Optimal Dynamic Scheduling of Wireless Networked Control Systems

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

TR2019-022 May 23, 2019

Abstract

Wireless networked control system is gaining momentum in industrial cyber-physical systems, e.g., smart factory. Suffering from limited bandwidth and nondeterministic link quality, a critical challenge in its deployment is how to optimize the closed-loop control system performance as well as maintain stability. In order to bridge the gap between network design and control system performance, we propose an optimal dynamic 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. The optimal dynamic scheduling strategy boils down to solving a nonlinear integer programming problem, which is further relaxed to a linear programming problem. The proposed strategy provably renders the closed-loop system meansquare 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. Simulation results show that the optimal dynamic scheduling can enhance control system performance and adapt to both constant and variable network background noises as well as physical disturbance.

ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)

Optimal Dynamic Scheduling of Wireless Networked Control Systems

Yehan Ma
Washington University in St. Louis
yehan.ma@wustl.edu

Jianlin Guo, Yebin Wang, Ankush Chakrabarty, Heejin Ahn, Philip Orlik
Mitsubishi Electric Research Labs
{guo, yebinwang, chakrabarty, hahn, porlik}@merl.com

Chenyang Lu
Washington University in St. Louis
lu@wustl.edu

ABSTRACT

Wireless networked control system is gaining momentum in industrial cyber-physical systems, e.g., smart factory. Suffering from limited bandwidth and nondeterministic link quality, a critical challenge in its deployment is how to optimize the closed-loop control system performance as well as maintain stability. In order to bridge the gap between network design and control system performance, we propose an optimal dynamic 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. The optimal dynamic scheduling strategy boils down to solving a nonlinear integer programming problem, which is further relaxed to a linear programming problem. The proposed strategy provably 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. Simulation results show that the optimal dynamic scheduling can enhance control system performance and adapt to both constant and variable network background noises as well as physical disturbance.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; *Sensors and actuators*; • **Networks** → *Cross-layer protocols*;

KEYWORDS

Cyber-physical system, wireless network, multi-loop control system, dynamic scheduling, optimization, link quality

1 INTRODUCTION

Wireless technology is gaining rapid adoption in industry automation for lowering deployment and maintenance costs in challenging industrial environments. Industrial standard organizations such as ISA100 [20], WirelessHART [14], and ZigBee [1], which are all based on the IEEE 802.15.4 physical layer [19], are strong proponents of wireless network for industrial automation. However, in wireless networked control systems (WNCSs), the primary use of wireless network is in monitoring. The status quo is that, it remains challenging to close the loop at the control-to-actuation side over wireless network due to multiple reasons.

First, wired networks, such as Ethernet, use twisted pairs and fiber optic links, resulting in high data rate of up to hundreds of Gbit/s. In contrast, wireless networks, especially low-power and low-cost industrial wireless networks, have limited throughput. For instance, IEEE 802.15.4 physical layer supports data rate of up to 250 kbit/s. The control performance of WNCSs largely depends on how much network resource they are able to obtain. Second, the physical isolation of wired networks ensure supreme link quality and resiliency to external environment changes. However, link qualities of wireless networks are prone to environmental factors such as obstacles, noises, interferences, extreme weather, as well as human interference in the form of cyber attacks. Poor link quality can cause significant data packet loss, resulting in degradation of the control performance. Finally, most wireless network designs focus on network performances, overlooking control performances which directly determine the profits and the safety of a factory. Therefore, a practical wireless network design for WNCSs must target the improvement of the control performance by taking limited network resource allocation and the impact of link quality 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 resource allocation on the physical control. We design an optimal dynamic scheduling strategy to optimize the control performance by allocating more network resources to needy loops and reducing the effects of network on physical control system, based on run-time predictions of link quality and physical control performance.

Our major contributions in this paper include:

(1) incorporate link quality prediction of wireless network;
(2) provide a tractable method for optimal network scheduling based on predictions of both link quality and the control performance;
(3) establish stability guarantees for the closed-loop system with optimal scheduling;
(4) illustrate the efficacy of our strategy on the high-fidelity TOSSIM simulation environment in spite of constant and variable background noises, and physical disturbance.

The rest of the paper is organized as follows. Sec. 2 reviews related works. Sec. 3 overviews WNCS. Sec. 4 describes wireless link quality prediction and its simulation results. Sec. 5 formulates the optimal scheduling problem and its linear programming relaxation. Sec. 6 details the stability analysis and condition of the proposed optimal scheduling method. Sec. 7 evaluates the simulation results. We present our conclusions in Sec. 8.

This work was done while Y. Ma was an intern with Mitsubishi Electric Research Labs.

2 RELATED WORK

The past decade has witnessed sustained interest in exploring WNCSs and expanding their applications over industry automation [26, 30], in the views of network design, control system design, and more recently, network and control co-design.

From a network design perspective, several approaches are presented to address resource allocation. For example, Huang et al. [18] 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. [44] allocate network resource 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 [11], channel selection [15], routing [37], and reachability-aware scheduling [9], etc. However, few are targeting optimizing control performance.

On the control system 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. [39] discuss Kalman filtering with intermittent measurement; Gao et al. [12] investigate robust output tracking control subject to time delay between controllers and actuators; Ma et al. [33] explore the design freedom of system architectures and propose a smart actuation architecture; Wang et al. [25, 42, 43] 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 resource 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 resource allocation designs tailored for control performance of WNCSs. Saifullah et al. determine [7, 36] sampling rates to optimize control performance. Gatsis et al. [13] propose distributed control-aware random network access policies for each sensor so that all control loops are stabilizable. Lješnjanin et al. [29] allocate network resource by finding optimal node, which minimize cost function of model predictive control (MPC), in every network time instant. Ma et al. [31, 32] propose the concept of holistic control that cojoins network reconfiguration and physical control over multi-hop mesh network. However, [13, 36] assume perfect link quality, and none of [7, 29, 31] models the effects of link quality on control performance. Peters et al. [35] present co-design of scheduler and controller by deriving optimal control as well as determining transmitting control commands in contention access period (CAP) or contention free period (CFP), or no transmission at all, targeting IEEE 802.15.4 MAC. 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 [11].

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

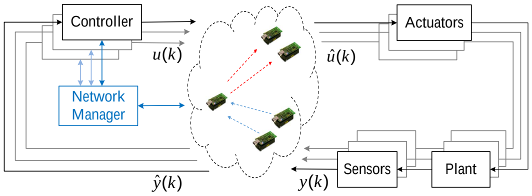

Figure 1: Architecture of WNCS (red and blue dashed arrows indicate actuation and sensing flows, respectively)

3 OVERVIEW OF WNCS

Fig. 1 shows the architecture of the multi-loop WNCS. The controllers are typically located far 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.

3.1 Physical plant and controller

We consider N control loops that share the same wireless network. Each control loop is associated with an individual plant. For the ith loop, 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 time index, $i \in \{1, 2, \ldots, N\}$ is the loop 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 wireless 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 change of variables [27].

At time k, a sensor sends measurements $y_i(k)$ to a controller over the wireless network. At the controller side, a state observer [39] estimates the states of the plant. Based on the estimated state $\hat{x}_i(k)$, the controller generates the control 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, $\hat{u}_i(k-1)$.

3.2 Wireless network

3.2.1 Wireless sensor-actuator network (WSAN). Using IEEE 802.15.4-based network, we schedule sensing and actuation flows of the control loops. A superframe is a collection of timeslots repeating in time. For IEEE 802.15.4-based network, in beacon enabled mode, the superframe is bounded by 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 attached devices, to identify the network, and to describe the structure of the superframes. During the inactive period, the coordinator and end nodes are able to enter a low-power mode, such as 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 the MAC scheme of carrier sense multiple access/ collision avoidance (CSMA/CA). For applications with real-time performance requirements, the network manager (NM) dedicates guaranteed time slots (GTSs) during CFP. As specified by IEEE 802.15.4 MAC protocol [19], the NM can allocate up to 7 slots in CFP. The limitations of IEEE 802.15.4 MAC protocol was discussed and modified by [3, 17], such that the number of slots assigned to CAP and CFP becomes a free design parameter. WirelessHART and ISA100 also support customized number of slots in CFP. In this paper, we target WNCSs with real-time performance requirements, thus focus on the scheduling of the CFP, whereas CAP can be reserved for other uses.

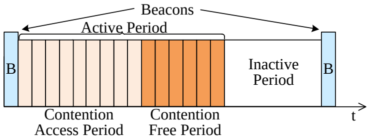

Figure 2: Structure of the IEEE 802.15.4 superframe

3.2.2 Network manager with better authority. The NM manages the network and its devices. In most network architectures, the NM, the application controller, and the coordinator of WSAN are co-located. Therefore, the NM communicates with controllers and the coordinator via a reliable wired network with ignorable packet drop and latency.

We propose a NM that utilizes the information of predicted link quality and the knowledge of predicted control performance from the controller to obtain optimal scheduling. As a result, the NM dynamically schedules the data flows of the WNCS based on its knowledge of both the wireless network and the physical plants at run-time. Then the NM notifies the coordinator of the updated schedule, and the coordinator broadcasts the updated schedule in the beacon at the beginning of the next superframe. In this way, field nodes that receive the beacon update their schedules accordingly.

REMARK 3.1. *In terms of a multi-loop WNCS, the NM allocates the network resource 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 assigned to be 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 skipped or triggered by the NM. Therefore, the network resource allocation of control loops is a special kind of event-triggered control tailored for a multi-loop WNCS.* □

3.3 Recover from beacon packet loss

In a star network, for upstream (sensing) flows, if a beacon message is received by an upstream node, the node wakes up and sends sensing flow at the assigned time slots indicated by the beacon. However, if the beacon message, which contains the updated schedule generated by the NM, is lost, dynamic scheduling may cause collisions between flows. For instance, if a sensor fails to receive the updated schedule, it will not be able to update its newly assigned time slots, and will keep transmitting sensing flows to the controller at previous assigned time slots, which may be assigned to other flows according to the updated schedule. Therefore, for simplicity, we propose to reserve fixed time slots for sensing flows and only dynamically schedule actuation flows.

For downstream (actuation) flows, if a beacon message is received by a downstream node, the node shall wake up and listen at the assigned time slots indicated by the beacon. We propose a packet loss recovery strategy to improve the resiliency of beacon packet loss. If no beacon packet has been received by a node, the node wakes up and keeps listening for the whole superframe. This strategy results in longer listening time and higher energy expenditure of wireless nodes if and only if the beacon message is lost. Besides, the longer listening time will not cause any collision.

4 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 [4]. The 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 [4].

4.1 Link quality prediction

Holt's additive trend prediction method [16, 40] is employed to predict PRR of next m transmissions,

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

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 steps ahead, $\widehat{PRR}(k+m|k)$ is the predicted PRR m transmissions ahead, α and γ $(0 < \alpha, \gamma < 1)$ are the level and slope smoothing parameter, respectively.

4.2 Results of link quality prediction

In our study, wireless traces from 4 links of the WSAN testbed at Washington University [37] have been collected, which contain the connectivity and RSSI data [24]. 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 [22, 23]. Fig. 3 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.

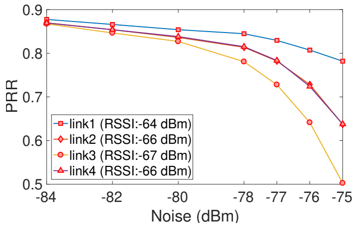

Figure 3: PRRs under various noise levels

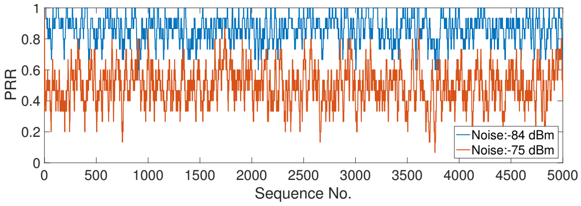

Figure 4: Sliding-window PRRs of link 3

Fig. 4 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. 5. 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. 6. 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.

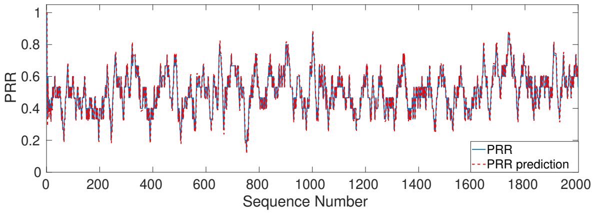

Figure 5: 1-step PRR prediction under noise −75dBm (link 3)

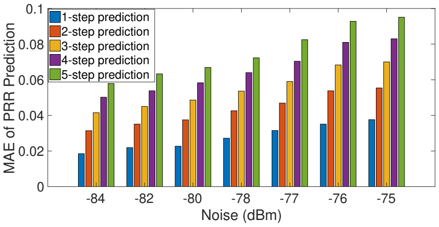

Figure 6: PRR prediction errors of link 3 under various noise

5 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. We formulate the optimal scheduling strategy as a nonlinear integer program, which is relaxed into a linear programming (LP) problem. Finally, we present a heuristic algorithm of sorting control loops by the descending order of their costs in each superframe for shortening the latency of needy loops.

5.1 Multi-loop control system modeling

5.1.1 Simplifications and assumptions. We use s to represent the schedule of next superframe. The number of transmission (η) is at the center of the tradeoff between reliability and network resources, i.e., more transmissions lead to a higher packet delivery ratio (PDR) at a cost of network resources [31]. Denote η_i the number of transmission of *loop i* in schedule s. 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 [28, 38, 39], and the WNCSs are more sensitive to packet loss on the actuation side of the wireless network [23]. We refer readers who are interested in sensing packet scheduling problem to [10].

In Secs. 5.2, 5.3, and 6, 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. For ease of analysis, we assume strict periodicity of actuation packets. This restriction is lifted in our simulation to allow realistic packet timing.

To simplify the problem, we assume all loops have the same sampling period. A potential method for relaxing this assumption is to use sampled-data control techniques discussed in, for example, [6], i.e., rewriting systems with different periods in the slowest time frame (least common multiple of all sampling periods).

5.1.2 Packet delivery modeling. Let a binary variables $\phi_i(k)$ denote end-to-end packet reception $\big(\phi_i(k) = 1\big)$ or loss $\big(\phi_i(k) = 0\big)$. 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. Given link failure ratio of *link i* (*loop i*) as $\beta_i = 1 - PRR_i$, we have PDR

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

Here, PDR is a function of link quality and schedule.

5.2 Optimal scheduling formulation

At time k, controller determines control $u(k)$ based on state $x(k)$ and system model (1). Network manager ought to come up with a schedule $s(k)$ based on $x(k), u(k)$, PRR, and system model (1). 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 $x_i(k), u_i(k)$, and ϕ_i as follows

(1) packet of *loop i* at $t = k$ arrives (closed loop):

$$\hat{u}_i^c(k) = u_i(k),\ x_i(k+1) = \hat{x}_i^c(k+1) = f_i\big(x_i(k), \hat{u}_i^c(k)\big), \tag{4}$$

(2) packet $u_i(k)$ is lost, and $\hat{u}_i(k-1)$ is actuated (open loop):

$$\hat{u}_i^o(k) = \hat{u}_i(k-1),\ x_i(k+1) = \hat{x}_i^o(k+1) = f_i\big(x_i(k), \hat{u}_i^o(k)\big), \tag{5}$$

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 $W_i > 0$ 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 $x(k) = \left[x_1(k) \;\; x_2(k) \;\; \dots \;\; x_N(k)\right]^T$, and $W = blkdiag(W_1, W_2, ..., W_N)$. *blkdiag* is the block-diagonalize operator that constructs a diagonal matrix from input matrices. We will see in Sec. 6 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 $\mathcal{J}_i\left(x_i(k+1)\right)$ is:

$$\mathbb{E}\Big(\mathcal{J}_i\left(x_i(k+1)\right)\Big) = \mu_{\phi_i}(s)\mathcal{J}_i\left(\hat{x}_i^c(k+1)\right) + \left(1-\mu_{\phi_i}(s)\right)\mathcal{J}_i\left(\hat{x}_i^o(k+1)\right), \tag{8}$$

where $\mathbb{E}$ is the expectation operator. Substituting (3) into (8) gives

$$\mathbb{E}\Big(\mathcal{J}_i\left(x_i(k+1)\right)\Big) = \mathcal{J}_i\left(\hat{x}_i^c(k+1)\right) + \Big(\mathcal{J}_i\left(\hat{x}_i^o(k+1)\right) - \mathcal{J}_i\left(\hat{x}_i^c(k+1)\right)\Big)\beta_i^{\eta_i}. \tag{9}$$

The optimal scheduling problem is formulated as:

$$\underset{\eta_i}{\text{minimize}} \quad \mathbb{E}\Big(\mathcal{J}\left(x(k+1)\right)\Big) \tag{10a}$$

$$\text{subject to} \quad \sum_{i=1}^{N} \eta_i \leq L \tag{10b}$$

$$\eta_i \in \{0, 1, \dots, L\}, \forall i \in \{1, 2, ..., N\}, \tag{10c}$$

where L is the total number of slots assigned for all actuation flows in each superframe. The constraint (10b) indicates the requirement of schedulability. The constraint (10c) means that the transmission number should be a non-negative integer. Problem (10) is an integer programming problem. Furthermore, the objective function is nonlinear in η as can be seen from (9). It is well-known that this class of problems is NP-hard [21].

5.3 Run-time optimal scheduling

Since we are targeting a scheduling problem that must be solved for every superframe, its tractability is of vital importance.

5.3.1 *Binary linear programming.*

We propose a transformation of variables to recast Problem (10) into a binary linear programming (BLP) problem. The resultant BLP problem is equivalent to Problem (10) 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 $\{0, 1\}^{N(L+1)}$:

$$\tilde{T}_{ij} = \begin{cases} 1, \; \eta_i = j, \; j \in \{0, 1, \dots, L\} \\ 0, \; \text{otherwise.} \end{cases} \tag{11}$$

We can represent $\mathbb{E}\Big(\mathcal{J}_i\left(x_i(k+1)\right)\Big)$ in (9) as

$$\mathbb{E}\Big(\mathcal{J}_i\left(x_i(k+1)\right)\Big) = \sum_{j=0}^{L} \underbrace{\Big(\mathcal{J}_i\left(\hat{x}_i^c(k+1)\right) + \Big(\mathcal{J}_i\left(\hat{x}_i^o(k+1)\right) - \mathcal{J}_i\left(\hat{x}_i^c(k+1)\right)\Big)\beta_i^j\Big)}_{q_{ij}} \tilde{T}_{ij}. \tag{12}$$

According to the linearity of mathematical expectation, the expectation of the overall cost function is equal to the sum of the expectations of the cost function of each loop, that is

$$\mathbb{E}\Big(\mathcal{J}\left(x(k+1)\right)\Big) = \sum_{i=1}^{N} \mathbb{E}\Big(\mathcal{J}_i\left(x_i(k)\right)\Big). \tag{13}$$

By defining $\tilde{T} = \left[\tilde{T}_{10} \;\; \tilde{T}_{11} \;\; \dots \;\; \tilde{T}_{1L} \;\; \tilde{T}_{20} \;\; \tilde{T}_{21} \;\; \dots \;\; \tilde{T}_{2L} \;\; \dots \;\; \tilde{T}_{NL}\right]^T$, we can see that the objective function is a linear function of $\tilde{T}$,

$$\mathbb{E}\Big(\mathcal{J}\left(x(k+1)\right)\Big) = Q\tilde{T}, \tag{14}$$

where $Q = \Big[\left[q_{10} \;\; q_{11} \;\; \dots \;\; q_{1L}\right], ..., \left[q_{N0} \;\; \dots \;\; q_{NL}\right]\Big]$. Problem (10) is reduced to a binary linear programming problem as follows

$$\underset{\tilde{T}_{ij}}{\text{minimize}} \quad Q\tilde{T} \tag{15a}$$

$$\text{subject to} \quad \sum_{i=1}^{N}\sum_{j=0}^{L} j\tilde{T}_{ij} \leq L \tag{15b}$$

$$\sum_{j=0}^{L} \tilde{T}_{ij} = 1 \tag{15c}$$

$$\tilde{T}_{ij} \in \{0, 1\}, \forall i \in \{1, 2, ..., N\}, \forall j \in \{0, 1, 2, ..., L\} \tag{15d}$$

Note that we rewrite the constraint (10b) as (15b). In order to ensure each loop i has unique η_i, we impose constraints (15c)-(15d). The transmission numbers can be recovered from $\tilde{T}$ using

$$\eta_i = \left[0 \;\; 1 \;\; 2 \;\; \dots \;\; L\right]\left[\tilde{T}_{i0} \;\; \tilde{T}_{i1} \;\; \tilde{T}_{i2} \;\; \dots \;\; \tilde{T}_{iL}\right]^T. \tag{16}$$

There are many integer linear programming solvers such as Gurobi, CPLEX, and MATLAB.

5.3.2 *Linear programming relaxation.*

By relaxing the binary constraint (15d) 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 (16). The complexity of LP is $O(\frac{m^3}{ln(m)}D)$ [2], where m is the space dimension, 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). As shown in Fig. 7, the advantage of LP relaxation in computational complexity appears when N increases.

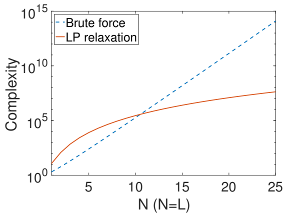

Figure 7: Complexity of optimal scheduling problem

Remark 5.1. *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 [31], we propose a heuristic method to achieve a feasible solution by iteratively reducing the largest element $\max\limits_{1 \leq i \leq N} \eta_i$ by one, until $\sum_{i=1}^{N} \eta_i \leq L$.* □

5.4 Heuristics of sorting loops in superframes

In previous sections, we assume that we schedule the actuation packet of each loop in the ascending order of the loop number. In this section, we provide an algorithm of determining the order of loops in each superframe given the solution η_i of optimal scheduling problem (15). 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.

Algorithm 1: Algorithm of sorting loops in each superframe

```
input  : Transmission numbers returned by optimal scheduling: $\eta_i$,
         predicted costs: $Cost_i = w_i x_i(k)$, $w_i$, $i \in \{1, 2, \ldots, N\}$
         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
    $\eta_i$_left ← $\eta_i$;
    //$\eta_i$_left represents unscheduled transmissions;
Cost_matrix ← $\begin{bmatrix} 1 & 2 & 3 & \ldots & N \\ Cost_1 & Cost_2 & & \ldots & Cost_N \end{bmatrix}$;
Sorted_loop_number ← sort loops (first row of Cost_matrix) in
  descending order of $Cost_i$ (second row of Cost_matrix);
while slot≤ $L$ and $\sum_{i=1}^{N} \eta_i > 0$ do
    for $i$ in Sorted_loop_number do
        if $\eta_i$_left > 0 then
            Schedule(Slot) ← $i$; Slot ← Slot + 1;
            $\eta_i$_left ← $\eta_i$_left − 1;
return Schedule;
```

6 STABILITY ANALYSIS

The aforementioned optimal scheduling strategy can improve the control performance of the multi-loop WNCS without loss of stability. In this section, we provide a condition of stability in the mean-square sense. According to [5], 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,

$$\mathbb{E}\Big(V\big(x(k+1)\big)\Big) - \mathbb{E}\Big(V\big(x(k)\big)\Big) \leq 0. \tag{17}$$

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.

6.1 MSS check of LTI system with fixed schedule

Consider a multi-loop DT-LTI system, where system dynamics of the loop i are given by

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k),\ u_i(k) = K_i x_i(k), \tag{18}$$

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.

To apply the stability analysis in [34], we model the closed-loop system dynamics over actuation networks with schedule s as a discrete-time stochastic system. According to [34], the closed-loop system dynamics of loop i are equivalent to the following augmented system

$$z_i(k+1) = \tilde{A}_{si}(s,k) z_i(k), \tag{19}$$

where

$$\tilde{A}_{si}(s,k) = \begin{bmatrix} A_i & B_i & 0 \\ 0 & 1-\phi_i(s) & \phi_i(s) \\ K_i A_i & 0 & K_i B_i \end{bmatrix}, z_i(k) = \begin{bmatrix} x_i(k) \\ \hat{u}_i(k) \\ u_i(k) \end{bmatrix}. \tag{20}$$

Similar to (4) and (5), $z_i(k+1)$ can be determined as follows

(1) packet at $t = k$ arrives ($\phi_i(k) = 1$):

$$\hat{u}_i(k) = u_i(k),\ z_i(k+1) = \hat{z}_i^c(k+1) = \tilde{A}_{si}^c z_i(k), \tag{21}$$

(2) packet at $t = k$ is lost, and $\hat{u}_i(k-1)$ is adopted ($\phi_i(k) = 0$):

$$\hat{u}_i(k) = \hat{u}_i(k-1),\ z_i(k+1) = \hat{z}_i^o(k+1) = \tilde{A}_{si}^o z_i(k), \tag{22}$$

where

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

Analogously, the multi-loop control system can be rewritten as

$$z(k+1) = \tilde{A}(s,k) z(k) \tag{23}$$

where $\tilde{A}(s,k) = blkdiag\big(\tilde{A}_{s1}(s,k), \tilde{A}_{s2}(s,k), \ldots, \tilde{A}_{sN}(s,k)\big)$, $z(k) = \begin{bmatrix} z_1(k) & z_2(k) & \ldots & z_N(k) \end{bmatrix}^T$. In order to prove stability properties of the closed-loop system, besides assumptions in Sec. 5.1.1, we make the following assumption.

Assumption 6.1. *Sequences* $\{\phi_i(k), k \in \mathbb{N}\}, \forall i \in \{1, 2, ..., N\}$, *are i.i.d.*

Note that this assumption is lifted in evaluation section to allow much more realistic radio propagation and noise models in TOSSIM [22]. Under Assumption 6.1, we can rewrite $\tilde{A}(s,k)$ in (23) as

$$\tilde{A}(s,k) = \tilde{A}_0 + \sum_{i=1}^{N} \tilde{A}_i p_i(k), \tag{24}$$

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 $\mathbb{E}\big(p_i(k)p_j(k)\big) = 0, \forall i,j \in \{1, 2, \ldots, N\}$, $\tilde{A}_0 = blkdiag(\tilde{A}_{01}, \tilde{A}_{02}, \ldots, \tilde{A}_{0N})$, $\tilde{A}_1 = blkdiag(A_{\phi_1}, \mathbf{0}, \ldots, \mathbf{0})$, $\tilde{A}_2 = blkdiag(\mathbf{0}, A_{\phi_2}, \mathbf{0}, \ldots, \mathbf{0})$, ..., $\tilde{A}_N = blkdiag(\mathbf{0}, \mathbf{0}, \ldots, A_{\phi_N})$,

$$\tilde{A}_{0i} = \begin{bmatrix} A_i & B_i & 0 \\ 0 & (1-\mu_{\phi_i}(s))I & \mu_{\phi_i}(s)I \\ K_i A_i & 0 & K_i B_i \end{bmatrix},$$

$$A_{\phi_i} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu_{\phi_i}(s)I & -\mu_{\phi_i}(s)I \\ 0 & 0 & 0 \end{bmatrix}, \ \sigma^2_{p_i}(s) = \frac{1}{\mu_{\phi_i}(s)} - 1.$$

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 6.1, we have that $p_i(k)$ is i.i.d with $\mathbb{E}\big(p_i(k)\big) = 0$ and $\mathrm{Var}\big(p_i(k)\big) = \sigma^2_{p_i}$.

For the discrete-time stochastic system (24), the following lemma gives a condition to check whether the system is MSS.

Lemma 6.1. [5](p131) *The system* (19) *is MSS if and only if there exists a positive definite matrix P satisfying*

$$\tilde{A}_0^T P \tilde{A}_0 - P + \sum_{i=1}^{N} \sigma^2_{p_i} \tilde{A}_i^T P \tilde{A}_i < 0. \tag{25}$$

Remark 6.2. *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 6.1. We have $P = blkdiag(P_1, P_2, ..., P_N)$.* □

6.2 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 6.3. *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* (10) *is also MSS.*

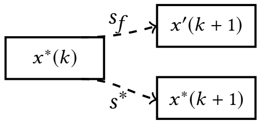

Figure 8: Diagram of stability proof

Proof. As shown in Fig. 8, 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 (17). Therefore,

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

Because the schedule s^* minimizes the objective function $\mathbb{E}\Big(\mathcal{J}\big(x(k+1)\big)\Big)$ in the optimization problem (10), we have

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

Combining (26) and (27), we derive

$$\mathbb{E}\Big(\mathcal{J}\big(x^*(k+1)\big)\Big) \leq \mathbb{E}\Big(\mathcal{J}\big(x^*(k)\big)\Big). \tag{28}$$

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 6.4. *For DT-LTI system* (23)*, for $P > 0$ satisfying Lemma 6.1 with s_f, we can interpret the function $\mathcal{J}(x) = x^T P x$ as a stochastic Lyapunov function with s_f* ([5] p132)*, and thus $\mathcal{J}(x)$ is also a Lyapunov function of the optimally scheduled system. This is why we choose a quadratic objective function in* (7)*.* □

Remark 6.5. *Although we set $\mathcal{J}(x)$ as a quadratic function to analyze MSS for DT-LTI systems, Proposition 6.3 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 [8], 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* (10)*.* □

7 EVALUATION

This section shows a systematic case study of the proposed scheduling strategy. On the physical plant side, we use four 3-state nonlinear double water-tank systems that share the same wireless network. On the network side, we collect IEEE.802.15.4 traces using TOSSIM, and then empirically evaluate our strategy under constant and variable network background noise levels as well as pulse physical disturbance.

7.1 Simulation settings

7.1.1 Physical control system. Consider four independent 3-state nonlinear double water-tank systems, each of which is modeled as follows [3, 24]:

$$\begin{aligned} \dot{L}_1 &= \frac{1}{\rho A_1}(\alpha u - \frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1}) \\ \dot{L}_2 &= \frac{1}{\rho A_2}(\frac{\sqrt{\rho g}}{\rho R_1}\sqrt{L_1} - \frac{\sqrt{\rho g}}{\rho R_2}\sqrt{L_2}) \\ \dot{L}_R &= \frac{1}{\rho A_R}(\frac{\sqrt{\rho g}}{\rho R_2}\sqrt{L_2} - \alpha u) \end{aligned} \tag{29}$$

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 (29) using the Euler method with sampling period of Δt, and have the discrete-time model

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

There are two types of plants, denoted by PLANT1 and PLANT2, that have different system parameters, shown in Table. 1. Systems 1 and 3 are PLANT1, and systems 2 and 4 are PLANT2.

Table 1: 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

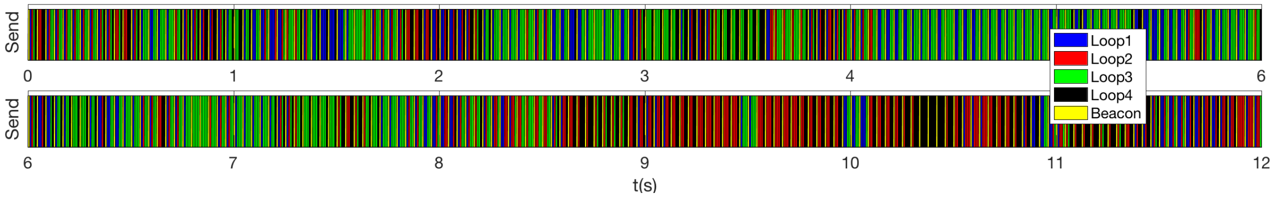

Figure 9: Optimal scheduling under constant noise −76 dBm (the upper plot is 0 − 6 s and lower plot is 6 − 12 s)

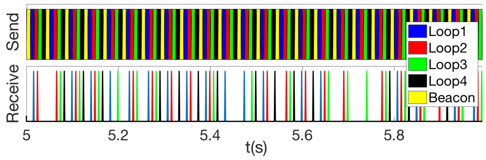

Figure 10: Periodic scheduling under noise −76 dBm

For the four systems, we design state feedback controllers that enable reference tracking. To evaluate the tracking performance, we choose the mean absolute error (MAE) metric:

$$MAE = \frac{1}{n+1}\sum_{k=0}^{n} |x(k) - x_{ref}(k)|, \tag{30}$$

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

7.1.2 Wireless network. We simulate the IEEE 802.15.4 beacon-enabled wireless network. Since we propose to use fixed scheduling for sensing flows in Sec. 5.1.1, 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. The first slot is assigned for a beacon message. The following four CFP slots are assigned for actuation flows of the 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. 5.3.2 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. 4.2, we get packet loss traces using the 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. [41]

7.2 Simulation results

We first 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 noises to show its adaptability and optimality, comparing with the periodic scheduling mechanism. In addition, we also evaluate the performance of our strategy for pulse physical disturbance.

7.2.1 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. 10, 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. 9, and the ratios of slot allocation for each control loop in different time intervals are shown in Fig. 11. Since the sizes of tanks of PLANT1 are smaller than those of PLANT2 as shown in Table.1, 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. 3. Fig. 12 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.

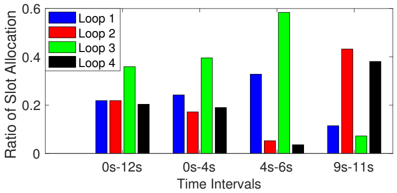

Figure 11: Slot allocation in various time intervals

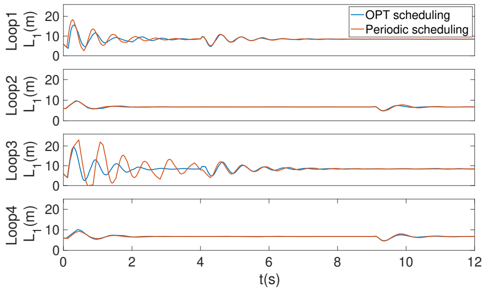

Figure 12: Response curves under noise level of −76 dBm

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. 9 and 11, 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%). This result shows that our OPT scheduling can adjust to physical disturbance.

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. 14 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

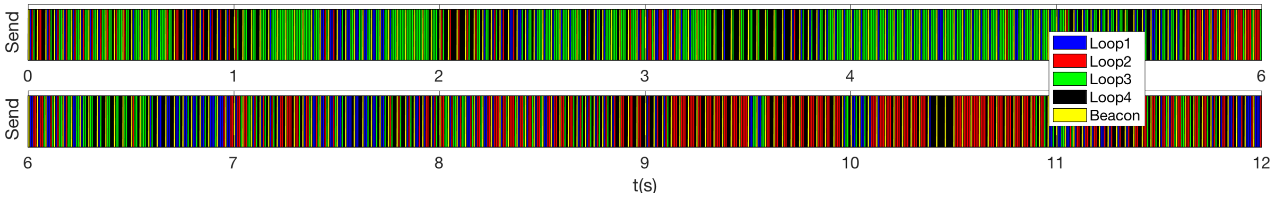

Figure 13: Optimal scheduling under variable noise level

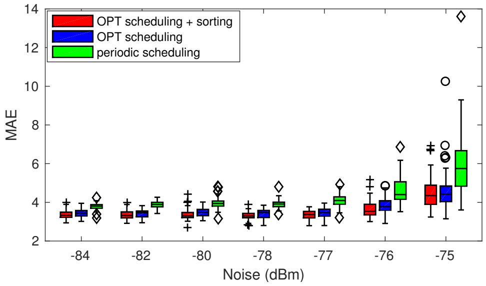

Figure 14: MAE under constant background noise levels

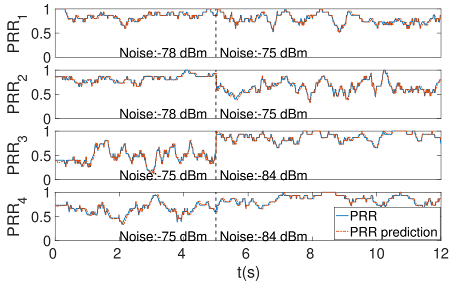

Figure 15: Run-time link quality variation

outperforms the periodic scheduling for all background noise levels. The advantage of the OPT scheduling becomes more apparent as the link 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 can further improve the control performance by considering the latency.

7.2.2 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. 15. 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. 3. The background noise changes at $t = 5$ s. The noise strengths of links 1 and 2 increase to −75 dBm, and that of links 3 and 4 decrease to −84 dBm. The PRR of link 2 becomes the worst in this case.

Under the noise levels shown in Fig. 15, the OPT schedule is shown in Fig. 13, and the ratios of slot allocation are shown in Fig. 16. 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 are shown in Fig. 17. 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.

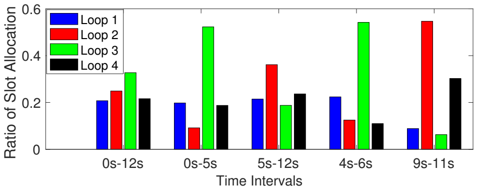

Figure 16: Slot allocation under variable noise level

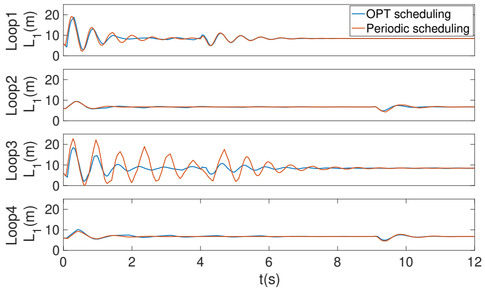

Figure 17: Response curves under variable noise level

Statistical results of control performance under variable noise levels are shown in Fig. 18. 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 runtime. 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 resource 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 resource. 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

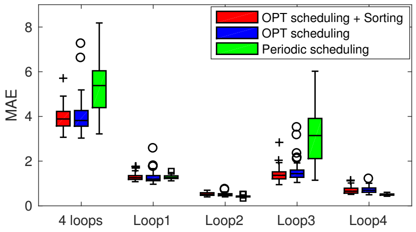

Figure 18: MAE under variable noise level

scheduling can balance the network resource allocation according to link quality and control performance among multiple loops.

8 CONCLUSIONS

In order to bridge the gap between wireless network design and physical control system performance, we propose an optimal dynamic scheduling strategy that optimizes control performance of multi-loop systems by allocating limited network resources based on predictions of both link quality and control performance at runtime. We formulate our optimal scheduling problem as a nonlinear integer programming problem, and then relax it to a linear programming problem for computational efficiency. Also, we provide a stability condition for the wireless networked control system that adopts the optimal scheduling. A systematic evaluation is performed based on four nonlinear double water-tank systems over a realistic IEEE 802.15.4 wireless network. Simulation results show that our optimal scheduling has significantly enhanced the adaptability of the system under both constant and variable background noise as well as physical disturbance.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Arvind Raghunathan at MERL for helpful discussions.

REFERENCES

[1] Z. Alliance. ZigBee. http://www.zigbee.org, 2018.

[2] K. M. Anstreicher. Linear programming in $O([n^3/ln(n)]L)$ operations. *SIAM J. on Optimization*, 9(4):803–812, 1999.

[3] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson. System architectures, protocols and algorithms for aperiodic wireless control systems. *IEEE Trans. Ind. Informat*, 10(1):175–184, 2014.

[4] 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 Trans. Sensor Networks*, 8(4):34, 2012.

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

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

[7] B. Demirel, Z. Zou, P. Soldati, and M. Johansson. Modular design of jointly optimal controllers and forwarding policies for wireless control. *IEEE Trans. Autom. Control*, 59(12):3252–3265, 2014.

[8] H. Deng, M. Krstic, and R. J. Williams. Stabilization of stochastic nonlinear systems driven by noise of unknown covariance. *IEEE Trans. Autom. Control*, 46(8):1237–1253, 2001.

[9] F. Dobslaw, T. Zhang, and M. Gidlund. End-to-end reliability-aware scheduling for wireless sensor networks. *IEEE Trans. Ind. Informat*, 12(2):758–767, 2016.

[10] 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. *arXiv*, 2018.

[11] 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 Trans. Veh. Technol*, 59(8):3917–3932, 2010.

[12] H. Gao and T. Chen. Network-based h_∞ output tracking control. *IEEE Trans. Autom. Control*, 53(3):655–667, 2008.

[13] K. Gatsis, A. Ribeiro, and G. J. Pappas. Control-aware random access communication. In *ICCPS*, pages 1–9. IEEE, 2016.

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

[15] D. Gunatilaka, M. Sha, and C. Lu. Impacts of channel selection on industrial wireless sensor-actuator networks. In *INFOCOM*, pages 1–9. IEEE, 2017.

[16] J. Guo and P. Orlik. Self-transmission control in iot over heterogeneous wireless networks. In *ICUFN*, pages 898–903. IEEE, 2017.

[17] A. Hernandez. Modification of the ieee 802.15. 4 implementation extended gts implementation. *KTH, Tech. Rep.,*, 2011.

[18] Y.-K. Huang, A.-C. Pang, and H.-N. Hung. An adaptive gts allocation scheme for ieee 802.15. 4. *IEEE Trans. Parallel Distrib. Syst.*, 19(5):641–651, 2008.

[19] IEEE. IEEE 802.15.4 standard. https://standards.ieee.org/findstds/standard/802.15.4-2015.html, 2015.

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

[21] M. Köppe. On the complexity of nonlinear mixed-integer optimization. In *Mixed Integer Nonlinear Programming*, pages 533–557. Springer, 2012.

[22] H. Lee, A. Cerpa, and P. Levis. Improving wireless simulation through noise modeling. In *IPSN*, pages 21–30. ACM, 2007.

[23] B. Li, Y. Ma, T. Westenbroek, C. Wu, H. Gonzalez, and C. Lu. Wireless routing and control: a cyber-physical case study. In *ICCPS*, pages 1–10. IEEE, 2016.

[24] B. Li, L. Nie, C. Wu, H. Gonzalez, and C. Lu. Incorporating emergency alarms in reliable wireless process control. In *ICCPS*, pages 218–227. IEEE, 2015.

[25] 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 Trans. Cybernetics*, 45(11):2378–2389, 2015.

[26] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang. A review of industrial wireless networks in the context of industry 4.0. *Wireless Netw.*, 23(1):23–41, 2017.

[27] Y. Li, Y. Chen, and I. Podlubny. Stability of fractional-order nonlinear dynamic systems: Lyapunov direct method and generalized mittag–leffler stability. *Comput. Math. Appl.*, 59(5):1810–1821, 2010.

[28] X. Liu and A. Goldsmith. Kalman filtering with partial observation losses. In *CDC*, volume 4, pages 4180–4186. IEEE, 2004.

[29] M. Lješnjanin, D. E. Quevedo, and D. Nešić. Packetized mpc with dynamic scheduling constraints and bounded packet dropouts. *Automatica*, 50(3):784–797, 2014.

[30] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proc. IEEE*, 104(5):1013–1024, 2016.

[31] Y. Ma, D. Gunatilaka, B. Li, H. Gonzalez, and C. Lu. Holistic cyber-physical management for dependable wireless control systems. *ACM Trans. Cyber-Physical Syst.*, 3(1):3, 2018.

[32] Y. Ma and C. Lu. Efficient holistic control over industrial wireless sensor-actuator networks. In *ICII*, pages 89–98. IEEE, 2018.

[33] 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 *CDC*, pages 1231–1237. IEEE, 2018.

[34] 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. *arXiv*, 2018.

[35] E. G. Peters, D. E. Quevedo, and M. Fu. Controller and scheduler codesign for feedback control over IEEE 802.15. 4 networks. *IEEE Trans. Control Syst. Technol.*, 24(6):2016–2030, 2016.

[36] 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 Trans. Embedded Comput. Syst.*, 13(4s):128, 2014.

[37] M. Sha, D. Gunatilaka, C. Wu, and C. Lu. Empirical study and enhancements of industrial wireless sensor–actuator network protocols. *IEEE IoT J.*, 4(3):696–704, 2017.

[38] Y. Shi and H. Fang. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *Int. J. Control*, 83(3):538–551, 2010.

[39] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control*, 49(9):1453–1464, 2004.

[40] J. W. Taylor. Exponential smoothing with a damped multiplicative trend. *Int. J. Forecasting*, 19(4):715–725, 2003.

[41] E. Toscano and L. L. Bello. Multichannel superframe scheduling for IEEE 802.15. 4 industrial wireless sensor networks. *IEEE Trans. Ind. Informat*, 8(2):337–350, 2012.

[42] Z. Wang, F. Yang, D. W. Ho, and X. Liu. Robust h_∞ control for networked systems with random packet losses. *IEEE SMC, Part B (Cybernetics)*, 37(4):916–924, 2007.

[43] J. Wu and T. Chen. Design of networked control systems with packet dropouts. *IEEE Trans. Autom. Control*, 52(7):1314–1319, 2007.

[44] Y. Zhan, Y. Xia, and M. Anwar. Gts size adaptation algorithm for ieee 802.15. 4 wireless networks. *Ad Hoc Netw.*, 37:486–498, 2016.