K_i x_i(k), \tag{20}$$

where $x_i(k) \in \mathbb{R}^{n_i}$ is the state vector, and $u_i(k) \in \mathbb{R}^{m_i}$ is the control input. Assume that the state feedback gain K_i renders the closed-loop subsystem (loop i) asymptotically stable in ideal network (network without packet loss).

To apply the stability analysis in [56], we model the closed-loop system dynamics over actuation networks with schedule s as a discrete-time stochastic system. According to [56], the closed-loop system dynamics of loop i are equivalent to the following augmented system

$$z_i(k+1) = \tilde{A}_{si}(s,k) z_i(k), \tag{21}$$

where $z_i(k) = \begin{bmatrix} x_i^T(k) & \hat{u}_i^T(k) & u_i^T(k) \end{bmatrix}^T$, and

$$\tilde{A}_{si}(s,k) = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 1-\phi_i(s) & \phi_i(s) \\ K_i A_i & 0 & K_i B_i \end{bmatrix}.$$

Similar to (4) and (5), $z_i(k+1)$ can be determined from $z_i(k)$, $u_i(k)$, $u_i(k-1)$, and $\phi_i(k)$, specifically,

1) with $\phi_i(k) = 1$,
$\hat{u}_i(k) = u_i(k),\ z_i(k+1) = \hat{z}_i^c(k+1) = \tilde{A}_{si}^c z_i(k)$,
2) with $\phi_i(k) = 0$, $\hat{u}_i(k-1)$ is adopted and
$\hat{u}_i(k) = \hat{u}_i(k-1),\ z_i(k+1) = \hat{z}_i^o(k+1) = \tilde{A}_{si}^o z_i(k)$,

where

$$\tilde{A}_{si}^c = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 0 & 1 \\ K_i A_i & 0 & K_i B_i \end{bmatrix},\ \tilde{A}_{si}^o = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 1 & 0 \\ K_i A_i & 0 & K_i B_i \end{bmatrix}.$$

Hence, the overall multi-loop control system can be rewritten as

$$z(k+1) = \tilde{A}(s,k) z(k) \tag{22}$$

where $\tilde{A}(s,k) = \mathrm{blkdiag}\big(\tilde{A}_{s1}(s,k), \tilde{A}_{s2}(s,k), \ldots, \tilde{A}_{sN}(s,k)\big)$, $z(k) = \begin{bmatrix} z_1(k) & z_2(k) & \ldots & z_N(k) \end{bmatrix}^T$. In order to prove stability properties of the closed-loop system, besides assumptions in Sec. IV-A, we make the following assumption.

Assumption V.1. *For the fixed network schedule case, sequences $\{\phi_i(k), k \in \mathbb{N}\}, \forall i \in \{1, 2, ..., N\}$, are i.i.d with the average of μ_{ϕ_i}.*

Assumption V.1 is used to establish the stability for closed-loop system with a fixed schedule case. Distribution of $\phi_i(k)$ depends on the distribution of $i.i.d$ transmission receptions and pre-defined fixed schedule s_f based on product distribution [57]. This assumption is lifted in evaluation section to allow more realistic radio propagation and noise models in TOSSIM [58].

Under Assumption V.1, we can rewrite $\tilde{A}(s,k)$ in (22) as

$$\tilde{A}(s,k) = \tilde{A}_0 + \sum_{i=1}^{N} \tilde{A}_i p_i(k), \tag{23}$$

where $p_i(k)$ are i.i.d. random variables with $\mathbb{E}\big(p_i(k)\big) = 0$, variance $\mathrm{Var}\big(p_i(k)\big) = \sigma_{p_i}^2$, and $\mathbb{E}\big(p_i(k)p_j(k)\big) = 0, \forall i, j \in \{1, 2, \ldots, N\}$,

$$\tilde{A}_0 = \mathrm{blkdiag}(\tilde{A}_{01}, \tilde{A}_{02}, \ldots, \tilde{A}_{0N}),$$
$$\tilde{A}_1 = \mathrm{blkdiag}(A_{\phi_1}, \mathbf{0}, \ldots, \mathbf{0}),$$
$$\tilde{A}_2 = \mathrm{blkdiag}(\mathbf{0}, A_{\phi_2}, \mathbf{0}, \ldots, \mathbf{0}),$$
$$\vdots$$
$$\tilde{A}_N = \mathrm{blkdiag}(\mathbf{0}, \mathbf{0}, \ldots, A_{\phi_N}),$$
$$\tilde{A}_{0i} = \begin{bmatrix} A_i & B_i & 0 \\ 0 & \big(1-\mu_{\phi_i}(s)\big)I & \mu_{\phi_i}(s)I \\ K_i A_i & 0 & K_i B_i \end{bmatrix},$$
$$A_{\phi_i} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu_{\phi_i}(s)I & -\mu_{\phi_i}(s)I \\ 0 & 0 & 0 \end{bmatrix},\ \sigma_{p_i}^2(s) = \frac{1}{\mu_{\phi_i}(s)} - 1.$$

Here, we let $p_i(k) = 1 - \frac{\phi_i(k)}{\mu_{\phi_i}}$ be binary random variable that takes 1 or $1 - \frac{1}{\mu_{\phi_i}}$ with $\mathbb{P}\big(p_i(k) = 1\big) = 1 - \mu_{\phi_i}$ and $\mathbb{P}\big(p_i(k) = 1 - \frac{1}{\mu_{\phi_i}}\big) = \mu_{\phi_i}$. From Assumption V.1, we have that $p_i(k)$ is i.i.d with $\mathbb{E}\big(p_i(k)\big) = 0$ and $\mathrm{Var}\big(p_i(k)\big) = \sigma_{p_i}^2$.

For the discrete-time stochastic system (23), Lemma V.1 gives a condition to check whether the system is MSS. MSS can be verified directly by solving the Lyapunov equation in P, $\tilde{A}_0^T P \tilde{A}_0 - P + \sum_{i=1}^{N} \sigma_{p_i}^2 \tilde{A}_i^T P \tilde{A}_i + I = 0$, and checking whether $P > 0$.

Lemma V.1. [55, pp. 131] *System* (21) *is MSS if and only if there exists a positive definite matrix P satisfying*

$$\tilde{A}_0^T P \tilde{A}_0 - P + \sum_{i=1}^{N} \sigma_{p_i}^2 \tilde{A}_i^T P \tilde{A}_i < 0. \tag{24}$$

Remark V.2. *For fixed pre-defined schedule s_f, if control loops are independent, where the states of one loop do not interact with those of other loops, each loop i can derive its own positive definite matrix (denoted as P_i) separately as single control loop in Lemma V.1. We have $P = \text{blkdiag}(P_1, P_2, ..., P_N)$.* □

B. Stability condition of optimal scheduling

Given the existence of a fixed schedule which renders the closed-loop system MSS, we can establish that the closed-loop system resulted from the optimal schedule is also MSS.

Proposition V.3. *If there exists a fixed schedule s_f such that the resultant closed-loop system is MSS, and $\mathcal{J}(x)$ is a stochastic Lyapunov function with s_f, then the closed-loop system with the optimal schedule s^* derived by solving* (12) *is also MSS.*

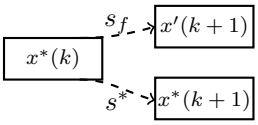


Fig. 5: Diagram of stability proof (apply the fixed schedule s_f and the optimal schedule $s^*(k)$ to $x^*(k)$, and then get $x'(k+1)$ and $x^*(k+1)$, respectively)

Proof. As shown in Fig. 5, we apply both the stabilizing fixed schedule s_f and the optimal schedule $s^*(k)$ to any state $x^*(k)$, and then get $x'(k+1)$ and $x^*(k+1)$, respectively.

Since $\mathcal{J}(x)$ is a stochastic Lyapunov function of the closed-loop system resulted from a fixed schedule s_f, $\mathcal{J}(x)$ satisfies $\mathcal{J}(x) > 0, \forall x \neq 0$, $\mathcal{J}(x) \to \infty$ as $\|x\| \to \infty$, and $\mathbb{E}\big(\mathcal{J}(x)\big)$ decreases along trajectories of the system, according to (19). Therefore,

$$\mathbb{E}\Big(\mathcal{J}\big(x'(k+1)\big)\Big) \leq \mathbb{E}\Big(\mathcal{J}\big(x^*(k)\big)\Big). \tag{25}$$

Because the schedule s^* minimizes the objective function $\mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big)$ in the optimization problem (12), we have

$$\mathbb{E}\Big(\mathcal{J}\big(x^*(k+1)\big)\Big) \leq \mathbb{E}\Big(\mathcal{J}\big(x'(k+1)\big)\Big). \tag{26}$$

Combining (25) and (26), we derive

$$\mathbb{E}\Big(\mathcal{J}\big(x^*(k+1)\big)\Big) \leq \mathbb{E}\Big(\mathcal{J}\big(x^*(k)\big)\Big). \tag{27}$$

For the optimally scheduled system (i.e. $s = s^*$), $\mathbb{E}\big(\mathcal{J}(x)\big)$ decreases along trajectories of the system, and satisfies $\mathcal{J}(x) > 0, \forall x \neq 0$, and $\mathcal{J}(x) \to \infty$ as $\|x\| \to \infty$. Therefore, $\mathcal{J}(x)$ is also a stochastic Lyapunov function of the optimally scheduled system. □

Remark V.4. *The stability condition in this section is only valid for the optimal solution in Sec. IV-D1, and is not valid for the near-optimal solution in Sec. IV-D2. Actually, as long as the linear programming relaxation gives $\mathbb{E}\Big(\mathcal{J}^*\big(x(k+1)\big)\Big)$ satisfying the inequality* (26)*, stability holds; otherwise, we can use the fixed schedule to guarantee stability.* □

Remark V.5. *For DT-LTI system* (22)*, and $P \succ 0$ satisfying Lemma V.1 with s_f, we can interpret the function $\mathcal{J}(x) = x^T P x$ as a stochastic Lyapunov function with s_f, see, for example [55, pp. 132], and thus $\mathcal{J}(x)$ is also a Lyapunov function of the optimally scheduled system. This justifies the adoption of a quadratic objective function in* (7). □

Remark V.6. *Although we set $\mathcal{J}(x)$ as a quadratic function to analyze MSS for DT-LTI systems, Proposition V.3 holds for other forms of $\mathcal{J}(x)$. That is, if there is a stochastic Lyapunov function $V(x)$ for nonlinear systems with a fixed schedule [59], then $V(x)$ is also a stochastic Lyapunov function for the closed-loop system rendered by the optimal schedule that minimizes $\mathbb{E}\big(V(x)\big)$ in* (12). □

VI. Evaluation

This section presents a systematic case study of the proposed scheduling strategy. On the physical plant side, we use four 3-state double water-tank systems that share the same wireless network. On the network side, we first apply simulations to evaluate the proposed optimal dynamic scheduling strategy and the efficacy of the network reconfiguration protocol. We simulate stochastic packet loss patterns using the TOSSIM simulator for IEEE 802.15.4 networks, and evaluate our strategy under 1) constant and variable network background noise levels and 2) pulse physical disturbance to the control plant. Furthermore, we implement the dynamic scheduling algorithm and mechanisms in a real IEEE 802.15.4 wireless network. We then build a network-in-the-loop simulator that integrates a real wireless network and simulated physical plants, in order to empirically evaluate the the proposed optimal dynamic scheduling strategy.

A. Simulation settings

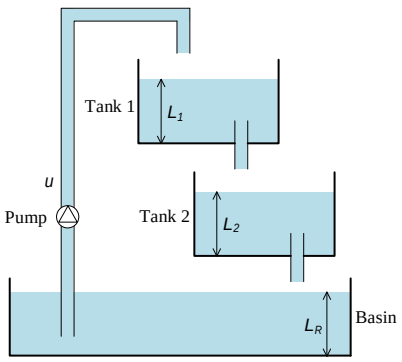


Fig. 6: Diagram of double water-tank systems

1) Physical control system: Consider four independent 3-state double water-tank systems, as shown in Fig. 6, each of which is modeled as follows [37], [40]:

$$\begin{aligned}
\dot{L}_1 &= \frac{1}{\rho A_1}(\alpha u - \frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1}) \\
\dot{L}_2 &= \frac{1}{\rho A_2}(\frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1} - \frac{\sqrt{\rho g}}{\rho R_2}\sqrt{L_2}) \\
\dot{L}_R &= \frac{1}{\rho A_R}(\frac{\sqrt{\rho g}}{\rho R_2}\sqrt{L_2} - \alpha u)
\end{aligned} \tag{28}$$

TABLE I: System parameters

PLANT1				PLANT2			
par	value	par	value	par	value	par	value
A_1	0.01	R_1	0.0006	A_1	0.12	R_1	0.0006
A_2	0.006	R_2	0.0008	A_2	0.007	R_2	0.0008
A_R	1	α	10	A_R	1	α	10

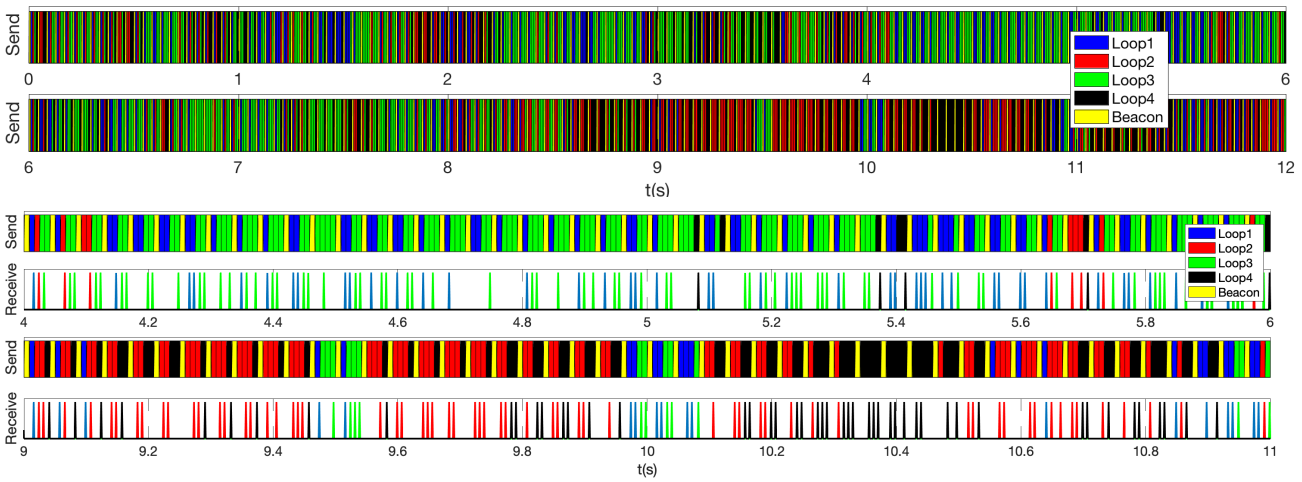


Fig. 7: Optimal scheduling under constant noise -76 dBm

where L_1, L_2, L_R are the liquid levels of the upper tank, lower tank and the basin, respectively; A_1, A_2, A_R are the cross-sectional areas of the tanks; and R_1, R_2 are the resistance parameters of pipes of upper and lower tanks. We discretize the continuous-time model (28) using the Euler method with sampling period of Δt, and have the discrete-time model

$$\begin{bmatrix} L_1(k+1) \\ L_2(k+1) \\ L_R(k+1) \end{bmatrix} = \begin{bmatrix} 1-\frac{\Delta t\sqrt{\rho g}}{\rho^2 R_1 A_1 \sqrt{L_1}} & 0 & 0 \\ \frac{\Delta t\sqrt{\rho g}}{\rho^2 R_1 A_2 \sqrt{L_1}} & 1-\frac{\Delta t\sqrt{\rho g}}{\rho^2 R_2 A_2 \sqrt{L_2}} & 0 \\ 0 & \frac{\Delta t\sqrt{\rho g}}{\rho^2 R_2 A_R \sqrt{L_2}} & 1 \end{bmatrix} \begin{bmatrix} L_1(k) \\ L_2(k) \\ L_R(k) \end{bmatrix} + \begin{bmatrix} \frac{\alpha \Delta t}{\rho A_1} \\ 0 \\ -\frac{\alpha \Delta t}{\rho A_2} \end{bmatrix} u(k).$$

There are two types of plants, denoted by PLANT1 and PLANT2, that have different system parameters, shown in Table. I. Systems 1 and 3 are of type PLANT1, and systems 2 and 4 are of type PLANT2. We have included multi-rate functionality in MATLAB/Simulink simulations of physical plants and the controllers. That is, plant models run at a high frequency of 960 Hz, whereas controllers and state observers execute at a relatively low frequency of 24 Hz.

For four loops, we design state feedback controllers for reference tracking. To evaluate the tracking performance, we choose the mean absolute error (MAE) metric to compare the performance among each control loop in order to illustrate the intuition and advantages of the scheduling strategy:

$$MAE = \frac{1}{n+1}\sum_{k=0}^{n} |x(k) - x_{ref}(k)|, \qquad (29)$$

where n is the number of samples, and x_{ref} is the reference state.

2) Wireless network: We simulate the IEEE 802.15.4 beacon-enabled wireless network. We have developed the functionalities of NM and media access in the simulator. NM predicts the link quality of the next period and solves the optimal scheduling problem, then transmits the updated schedule within the beacon message. We utilize multi-rate functionality in MATLAB/Simulink to support the timing of accessing the media by wireless nodes of different control loops. The resultant media access schedule and transmission outcomes can be derived from the simulator (as shown in Figs. 7, 12, and 21). Since we propose to use fixed scheduling for sensing flows in Sec. IV-A, in our simulation, we focus on scheduling actuation flows by assuming sensors having wired connection to controllers. Each superframe has five slots and the slot duration is 8.3 ms, and the superframe is updated at 24 Hz. The first slot is assigned for a beacon frame. The following four CFP slots are assigned for actuation flows of four control loops. Given W as the identity matrix in the objective function $\mathcal{J}(x)$ in (7), we solve the relaxed linear optimization problem described in Sec. IV-D2 using MATLAB/*linprog* solver. In simulation, we collect wireless traces from 4 links (8 nodes) of the WSAN testbed at Washington University. As described in Sec. VI-B1, we get packet loss traces using RSSI and set controlled noise strength as inputs of the TOSSIM simulator. For simplicity, we use single channel in evaluation. Note that the supported number of control loops can be scaled up by simultaneously accessing up to 16 channels of IEEE 802.15.4 PHY. [60]

B. Simulation results

We first evaluate PRR prediction described in Sec. IV-B. We then run the WNCS simulations under different levels of constant network background noise. We then evaluate the performance of our optimal scheduling strategy under variable background noise to show its adaptability and optimality, comparing with the periodic scheduling mechanism. In addition, we evaluate how the optimal scheduling strategy performs under pulse physical disturbance.

1) Results of link quality prediction: In our study, wireless traces from 4 links of the WSAN testbed at Washington University [14] have been collected, which contain the connectivity and RSSI data [37]. In addition, we use controlled background noise strength to simulate various network conditions. Both the RSSI and controlled noise strength are fed into a high-fidelity wireless simulator – TOSSIM [45], [58]. Fig. 8 shows PRRs (91,000 packets for each data point) of four links under controlled noise levels. The PRRs vary among

links under the same noise levels since the RSSIs are different. The PRR under the lowest noise level (-84 dBm) is the highest. Under the same noise levels, links with higher RSSIs ($link1 > link2 = link4 > link3$) yield higher PRRs.

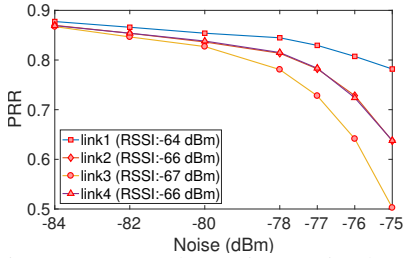


Fig. 8: PRRs under various noise levels

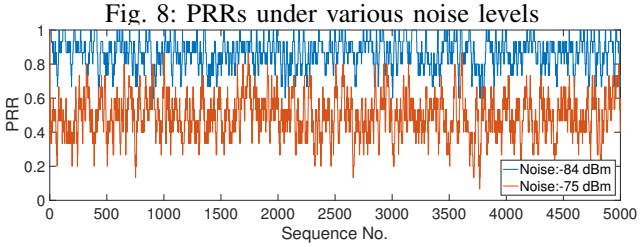


Fig. 9: Sliding-window PRRs of link 3

Fig. 9 shows the sliding-window PRRs of link 3 under noise levels of -84 dBm and -75 dBm, respectively. The horizontal axis is the number of packets transmitted via link 3. The window size is 15 in this case study. 1-step PRR prediction results are shown in Fig. 10. We use link 3 under noise level of -75 dBm as an example, and we choose $\alpha = 0.9, \gamma = 0.1$ in (2). We can see that PRR prediction (red dashed line) matches well with measured PRR (blue solid line). The mean absolute error (MAE) of the PRR predictions is shown in Fig. 11. The prediction error increases as the prediction step size increases. 1-step prediction error is less than 4%, and 5-step prediction error is less than 10%. Note that as the noise level increases from -84 dBm to -75 dBm, the prediction error increases. This indicates that the noise level affects the prediction accuracy. However, we achieve more than 90% of prediction accuracy for all simulated scenarios.

2) Performance under constant background noise: We run the WNCS simulations of optimal (OPT) scheduling under several background noise levels. Our baseline is the WNCS that adopts a static periodic schedule as shown in Fig. 12, in which GTS slots are uniformly scheduled to the four control loops. Under noise level of -76 dBm, the optimal schedule is shown in Fig. 7, and the ratios of slot allocation for each control loop in different time intervals are shown in Fig. 13. Since the sizes of tanks of PLANT1 are smaller than those of PLANT2 as shown in Table.I, PLANT1 (loops 1 and 3) is more sensitive to packet loss and performs worse than PLANT2 (loops 2 and 4) during transient responses (first 4 s). During the first 4 s, the NM scheduled most of slots to loop 1 (24.2%) and loop 3 (39.6%) and much less slots to loop 2 (17.2%) and loop 4 (19.0%). More slots are scheduled to loop 3 than loop 1 since loop 3 has worse link quality as shown in Fig. 8. Fig. 14 shows the responses of the upper tanks of the four loops. The OPT scheduling significantly improves the control performance of loops 1 and 3 and maintains similar performance of loops 2 and 4, compared with the periodic scheduling.

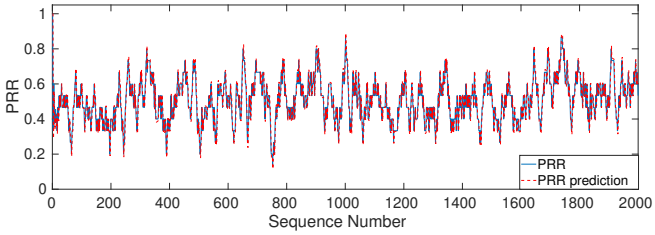


Fig. 10: 1-step PRR prediction under noise -75dBm (link 3)

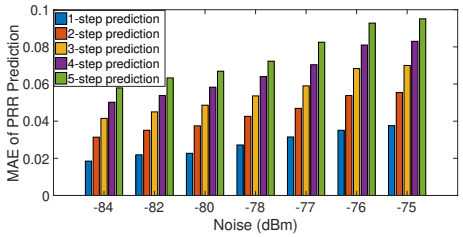


Fig. 11: PRR prediction errors of link 3 under various noise

In addition, to show the adaptability of our OPT scheduling with respect to physical disturbance, we add pulse physical disturbance to loops 1 and 3 at $t = 4$ s, and to loops 2 and 4 at $t = 9$ s. As shown in Figs. 7 and 13, during $t = 4$ to 6 s, most of the slots are assigned to loop 1 (32.8%) and loop 3 (58.4%), and only a few slots are assigned to loop 2 (5.2%) and loop 4 (3.6%) since they are in steady states. During $t = 9$ to 11 s, most of the slots in OPT schedule are scheduled to loop 2 (43.2%) and loop 4 (38.1%). These results show that our OPT scheduling can adjust to physical disturbance. For a more extensive performance evaluation illustrating the advantage of OPT scheduling over periodic scheduling, we show aggregated response curves of periodic scheduling and OPT scheduling for 50 rounds of simulation, respectively in Figs. 15 and 16. We can see the obvious advantage of OPT scheduling over periodic scheduling, especially for loop 3.

We run simulations of three scheduling strategies: (1) combining OPT scheduling and sorting with identical weights (OPT scheduling + Sorting), (2) OPT scheduling, and (3) periodic scheduling, for 50 times. Fig. 17 shows the boxplots of MAEs of each scheduling strategy under different noise levels. The control performance degrades as the background noise increases. The OPT scheduling outperforms the periodic scheduling for all background noise levels. The advantage of the OPT scheduling becomes more apparent as the link

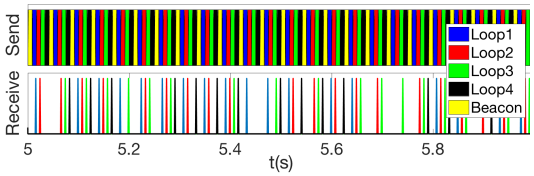


Fig. 12: Periodic scheduling under noise -76 dBm

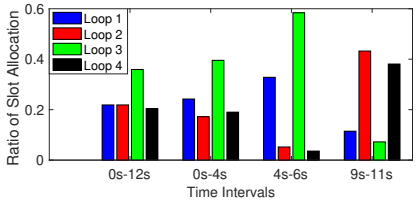


Fig. 13: Slot allocation in various time intervals

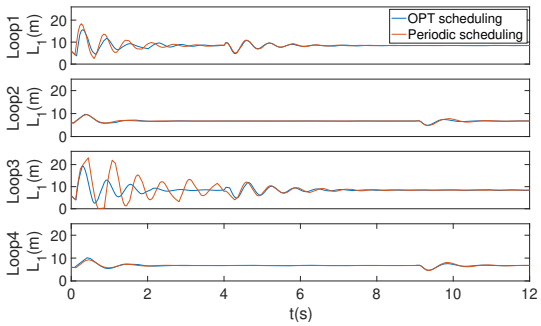


Fig. 14: Response curves under noise level of -76 dBm

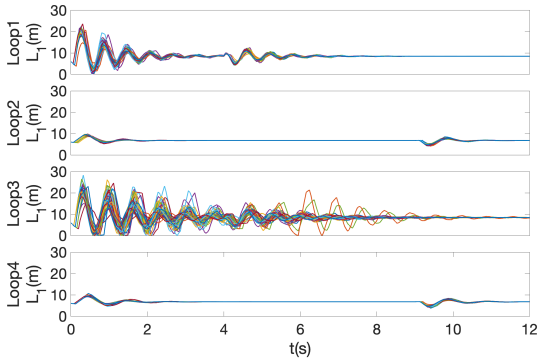


Fig. 15: Aggregated response curves of periodic scheduling under noise level of -76 dBm (50 rounds)

quality degrades. This is because the OPT schedule adjusts transmissions based on link quality and control performance and thus is more robust to noise. The sorting algorithm further improves the control performance by considering latency. We also compare the sum of the quadratic cost function over the simulation interval. As shown in Fig. 18, the cost function results are consistent with the MAE results.

3) Performance under variable background noise: In this section, we evaluate our OPT scheduling under variable background noise to show its adaptability and optimality when network conditions change. Variable background noise patterns are shown in Fig. 19. In the first 5 s, the noise levels of links 1 and 2 are -78 dBm, and those of links 3 and 4 are -75 dBm. Therefore the PRRs of links 3 and 4 are lower than links 1 and 2. The PRR of link 3 is the worst as shown in Fig. 8. The background noise changes at $t = 5$ s. The noise strength of links 1 and 2 increases to -75 dBm, and that of links 3 and 4 decrease to -84 dBm. The PRR of link 2 becomes the

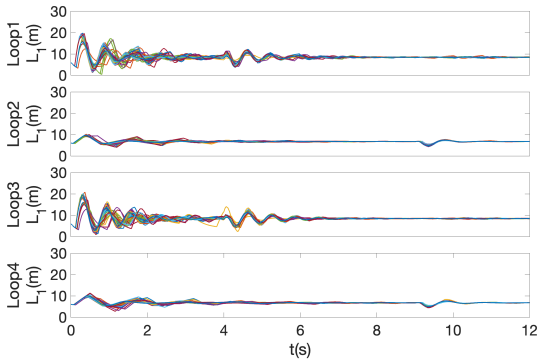


Fig. 16: Aggregated response curves of OPT scheduling under noise level of -76 dBm (50 rounds)

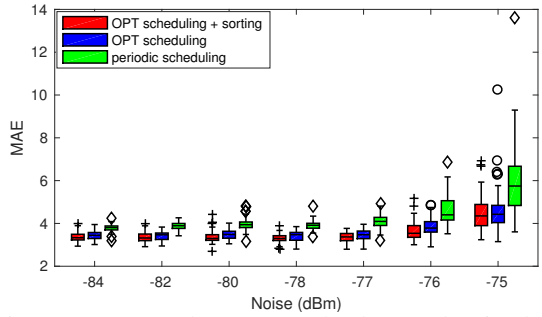


Fig. 17: MAE under constant background noise levels

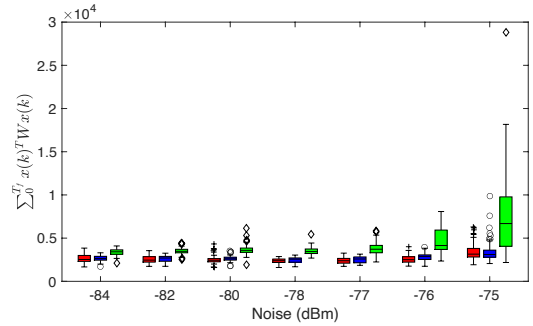


Fig. 18: Cost function under constant background noise levels

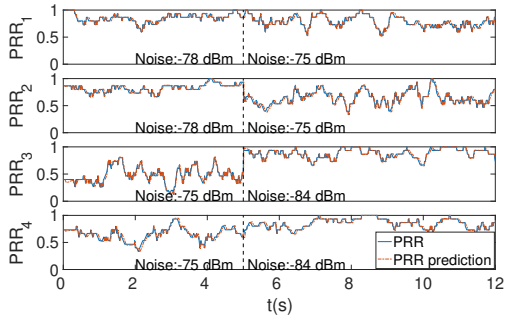


Fig. 19: Run-time link quality variation

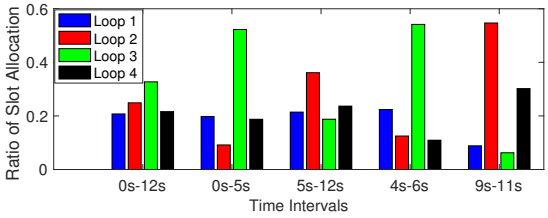


Fig. 20: Slot allocation under variable noise levels

worst in this case.

Under the noise levels shown in Fig. 19, the OPT schedule is shown in Fig. 21, and the ratios of slot allocation are shown in Fig. 20. The NM schedules more slots to loop 3 (52.3%) than other loops during the first 5 s because loop 3 has the worst network condition. The NM in the variable noise levels schedules more slots to loop 4 than in the constant noise case during the first 5 s since link 4 has the worse network condition than links 1 and 2. More slots are scheduled to loop 2 (36.1%) during the last 7 s (5 s to 12 s) since link 2 has the worst network condition. Due to physical disturbance at 4 s to loops 1 and 3, and at 9 s to loops 2 and 4, many slots from 4 s to 6 s are assigned to loops 1 (22.4%) and 3 (54.2%), and many slots from 9 s to 11 s are assigned to loops 2 (54.7%) and 4 (30.2%). The response curves of OPT and periodic scheduling

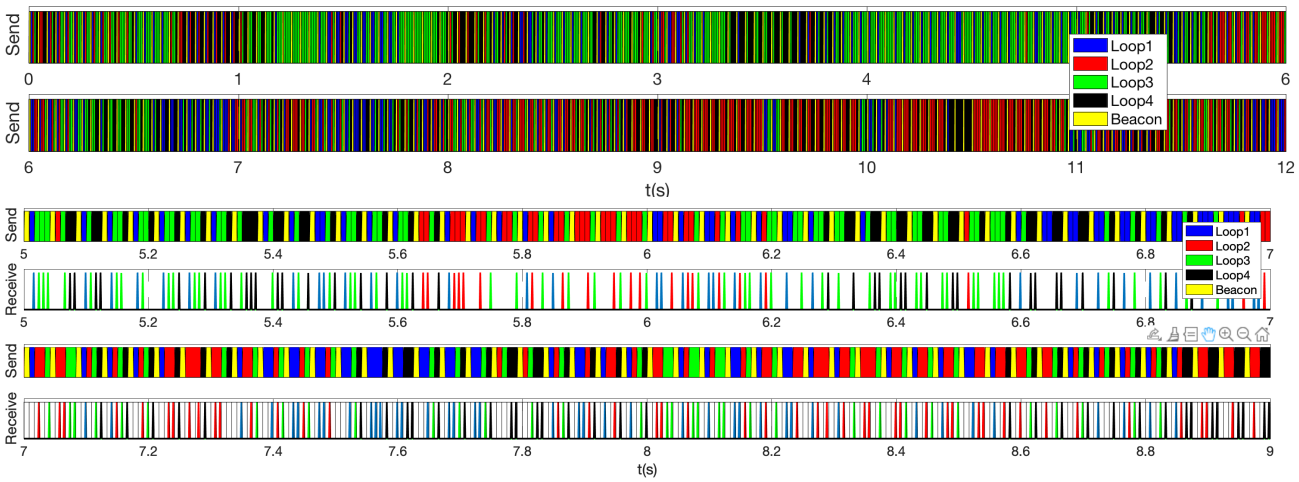


Fig. 21: Optimal scheduling under variable noise levels

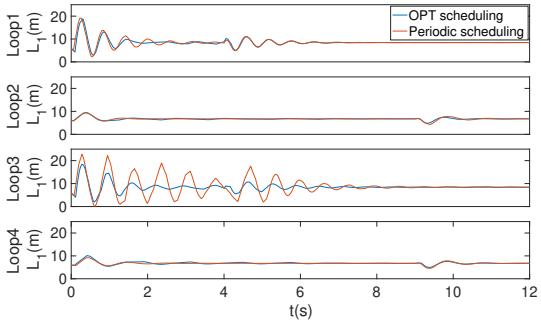


Fig. 22: Response curves under variable noise levels

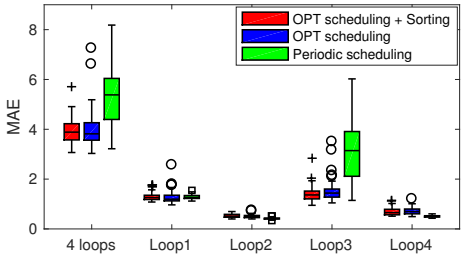


Fig. 23: MAE under variable noise levels

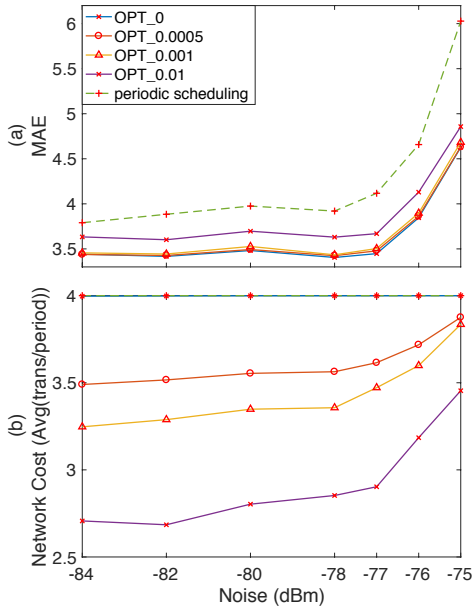


Fig. 24: Trade-offs between control performance and communication cost (with variable ϵ)

are shown in Fig. 22. The control performance using the OPT scheduling is improved for loops 1 and 3 compared with the periodic scheduling, and remains similar for loops 2 and 4. Therefore, we can conclude that our OPT scheduling can adapt to both physical disturbance and varying network condition at the same time.

Statistical results of control performance under variable noise levels are shown in Fig. 23. In terms of the total MAEs of four control loops (first group of the boxplots), the OPT scheduling outperforms the periodic scheduling, the OPT scheduling combined with sorting is better than only the OPT scheduling. The OPT scheduling optimizes the total cost function of all control loops by allocating more network resources to needy loops and links at run-time. When we look into the performance of individual control loop, compared with the periodic scheduling, the control performance of loop 3 is significantly improved by the OPT scheduling since loop 3 is allocated more network resources by the OPT scheduling.

The performance of loops 2 and 4 downgrades a little since they have relatively low MAEs and therefore less allocated network resources. Note that the extent of improvement in loop 3 is much larger than the downgrade in loops 2 and 4. The results show that the OPT scheduling can balance the network resources allocation according to link quality and control performance among multiple loops.

C. Trade-off between control performance and communication cost

We now evaluate the extended scheduling algorithm incorporating both control performance and communication cost proposed in Sec. IV-F. The new optimization problem is solved by brute-force search in the feasible set. We evaluate the computation latency of the brute-force search in this problem setting in MATLAB/Simulink on a 2.6 GHz Intel Core i7

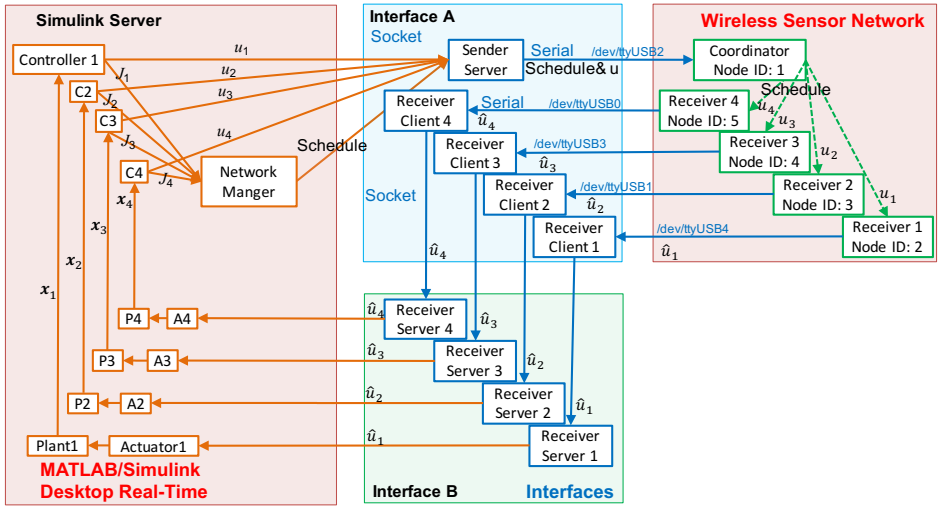


Fig. 25: Architecture for network-in-the-loop simulation

processor. The worst-case latency is 1 ms over 9000 runs, which indicates the tractability of (18) in this case study. Fig. 24 shows the overall control performance and communication cost of four control loops under physical disturbance and variable wireless noise. Each data point of Fig. 24(a) shows the average MAE of 50 rounds of simulations. Each data point of Fig. 24(b) shows the average total number of transmissions of four control loops per periods over 50 rounds of simulations. The numerical values of OPT_ϵ in legends are variable ϵ. We can see that as the weight of communication cost ϵ increases, the optimal scheduler is more likely to avoid transmitting an actuation command when control performance is relatively good. As a result, communication cost is saved at the cost of a moderate deterioration of control performance, compared with the original OPT scheduling (OPT_0) and periodic scheduling. By fine-tuning ϵ, we can achieve comparable control performance with original OPT scheduling at a significantly lower communication cost (e.g. when $\epsilon = 0.001$).

D. Network-in-the-loop Simulation

1) System design and implementation: The architecture of the network-in-the-loop simulator is illustrated in Fig. 25. The system comprises three computers and a WSAN. The physical plants, controllers, sensors, actuators, and NM are simulated in MATLAB/Simulink Desktop Real-time (SLDRT) running on a computer called the *Simulink Server*. SDLRT provides a real-time kernel for executing Simulink models, including library blocks that connect to I/O devices [61]. The two additional computers, *Interfaces A* and *B* provide interfaces to coordinate the Simulink simulations and wireless communication. The three computers are connected through Ethernet.

The WSAN is a star network with five TelosB nodes, each equipped with TI CC2420 (2.4 GHz) radio compliant with the IEEE 802.15.4 standard and a TI MSP430 microcontroller. One of the TelosB node serves as the Coordinator and the other TelosB nodes as the wireless interfaces of the actuators. The TelosB nodes communicate with each other over the wireless network based on the IEEE 802.15.4 MAC and the transmission schedule.

Interface A communicates with the coordinator through USB. Interface A emulates the network gateway by relaying 1) the actuation commands u from the controllers and 2) the transmission schedule s^* generated by the NM to the coordinator, which then forwards them to the devices over the WSAN.

Interface A and Interface B are also responsible for forwarding the actuation commands from the TelosB nodes to the actuators simulated in Simulink. When a TelosB node receives actuation command $\hat{u}$, it sends it to Interface A over USB. Interface A then forwards it to Interface B, which in turn feeds it to the actuators simulated in SLDRT on the Simulink Server. This process emulates an actuator that receives actuation commands from an WSAN over its wireless interface.

We implement our dynamic scheduling mechanism over the Time Slotted Channel Hopping (TSCH) MAC layer of the IEEE 802.15.4-2015 [62], [63] in Contiki OS. Channel hopping is disabled to be consistent with IEEE 802.15.4 MAC. The coordinator packages and broadcasts beacon frames containing updated schedules s^*. All nodes update their schedules based on the beacon frame received during run-time. The actuation commands u are transmitted to corresponding receiver nodes in to the scheduled time slots.

For the rest of the experiments, four double-water-tank systems are simulated in SLDRT, the parameters of which are listed in Table II. The sampling period of each control loop is 1.5 s. The duration of each round of network-in-the-loop simulation is 150 s. In order to emulate packet losses in a controlled fashion, we intentionally drop packets on the receiver side based on TOSSIM traces following similar approaches employed in previous wireless control experiments [56], [64] .

TABLE II: Parameters of network-in-the-loop simulations

PLANT1				PLANT2			
par	value	par	value	par	value	par	value
A_1	0.1	R_1	0.0006	A_1	0.1	R_1	0.0006
A_2	0.13	R_2	0.0008	A_2	0.13	R_2	0.0008
A_R	1	α	10	A_R	1	α	10

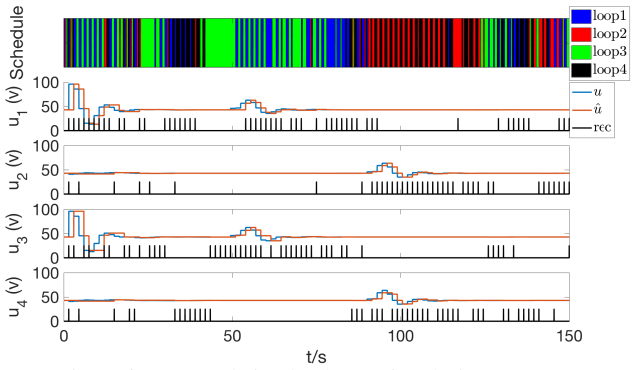


Fig. 26: Network-in-the-loop simulation (OPT)

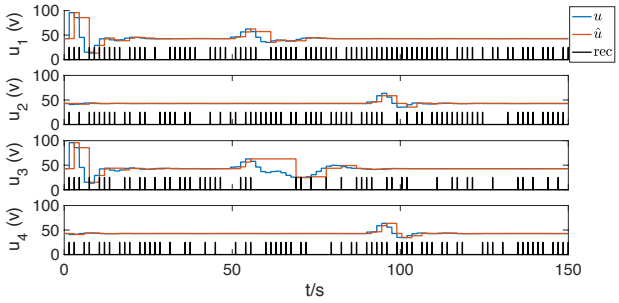


Fig. 27: Network-in-the-loop simulation (periodic scheduling)

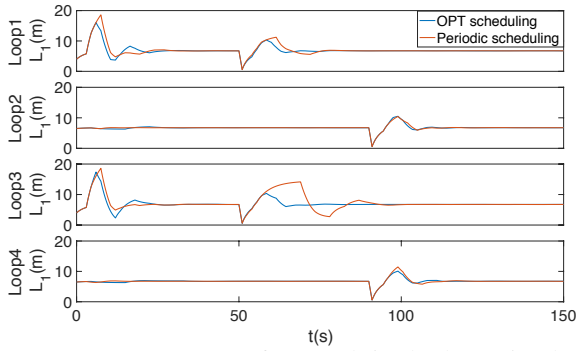
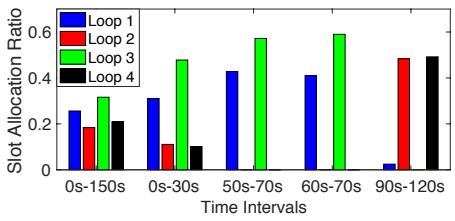
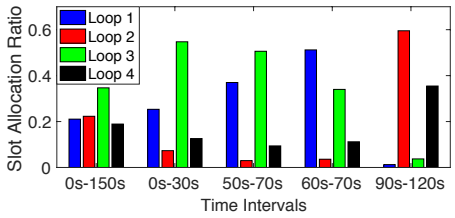


Fig. 28: Response curves of network-in-the-loop simulations

2) Experimental results: First, we run experiments under physical disturbance. We inject pulse physical disturbance into loops 1 and 3 at $t = 50$ s, and into loops 2 and 4 at $t = 90$ s. The background noise level is -75 dBm (PRRs of links 1 to 4 are 78%, 63%, 50%, and 63%, respectively). Figs. 26 and 27 show how actuation command packets of each control loop are transmitted with OPT and periodic scheduling, respectively. The first sub-figure of Fig. 26 shows the slot allocation in the optimal schedule. The rest of the sub-figures show u, $\hat{u}$, and flag of packet reception rec of each control loop. When u varies significantly and hence it is critical to updating u, OPT allocates more transmission slots to that loop. Consequently, $\hat{u}$ tracks u better when compared with fixed periodic scheduling in Fig. 27. As shown in Fig. 29a, during $t = 50$ to 70 s, most of the slots are assigned to loop 1 (41.0%) and loop 3 (59.0%), while none of the slots are assigned to loop 2 and loop 4 since they are in steady states. In addition, more slots are allocated to loop 3 than loop 1 since loop 3 suffers from worse link quality. During $t = 90$ to 120 s, most of the slots in the OPT schedule are scheduled to loop 2 (48.3%) and loop 4 (49.2%). These results show that OPT can effectively adapt to physical disturbance.



(a) Constant noise level



(b) Varying noise level

Fig. 29: Slot allocation of optimal schedule

Next we evaluate OPT under variable background noise. In the first 60 s, the noise levels of links 1 and 2 are -84 dBm (the PRRs of links 1 and 2 are 89% and 88%, respectively), and those of links 3 and 4 are -75 dBm (50% and 63%, respectively). The background noise changes at $t = 60$ s. The noise strength of links 1 and 2 increases to -75 dBm (78% and 63%, respectively), and that of links 3 and 4 decreases to -84 dBm (88% and 88%, respectively). The physical disturbance remains the same as previous settings. The ratios of slot allocation are shown in Fig. 29b. Loop 3 is scheduled more slots (54.7%) than other loops during the first 30 s because loop 3 has the worst network condition. In order to adjust to varying noise level, more slots from 60 s to 70 s are assigned to loop 1 (51.2%) than loop 3 (34.0%), and more slots from 90 s to 120 s are assigned to loops 2 (59.5%) than 4 (35.4%), compared with Fig. 29a.

Experimental results under both physical disturbance and wireless interference are consistent with the simulation results in Sec. VI-B, which shows the feasibility and efficacy of OPT in real IEEE 802.15.4 networks.

VII. Conclusions

This paper proposes an optimal dynamic scheduling approach that optimizes control performance of multi-loop systems by allocating limited network resources based on both plant and network states at run-time. We formulate the optimal scheduling problem as a nonlinear integer programming problem, and then relax it to a linear programming problem for computational efficiency. In addition, we provide a stability condition for the wireless networked control system that adopts the optimal scheduling. A systematic evaluation is performed based on four double water-tank systems and simulations of a realistic IEEE 802.15.4 wireless network. Furthermore, we design and implement the optimal scheduling approach on IEEE 802.15.4 devices, and network-in-the-loop simulation integrating real wireless networks and simulated physical plants. Simulation and experimental results show that dynamic optimal scheduling enables the system to adapt to wireless interference and physical disturbances. This work therefore demonstrates the advantages of a cyber-physical

approach to transmission scheduling based on both wireless and physical states in wireless control systems.

ACKNOWLEDGMENT

The authors would like to thank Dr. Arvind Raghunathan at Mitsubishi Electric Research Laboratories for helpful discussions.

APPENDIX A
CALCULATING EXPECTATION OF OVERALL CONTROL COST

Simplify derivation by ignoring time index,

$$\mathbb{E}\big(\mathcal{J}(x)\big) = \sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\}}} \sum_{i=1}^{N} \mathcal{J}_i(x_i)\mathbb{P}(x_1,...,x_N), \tag{30}$$

where

$$\mathbb{P}(x_1,...,x_N) = \sum_{\substack{\phi_i=\{0,1\},\\ \text{for } i=\{1,\dots,N\}}} \mathbb{P}(x_1,...,x_N|\phi_1,...,\phi_N)\mathbb{P}(\phi_1,...,\phi_N).$$

For given schedule $s(k)$, we assume packet deliveries among different data links are independent. Thus we have

$$\mathbb{E}\big(\mathcal{J}(x)\big) = \sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\}}} \Big[\Big(\sum_{i=1}^{N}\mathcal{J}_i(x_i)\Big)\cdot \Big(\sum_{\substack{\phi_i=\{0,1\},\\ \text{for } i=\{1,\dots,N\}}} \mathbb{P}(x_1|\phi_1)\cdot...\cdot\mathbb{P}(x_N|\phi_N)\cdot\mathbb{P}(\phi_1)\cdot...\cdot\mathbb{P}(\phi_N)\Big)\Big]. \tag{31}$$

Define Δ_i, $\Delta_i = 1$ if $x_i = \hat{x}_i^c$, and $\Delta_i = 0$ if $x_i = \hat{x}_i^o$. According to (8), we consider loop m,

$$\begin{aligned}
\mathbb{E}\big(\mathcal{J}(x)\big) &= \sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\}}} \Big(\sum_{i=1}^{N}\mathcal{J}_i(x_i)\Big)\prod_{i=1}^{N}\mathbb{P}(\phi_i=\Delta_i)\\
&= \sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\},\\ i\neq m}} \Big[\Big(\mathcal{J}_i\big(\hat{x}_m^c\big)+\sum_{\substack{i=1,\\ i\neq m}}^{N}\mathcal{J}_i(x_i)\Big)\cdot\mathbb{P}(\phi_m=1)\cdot \prod_{\substack{i=1,\\ i\neq m}}^{N}\mathbb{P}(\phi_i=\Delta_i) + \Big(\mathcal{J}_i\big(\hat{x}_m^o\big)+\sum_{\substack{i=1,\\ i\neq m}}^{N}\mathcal{J}_i(x_i)\Big)\cdot\mathbb{P}(\phi_m=0)\cdot \prod_{\substack{i=1,\\ i\neq m}}^{N}\mathbb{P}(\phi_i=\Delta_i)\Big]\\
&= \Big(\mathcal{J}_i\big(\hat{x}_m^c\big)\mathbb{P}(\phi_m=1)+\mathcal{J}_i\big(\hat{x}_m^o\big)\mathbb{P}(\phi_m=0)\Big)\cdot \underbrace{\sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\},\\ i\neq m}} \prod_{\substack{i=1,\\ i\neq m}}^{N}\mathbb{P}(\phi_i=\Delta_i)}_{\text{All cases}=1}\\
&\quad + \underbrace{\sum_{\substack{x_i=\{\hat{x}_i^c,\hat{x}_i^o\},\\ \text{for } i=\{1,\dots,N\},\\ i\neq m}} \Big(\sum_{\substack{i=1,\\ i\neq m}}^{N}\mathcal{J}(x_i)\Big)\prod_{\substack{i=1,\\ i\neq m}}^{N}\mathbb{P}(\phi_i=\Delta_i)}_{\text{Expectation of overall control cost for } N-1 \text{ loops}}.
\end{aligned}$$

After calculating the rest of $N-1$ loops, we can get the expectation of overall control cost,

$$\mathbb{E}\big(\mathcal{J}(x)\big) = \sum_{i=1}^{N}\Big(\mathcal{J}_i(\hat{x}_i^c)\mu_{\phi_i}(s) + \mathcal{J}_i(\hat{x}_i^o)\big(1-\mu_{\phi_i}(s)\big)\Big).$$

REFERENCES

[1] Y. Ma, J. Guo, Y. Wang, A. Chakrabarty, H. Ahn, P. Orlik, and C. Lu, "Optimal dynamic scheduling of wireless networked control systems," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 77–86.

[2] ISA100, "Wireless systems for automation." http://www.isa100wci.org, 2018.

[3] F. Group, "WirelessHART specification." https://fieldcommgroup.org/technologies/hart/hart-technology, 2018.

[4] Z. Alliance, "ZigBee." http://www.zigbee.org, Retrieved May 20, 2020.

[5] IEEE, "IEEE 802.15.4 standard." https://standards.ieee.org/findstds/standard/802.15.4-2015.html, Retrieved May 20, 2020.

[6] 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, 2016.

[7] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 978–1013, 2017.

[8] ABB, "WirelessHART networks: 7 myths that cloud their consideration for process control Measurement made easy," Retrieved September 20, 2021.

[9] Emerson™, "White paper: Emerson wireless security – WirelessHart and Wi-Fi (2017)," Retrieved September 20, 2021.

[10] Y.-K. Huang, A.-C. Pang, and H.-N. Hung, "An adaptive gts allocation scheme for ieee 802.15. 4," *IEEE transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 641–651, 2008.

[11] Y. Zhan, Y. Xia, and M. Anwar, "GTS size adaptation algorithm for ieee 802.15. 4 wireless networks," *Ad Hoc Networks*, vol. 37, pp. 486–498, 2016.

[12] A. Faridi, M. R. Palattella, A. Lozano, M. Dohler, G. Boggia, L. A. Grieco, and P. Camarda, "Comprehensive evaluation of the IEEE 802.15. 4 MAC layer performance with retransmissions," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3917–3932, 2010.

[13] D. Gunatilaka, M. Sha, and C. Lu, "Impacts of channel selection on industrial wireless sensor-actuator networks," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[14] M. Sha, D. Gunatilaka, C. Wu, and C. Lu, "Empirical study and enhancements of industrial wireless sensor–actuator network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 696–704, 2017.

[15] F. Dobslaw, T. Zhang, and M. Gidlund, "End-to-end reliability-aware scheduling for wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 758–767, 2016.

[16] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[17] H. Gao and T. Chen, "Network-based h_∞ output tracking control," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 655–667, 2008.

[18] Y. Ma, Y. Wang, S. Di Cairano, T. Koike-Akino, J. Guo, P. Orlik, and C. Lu, "A smart actuation architecture for wireless networked control systems," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1231–1237.

[19] Z. Wang, F. Yang, D. W. Ho, and X. Liu, "Robust h_∞ control for networked systems with random packet losses," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 4, pp. 916–924, 2007.

[20] H. Li, C. Wu, P. Shi, and Y. Gao, "Control of nonlinear networked systems with packet dropouts: interval type-2 fuzzy model-based approach," *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2378–2389, 2015.

[21] J. Wu and T. Chen, "Design of networked control systems with packet dropouts," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1314–1319, 2007.

[22] A. Saifullah, C. Wu, P. B. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen, "Near optimal rate selection for wireless control systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 4s, p. 128, 2014.

[23] B. Demirel, Z. Zou, P. Soldati, and M. Johansson, "Modular design of jointly optimal controllers and forwarding policies for wireless control," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3252–3265, 2014.

[24] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Control-aware random access communication," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2016, pp. 1–9.

[25] D. J. Antunes, W. Heemels, J. P. Hespanha, and C. Silvestre, "Scheduling measurements and controls over networks—part i: Rollout strategies for protocol design," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 2036–2041.

[26] ——, "Scheduling measurements and controls over networks—part ii: Rollout strategies for simultaneous protocol and controller design," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 2042–2047.

[27] M. Lješnjanin, D. E. Quevedo, and D. Nešić, "Packetized mpc with dynamic scheduling constraints and bounded packet dropouts," *Automatica*, vol. 50, no. 3, pp. 784–797, 2014.

[28] 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, p. 3, 2018.

[29] Y. Ma and C. Lu, "Efficient holistic control over industrial wireless sensor-actuator networks," in *2018 IEEE International Conference on Industrial Internet (ICII)*. IEEE, 2018, pp. 89–98.

[30] D. Kim et al., "Sampling rate optimization for ieee 802.11 wireless control systems," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2019.

[31] M. Eisen, M. M. Rashid, A. Ribeiro, and D. Cavalcanti, "Scheduling low latency traffic for wireless control systems in 5g networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[32] E. G. Peters, D. E. Quevedo, and M. Fu, "Controller and scheduler codesign for feedback control over IEEE 802.15. 4 networks," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2016–2030, 2016.

[33] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, "How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime," *IEEE control systems magazine*, vol. 23, no. 3, pp. 16–30, 2003.

[34] E. Eyisi, J. Bai, D. Riley, J. Weng, W. Yan, Y. Xue, X. Koutsoukos, and J. Sztipanovits, "Ncswt: An integrated modeling and simulation tool for networked control systems," *Simulation Modelling Practice and Theory*, vol. 27, pp. 90–111, 2012.

[35] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 16, 2010.

[36] 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, pp. 12 454–12 473, 2015.

[37] B. Li, L. Nie, C. Wu, H. Gonzalez, and C. Lu, "Incorporating emergency alarms in reliable wireless process control," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. IEEE, 2015, pp. 218–227.

[38] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 126–137.

[39] M. Pajic, J. Le Ny, S. Sundaram, G. J. Pappas, and R. Mangharam, "Closing the loop: A simple distributed method for control over wireless networks," in *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2012, pp. 25–36.

[40] 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, pp. 175–184, 2014.

[41] 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 *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*. ACM, 2019, pp. 97–108.

[42] Y. Li, Y. Chen, and I. Podlubny, "Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized mittag–leffler stability," *Computers & Mathematics with Applications*, vol. 59, no. 5, pp. 1810–1821, 2010.

[43] A. Hernandez, "Modification of the IEEE 802.15. 4 implementation extended GTS implementation," *KTH, Tech. Rep.,*, 2011.

[44] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "Wirelesshart: Applying wireless technology in real-time industrial process control," in *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 2008, pp. 377–386.

[45] B. Li, Y. Ma, T. Westenbroek, C. Wu, H. Gonzalez, and C. Lu, "Wireless routing and control: a cyber-physical case study," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2016, pp. 1–10.

[46] X. Liu and A. Goldsmith, "Kalman filtering with partial observation losses," in *2004 43rd IEEE Conference on Decision and Control (CDC)*, vol. 4. IEEE, 2004, pp. 4180–4186.

[47] Y. Shi and H. Fang, "Kalman filter-based identification for systems with randomly missing measurements in a network environment," *International Journal of Control*, vol. 83, no. 3, pp. 538–551, 2010.

[48] M. Eisen, M. M. Rashid, K. Gatsis, D. Cavalcanti, N. Himayat, and A. Ribeiro, "Control aware radio resource allocation in low latency wireless control systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7878–7890, 2019.

[49] T. Chen and B. A. Francis, *Optimal sampled-data control systems*. Springer Science & Business Media, 2012.

[50] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, p. 34, 2012.

[51] J. W. Taylor, "Exponential smoothing with a damped multiplicative trend," *International journal of Forecasting*, vol. 19, no. 4, pp. 715–725, 2003.

[52] J. Guo and P. Orlik, "Self-transmission control in IoT over heterogeneous wireless networks," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 898–903.

[53] M. Köppe, "On the complexity of nonlinear mixed-integer optimization," in *Mixed Integer Nonlinear Programming*. Springer, 2012, pp. 533–557.

[54] K. M. Anstreicher, "Linear programming in $O([n^3/ln(n)]L)$ operations," *SIAM J. on Optimization*, vol. 9, no. 4, pp. 803–812, 1999.

[55] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[56] 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 *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 97–108.

[57] L. Leemis, "The product of n mutually independent bernoulli random variables is bernoulli," http://www.math.wm.edu/~leemis/chart/UDR/PDFs/BernoulliP.pdf, Retrieved December 27, 2020.

[58] H. Lee, A. Cerpa, and P. Levis, "Improving wireless simulation through noise modeling," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 21–30.

[59] H. Deng, M. Krstic, and R. J. Williams, "Stabilization of stochastic nonlinear systems driven by noise of unknown covariance," *IEEE Transactions on Automatic Control*, vol. 46, no. 8, pp. 1237–1253, 2001.

[60] E. Toscano and L. L. Bello, "Multichannel superframe scheduling for IEEE 802.15. 4 industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 337–350, 2012.

[61] Simulink, "Simulink Desktop Real-Time," https://www.mathworks.com/products/simulink-desktop-real-time.html, Retrieved July. 6, 2020.

[62] S. Duquennoy, A. Elsts, B. Al Nahas, and G. Oikonomo, "Tsch and 6tisch for contiki: Challenges, design and evaluation," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2017, pp. 11–18.

[63] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled tsch," in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 337–350.

[64] 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, pp. 1–32, 2019.