

# Efficient Holistic Control over Industrial Wireless Sensor-Actuator Networks

Yehan Ma

Cyber-Physical Systems Laboratory  
Washington University in St. Louis, USA

Chenyang Lu

Cyber-Physical Systems Laboratory  
Washington University in St. Louis, USA

**Abstract**—Process automation is embracing wireless sensor-actuator networks (WSANs) in the era of Industrial Internet. Despite the success of WSANs for monitoring applications, feedback control poses significant challenges due to data loss and stringent energy constraints in WSANs. *Holistic control* adopts a cyber-physical system approach to overcome the challenges by orchestrating network reconfiguration and process control at run time. In this paper, we explore efficient holistic control designs to maintain control performance while reducing the communication cost. The contributions of this work are four-fold: (1) We introduce a holistic control architecture that integrates low-power wireless bus (LWB) and two control strategies, *rate adaptation* and *self-triggered control*, specifically proposed to reduce communication cost; (2) We design novel wireless network mechanisms to support rate adaptation and self-triggered control, respectively, in a multi-hop WSAN; (3) We build a real-time network-in-the-loop simulator that integrates MATLAB/Simulink and a three-floor WSAN testbed to evaluate wireless control systems; (4) We empirically explore the tradeoff between communication cost and control performance under alternative holistic control approaches. Our case studies show that rate adaptation and self-triggered control offer advantages in control performance and energy efficiency, respectively, in normal operating conditions. The advantage in energy efficiency of self-triggered control, however, may diminish under harsh physical and wireless conditions due to the cost of recovering from data loss and physical disturbances.

**Index Terms**—industrial wireless control, multi-hop mesh network, network reconfiguration, network-in-the-loop simulation, cyber-physical systems

## I. INTRODUCTION

Wireless sensor-actuator networks (WSANs) are being adopted in industrial process automation for their advantages in reducing deployment and maintenance costs. However, while existing WSANs are usually used for monitoring, it remains challenging to support feedback control loops over WSANs. First, the control performance of industrial processes can be affected by data losses in WSANs. In industry, control performance is closely related not only to the factory's profits, but also machine operator's safety and the environment. Second, given the difficulty to replace batteries in harsh industrial environments, it is crucial to improve the energy efficiency of WSANs while maintaining control performance. Finally, wireless control systems (WCS) must be resilient to both disturbances to the physical plant and external interferences to the wireless networks. Therefore, a practical and dependable industrial WCS must meet the following requirements: (1)

*control performance*, which brings economic benefits; (2) *energy efficiency*, which reduces maintenance cost; and (3) *resiliency*, which prevents accidents.

Traditionally, the wireless network and the physical process are managed separately in a WCS at run time. The lack of coordination between network and plant management forces conservative designs that trade energy for control performance. For example, a WCS may rely on high sampling rates to guarantee control performance under worst-case conditions, even though the same sampling rates may result in excessive communication cost under normal conditions. Conversely, a less conservative design may result in a fragile system vulnerable to physical disturbances and/or wireless interference.

In contrast to the traditional approach, the *holistic control* approach aims to enhance the resiliency and efficiency of WCS by *cojoining* network reconfiguration and process control [22]. The core of holistic control is a *holistic controller* that generates not only control inputs to physical plants, but also network configurations based on control needs at run time.

In this work we explore efficient holistic control designs to maintain control performance at low energy cost. We develop holistic control approaches that incorporate two alternative strategies, *rate adaptation* (RA) and *self-triggered control* (ST) [38], proposed specifically to improve the efficiency of control systems. We note that RA introduces adaptation in a traditional *time-driven* control approach, while ST is a representative *event-driven* control approach. Exploring both strategies in holistic control allows us to investigate the design tradeoff involved in holistic control design. Specifically, the contributions of this work are four-fold.

- We introduce a new *holistic control architecture* that integrates multi-hop wireless networks running the Low-power Wireless Bus (LWB) protocol and two alternative control strategies, *rate adaptation* and *self-triggered control*;
- We design *robust network adaptation* mechanisms to support rate adaptation and self-triggered control, respectively, in multi-hop LWB networks;
- We build *RT-WCPS*, a real-time network-in-the-loop simulator that integrates MATLAB/Simulink and a physical WSAN testbed to evaluate wireless control systems;
- We empirically explore the tradeoff between communication cost and control performance under alternative holistic control approaches.

Our case studies show that RA and ST offer advantages in control performance and energy efficiency, respectively, in normal operating conditions. The advantage in energy efficiency of ST, however, may diminish under harsh physical and wireless conditions due to the cost of recovering from data loss and physical disturbances.

The rest of the paper is organized as follows: Sec. II reviews related works on WCS designs. Sec. III introduces the system architecture of holistic control system. Secs. IV and V detail the control and network designs of RA and ST. Sec. VI presents the real-time wireless cyber-physical simulator (RT-WCPS), and Sec. VII analyzes the experimental results.

## II. RELATED WORK

The problem of resilient and efficient wireless control has been investigated in the fields of control theory, wireless networks, and more recently network-control co-designs.

In control theory, state observers [37] (e.g., extended Kalman filter) have been introduced to handle packet loss and communication latency in networked control systems. To reduce communication cost, aperiodic control has been proposed as an alternative to periodic time-driven control. Examples include event-triggered control [24], [26] and self-triggered control [25]. However, existing implementation of aperiodic control was based on a single-hop wireless network [5] instead of the multi-hop WSNs that are widely adopted in process industrials due to their flexibility and scalability in industrial environments. Supporting aperiodic control on a multi-hop WSN is challenging because industrial WSN standards usually employ TDMA protocols for predictable communication. The aperiodic communication triggered by aperiodic control is incompatible with the periodic, time-driven nature of communication in industrial multi-hop WSNs.

In wireless networks, given latency, packet delivery, energy consumption bounds by the control designers, on-line network adaptations can achieve optimized energy-efficiency [28], reliability [9], and real-time performance [43] under various wireless channel conditions and network topologies, but few are cognizant of control performance directly.

In cyber-physical systems, recent effort on network-control co-design aims to jointly optimize the wireless network and control at *design* time [29]. Wireless network designs have been tailored for control, e.g., sampling rate optimization [8], [30] and asymmetric routing for control [19]. These efforts focus on offline designs instead of online adaptation, which limits the resiliency and efficiency of WCS operating in dynamic conditions (e.g., under network interference and physical disturbance).

In previous work [22], we proposed the concept of holistic control that cojoins network management and physical control at run time. As a simple proof of concept, we presented a holistic control example that adjusts the numbers of transmissions based on physical states. In this paper, we generalize the designs of holistic control by incorporating more sophisticated control approaches, namely RA and ST. The new control approaches require more sophisticated network reconfiguration

mechanisms that are both efficient and robust. Furthermore, the alternative control approaches (time-driven vs. event-driven) allows us to explore the design tradeoff involved holistic control in multi-hop WSNs.

Simulation tools are of vital importance to wireless control. Truetime [7] supports co-simulation of controller tasks, network transmissions, and continuous plant dynamics. Wireless cyber-physical simulator (WCPS) [18] integrates MATLAB/Simulink and TOSSIM [17]. However, those simulators produce simulations that cannot always capture the real-world behavior of WSNs. Network-in-the-loop simulations were implemented in [27]. Experiments presented in [5] integrate two double-tank systems with a single-hop wireless network. [23] and [6] integrate two real inverted pendulums and a 13-node WSN multi-hop testbed, achieving sampling rates of tens of millisecond. However, the physical plants in laboratory settings used in those experiments cannot represent large-scale industrial processes. In this work we design and implement a network-in-the-loop simulator which integrates MATLAB/Simulink simulations and a 70-node WSN testbed.

## III. WIRELESS CONTROL SYSTEM ARCHITECTURE

Fig. 1 depicts the holistic wireless control architecture. The holistic controllers (1) control the physical plants by communicating with sensors and actuators through a multi-hop WSN, and (2) reconfigure the WSN based on control needs at run time. Multiple control loops share the same WSN.

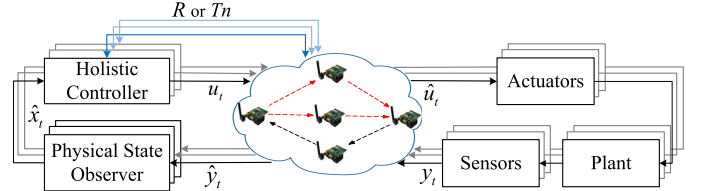


Fig. 1: Holistic WCS architecture.

### A. Physical control system

In this paper, our control design and analysis model the physical plant as a linear time-invariant system (LTI) described by (1) since a wide variety of systems can be represented with satisfactory accuracy by LTI model. The proposed wireless network reconfiguration mechanisms however are not limited to LTI systems, and are also applicable to nonlinear and time-variant systems.

$$x_{t+1} = Ax_t + Bu_t, y_t = Cx_t, \quad (1)$$

where  $t$  is the time index,  $x_t \in \mathbb{R}^n$  is the state vector,  $u_t \in \mathbb{R}^m$  is the input vector,  $y_t \in \mathbb{R}^p$  is the output vector,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ , and  $C \in \mathbb{R}^{p \times n}$ . We assume that the pair  $(A, B)$  is controllable and that the pair  $(A, C)$  is observable. A linear state feedback controller  $u_t = Kx_t$  renders the closed-loop system asymptotically stable.

At time  $t$ , a sensor sends its measurements  $y_t$  to a remote holistic controller over the multi-hop WSN. On the controller side, a state observer [37] estimates the states of the plant. Based on the estimated state vector  $\hat{x}_t$ , the feedback controller generates the control command  $u_t = K\hat{x}_t$  and sends it to the actuator through the WSN. The actuators then apply  $\hat{u}_t$  to

the physical plant. If  $u_t$  fails to be delivered by the deadline, the actuator reuses the control input of last period,  $\hat{u}_{t-1}$ .

The stability of this control system can be represented by the Lyapunov function. System (1) is stable if there exists a positive definite Lyapunov function [42]

$$V(x_t) = x_t^T P x_t, \quad (2)$$

such that

$$V(x_{t+1}) - V(x_t) = x_t^T ((A+BK)^T P (A+BK) - P) x_t = -x_t^T Q x_t, \quad (3)$$

where  $P, Q$  are positive definite matrixes, with  $P$  satisfying the discrete-time Lyapunov equation:

$$(A+BK)^T P (A+BK) - P = -Q. \quad (4)$$

### B. Wireless sensor-actuator network

1) *Low-Power Wireless Bus (LWB)*: The WSN extends the LWB [12] protocol to support data communication and network reconfiguration for holistic control. LWB is based on Glossy [11], a fast-flooding protocol that exploits the constructive interference among concurrent transmissions of radios compatible with the IEEE 802.15.4 standard. The flooding process is entirely driven by radio events, i.e., a transmission is triggered by completing a packet reception, which drastically speeds up the process. Under LWB, nodes take turns to flood their packets using Glossy according to a global schedule [12]. A sink node is responsible for disseminating the schedule to all nodes in the network.

Adopting LWB as the underlying communication protocol brings significant benefits. Thanks to Glossy flooding, communication in LWB is topology independent. Additionally, fast Glossy flooding achieves propagation latency within 10 ms over 100 nodes (8 hops, 3 transmissions). We can take the advantage to realize fast network reconfiguration by quickly flooding new network configurations across the entire network, an important feature as network reconfiguration is a key element of holistic control.

2) *Implicit scheduling of multi-rate LWB*: Unlike prior work [43] which uses a centralized scheduler node to operate real-time scheduling algorithms, we tailored LWB for implicit scheduling. In implicit scheduling, all nodes schedule themselves based on information from holistic controllers, such as flooding rates or next event timers of each control loop.

We define a data flow of WSN as  $f_{ij}$ , which has a source node  $s_{ij}$  and a destination node  $d_{ij}$ , where  $i \in \{1, 2, 3, \dots, n\}$  is the control loop index, and  $j \in \{1, 2, 3, \dots, m_i\}$  is the flow index of the control loop  $i$  ( $l_i$ ). Accordingly,  $n$  is the number of control loops, and  $m_i$  is the number of data flows in  $l_i$ . For example, the control loop  $l_1$  has two data flows  $f_{11}$  and  $f_{12}$ , among which  $f_{11}$  is a sensing flow transmitting measurements from a sensor node ( $s_{11}$ ) to a controller node ( $d_{11}$ ), and  $f_{12}$  is an actuation flow transmitting control command from a controller node ( $s_{12}$ ) to an actuator ( $d_{12}$ ). Note that a MIMO control loop can have multiple sensing and actuation flows. The update rate of control commands in the control loop  $l_i$  is denoted as  $R_i$ . The operation period of  $l_i$  is  $T_i = \frac{1}{R_i}$ . We assume the rates of the flows in one control loop are equal.

In implicit scheduling of data flows, each node stores a static global schedule of all data flows, denoted by entries  $f_{ij}[s_{ij}, d_{ij}, t_{ij}]$ ,  $t_{ij}$  is the relative time slot reserved for flow  $f_{ij}$  in LWB period  $T = \frac{1}{R}$ . LWB operates at the highest rate of all the control loops,  $R = \max_{1 \leq i \leq n} R_i$ . Fig. 2 shows a simple static schedule. We assume there are three control loops and each loop has one flow. All loops have same rate  $R_1 = R_2 = R_3 = \frac{1}{T}$ . Thus, the rate of LWB is  $R = \frac{1}{T}$ . Therefore, we get the static schedule entries:  $f_{11}[2, 1, 1]$ ,  $f_{21}[3, 4, 2]$ ,  $f_{31}[4, 1, 3]$ . We can see that in each period  $T$ , the synchronization message  $S$  is flooded by the sink node in the beginning of every period, followed by three data slots assigned for three flows.

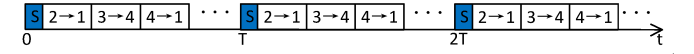


Fig. 2: LWB with static global schedule. ( $f_{11}$ , node2  $\rightarrow$  node1,  $\frac{1}{T}$  Hz;  $f_{21}$ , node3  $\rightarrow$  node4,  $\frac{1}{T}$  Hz;  $f_{31}$ , node4  $\rightarrow$  node1,  $\frac{1}{T}$  Hz.)

This static schedule is a “worst-case-guaranteed” schedule since it is designed based on the most strict latency requirements of the control loops under worst-case conditions. In other words, the WSN operates at its highest rate and resource cost when adopting this static schedule. Note that the static schedule can be calculated offline using any scheduling algorithm, such as earliest deadline first or rate monotonic. Because industrial process control systems are large-scale, most of the sampling rates are lower than 1 Hz [41]. By adopting fast Glossy flooding (flooding a packet over 100 nodes within 10 ms [11]), WSN can guarantee the schedulability of tens of data flows, which indicates the feasibility of the static schedule. We refer interested readers to [15], [43] for network designs with tighter real-time requirements.

To implement multi-rate LWB using implicit scheduling, besides the static global schedule, the only information that all nodes need are the rates of all the control loops  $R_i$ . In order to make the *implicit scheduling* work properly, the potential  $T_i$  of all the loops should be set to integral multiples of the shortest period  $T$ . Then each node can independently decide whether to flood  $f_{ij}$  or sleep at  $t_{ij}$  within the time interval  $[(k-1)T, kT]$ ,  $k = 1, 2, 3, \dots$ , depending on  $R_i$ . Fig. 3 shows an example of the *implicit scheduling* with the static schedule in Fig. 2, where  $R_1 = \frac{1}{T}$ ,  $R_2 = \frac{1}{2T}$ , and  $R_3 = \frac{1}{4T}$ . All nodes flood  $f_{11}$  at the first data slot of every period  $T$ , flood  $f_{21}$  at the second data slot every other period  $T$ , and flood  $f_{31}$  at the third data slot every  $4T$ . They sleep at the rest blank data slots.



Fig. 3: Implicit scheduling. ( $f_{11}$ , node2  $\rightarrow$  node1,  $\frac{1}{T}$  Hz;  $f_{21}$ , node3  $\rightarrow$  node4,  $\frac{1}{2T}$  Hz;  $f_{31}$ , node4  $\rightarrow$  node1,  $\frac{1}{4T}$  Hz.)

In implicit scheduling, since each node stores the static schedule, the network reconfiguration commands can be generated by any source nodes in WSN distributively, in contrast to *centralized scheduling* in which the whole schedule is sent by the sink in the beginning of each period  $T$ . We will present how network reconfiguration signals, such as  $R_i$ , are disseminated in Sec. IV-B and V-B.

### C. Holistic management

As shown in Fig. 4, we develop a holistic control architecture that bridges the gap between the plant control and WSAN management. Based on the current status of physical plants and WSAN, the holistic controller generates two kinds of commands at the same time, one for dynamically adjusting the network configuration, and the other for operating the physical plants. In the following two sections, we focus on two specific efficient holistic control designs: rate adaptation and self-triggered control over a multi-hop mesh network.

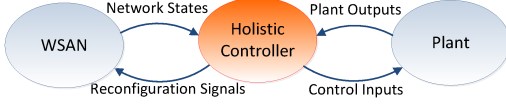


Fig. 4: Holistic management of WCSs.

### IV. RATE ADAPTATION

The data flow rates of a WSAN have direct impacts on control performance and energy cost. The higher the rates, the better the control performance, but also the higher the energy cost [8]. In this section, in order to ensure the control performance while reducing the network energy cost, we adjust the rates of the WSAN based on the changing control performance during runtime. We will then introduce the holistic controller design and the network design of rate adaptation.

#### A. Control design

We employ a similar adaptation algorithm proposed in [22] (Alg. 1). The value of the Lyapunov function (2), regarded as the metric of the control performance, provides the bounds of the state error. Given (2),

$$\alpha_1 \|x_t\|^2 \leq V(x_t) \leq \alpha_2 \|x_t\|^2, \quad (5)$$

where  $\alpha_1$  and  $\alpha_2$  are the smallest and largest eigenvalues of  $P$ , respectively. Therefore we use the value of the Lyapunov function to trigger the rate adaptation. Given a certain customized state error bound, denoted as  $se = \|x_{se}\|^2$ , we set the rate increasing threshold  $V_{Ith} = \alpha_1 \|x_{se}\|^2$ . Based on (5), we have  $\|x_t\|^2 \leq \|x_{se}\|^2$ , if  $V(x_t) \leq V_{Ith}$ . Furthermore, we adopt a more stringent decreasing threshold to indicate that the system performs well. The decreasing threshold is  $V_{Dth} = \lambda \alpha_1 \|x_{se}\|^2$ ,  $\lambda \in (0, 1)$ . If  $V(x_t)$  remains below  $V_{Dth}$  for a customized time interval  $\tau$ , the control system is regarded in good condition.

Given (3),

$$V(x_{t+1}) - V(x_t) \leq -\beta \|x_t\|^2, \quad (6)$$

where  $\beta$  is the smallest eigenvalue of  $Q$ . Given (5) and (6), we can get the upper bound of the ideal Lyapunov function, described by (7). We set this upper bound as the trigger of succeeding rate increases.

$$V(x_{t+j}) \leq (1 - \frac{\beta}{\alpha_2})^j V(x_t). \quad (7)$$

The rate adaptation algorithm of a holistic controller is presented in Alg. 1, and its complexity is  $O(1)$ .

Our rate adaptation strategy cannot mathematically guarantee closed-loop system stability over the WSAN unless extra

---

#### ALGORITHM 1: Rate adaptation algorithm for $l_i$

---

**Input:**  $x_t, t, \tau, t_0 = t, \lambda$ , candidate rates (ascending):  $\{R_{i1}, R_{i2}, \dots, R_{iq}\}$ , current  $R_i = R_{ij}, A, B, K, P, Q$

**Output:** updated  $R_i$

Calculate  $V(x_t)$  as defined in (2), and  $V_{Dth}, V_{Ith}$ ;

**if**  $V(x_t)$  remains below  $V_{Dth}$  for a time interval of  $\tau$ , **and**

$R_i > R_{i1}$  **then**

$R_i \leftarrow R_{ij-1}$ ; //  $R_i$  decreases

**else if**  $V(x_t) > V_{Ith}$  **and**  $R_i < R_{iq}$  **then**

**if** last rate adaptation is a decrease **then**

$R_i \leftarrow R_{ij+1}$ ; //  $R_i$  increases

$t_0 \leftarrow t$ ;

**end**

**if** last rate adaptation is an increase **and**

$V(x_t) > (1 - \frac{\beta}{\alpha_2})^{t-t_0} V(x_{t_0})$  **then**

$R_i \leftarrow R_{ij+1}$ ; // succeeding  $R_i$  increases

$t_0 \leftarrow t$ ;

**end**

**else**

$R_i$  remains constant

**end**

---

assumptions are considered, which is a common approach in the literature [3], [13]. The applicability of these assumptions depends on the properties of the industrial process, thus we avoid imposing a particular framework in this paper. Instead, our algorithm takes a best-effort approach towards balancing the closed-loop performance and network rate, which is a practical heuristic in real-world scenarios. The relationships between the rate and control performance are well studied, and we refer interested readers to [32], [40]. If we set all potential rates higher than the lowest stabilizing rate, the closed-loop stability can be achieved.

#### B. Network reconfiguration

In this section, we present a run-time rate adaptation protocol for a mesh WSAN. We allow each holistic controller to adjust the WSAN rate of its own loop, according to Alg. 1.

1) *Candidate rates selection:* Sec. IV-A considers when to adjust the rate of each loop. The candidate rates are also important design factors. To ensure that the rate transient processes work properly, the potential rates of each loop need to be designed intentionally. First, according to Sec. III-B2, when the offline scheduler schedules data flow  $f_{ij}$ , it reserves time slots for  $R = \max_{1 \leq i \leq n} R_i$ . LWB operates at the rate of  $R$ . Second, the candidate period  $T_i$  of all the loops should be set to integral multiples of the shortest period  $T = \frac{1}{R}$ . Third, to ensure that the rate adaptations work properly with packet loss recovery, which will be discussed later in Sec. IV-B3, the rate selection strategy should generate as many common time slots among different candidate rates as possible for each flow, e.g.,  $(\frac{1}{T}, \frac{1}{2T}, \frac{1}{4T})$  or  $(\frac{1}{T}, \frac{1}{3T}, \frac{1}{9T})$ .

2) *Network reconfiguration based on piggyback:* The holistic controller of  $l_i$  adopts a *piggyback* mechanism to disseminate a newly computed  $R_i$  for data flows  $f_{ij}, j = 1, 2, \dots, m_i$ . The holistic controller of  $l_i$  piggybacks  $R_i$  with the actuation



command. Therefore the data field of the actuation packet is  $[l_i, R_i, \text{Data}_i]$ . Because of the flooding nature of LWB, all nodes in the network can receive this update. Once a node receives an updated  $R_i$ , it will calculate a new schedule based on  $R_i$ , as described in Sec. III-B2.

The distributed network reconfiguration based on piggyback has several benefits over the conventional centralized network reconfiguration. First, this piggyback mechanism helps reduce energy cost by utilizing existing actuation data flows, saving the time and energy needed to calculate and deliver the whole schedule in every period. Second, the network reconfiguration commands can be flooded by any source nodes in WSN distributively, in contrast to *centralized scheduling*, in which the whole schedule should be sent by the sink. In addition, implicit and distributed scheduling using piggyback is more reliable than a centralized scheduler. Packet loss in implicit scheduling affects only one loop, but the packet loss of centralized scheduling can affect all data flows.

3) *Packet loss recovery*: If a node loses the packet with the updated rate of  $l_i$ , it will continue to use the current  $R_i$  until another actuation packet of  $l_i$  is received. Therefore, it is possible that, at the same time, different nodes along the route of a flow are using different rates. Nevertheless, it is still possible for nodes to eventually receive the update. If all candidate rates share more common slots, as described in Sec. IV-B1, the node will recover faster from packet loss.

## V. SELF-TRIGGERED CONTROL

Self-triggered control [25], an aperiodic event-driven control design, improves the efficiency of the network. The first single-hop wireless network protocol for aperiodic control is presented in [5]. However, due to the lack of network protocol, aperiodic control designs have not been adopted in multi-hop mesh networks. In this section, we respectively introduce control design and network design of self-triggered control.

### A. Control design

Intuitively, self-triggered control triggers sensing and actuation events only when certain control performance is predicted to be lost. The self-triggered strategy we present in this paper is motivated by [5]. The desired control performance is defined by a decreasing function  $S(x_t)$ , upper bounding the evolution of Lyapunov function  $V(x_t)$ :  $V(x_t) \leq S(x_t)$ . The time of the next sensing and actuation events is  $t_k = \min\{t > t_{k-1} | V(x_t) - S(x_t) \geq 0\}$ . Self-triggered control and its stability are well studied in the literature. We refer interested readers to [4], [14], [25]. Here, we adopt a feasible decreasing  $S(x_t)$ , as follows:

$$S(x_t) = V(x_{t_{k-1}})e^{-\gamma V(x_{t_{k-1}})^\delta (t-t_{k-1})}, \quad (8)$$

We induce the term  $\gamma V(x_{t_{k-1}})^\delta$ ,  $\gamma, \delta > 0$ , which makes the decreasing rate of  $S(x_t)$  adapt to the value of the Lyapunov function (state error). That is, when  $V(x_{t_{k-1}})$  is large, which indicates severe state error, the  $S(x_t)$  decreases faster. Therefore, the sensing and actuation events are more likely to be triggered. On the other hand, when  $V(x_{t_{k-1}})$  is small, which indicates the current states are close to equilibrium point, the

$S(x_t)$  decreases slower. The sensing and actuation events are unnecessary and less likely to be triggered.

### B. Network protocol for self-triggered control

1) *Self-triggered transmissions*: Due to the predictive nature of self-triggered control, the network knows *a priori* when the event will be triggered by the holistic controllers. Therefore, nodes know the next time when they should wake up and flood data. Within the inter-transmission interval, the nodes sleep. In this way, the energy costs of nodes can be reduced compared with periodic control at the highest rate.

Similar to the network protocol of rate adaptation, the holistic controller uses the piggyback mechanism to disseminate a newly computed time of next transmission  $Tn_i$  for all data flows of  $l_i$ . Again,  $Tn_i$  should be integral multiples of  $T$ . The holistic controller piggybacks  $Tn_i$  with the actuation command. Therefore the data field of the actuation packet is  $[l_i, Tn_i, \text{Data}_i]$ . Because of flooding, all nodes in the network can receive this update. In a node, each data flow has an event timer. Once a node receives a  $Tn_i$ , it will set the value of  $\text{Timer}_{ij}$  to  $Tn_i$  and start counting down from the next period. If the  $\text{Timer}_{ij}$  expires in a node, the node will wake up and flood in relative slots  $t_{ij}$ . Fig. 5 shows an example of self-triggered transmissions based on LWB. At the first period,  $f_{21}$  is flooded, and node 3, which is the *source* of  $f_{21}$ , receives and floods  $Tn_2 = 3T$  at slot that is assigned for  $f_{21}$ . Therefore, the next  $f_{21}$  is transmitted  $3T$  later at the fourth period. At the second period,  $f_{11}$  and  $f_{31}$  are transmitted, and  $Tn_1 = T$ ,  $Tn_3 = 2T$ , respectively. Therefore, the next  $f_{11}$  is transmitted at the third period, and  $f_{31}$  at the fourth period.

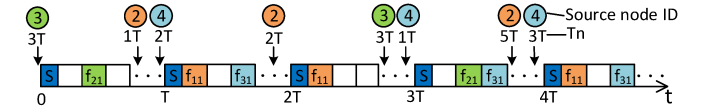


Fig. 5: Self-triggered transmission based on LWB ( $f_{11}$ , node2  $\rightarrow$  node1;  $f_{21}$ , node3  $\rightarrow$  node4;  $f_{31}$ , node4  $\rightarrow$  node1).

2) *Packet loss recovery*: If all nodes receive  $Tn_i$ , and are synchronized well, they will wake up and flood  $f_{ij}$  at the same time. However, unlike rate adaptation based on LWB, which can self-recover from packet loss if the candidate rates have been carefully selected, self-triggered transmissions based on LWB are less resilient to packet loss. If a node fails to receive  $Tn_i$ , it is possible that it will not wake up at the right time for the next transmission and will become unsynchronized with other nodes for  $f_{ij}$  forever. Therefore, it is of vital importance to come up with effective and efficient strategies to recover from packet loss. We propose the following packet loss recovery strategy: if a node wakes up but does not receive a packet with  $Tn_i$ , it should re-awake at the highest rate  $R$ , until another packet with  $Tn_i$  is received.

## VI. REAL-TIME WCPS

To experiment with wireless control over real-world WSNs, we develop a real-time wireless cyber-physical simulator (RT-WCPS). In this section, we first present the architecture of RT-WCPS. Then we analyze its real-time performance.

### A. Architecture of RT-WCPS

RT-WCPS integrates *MATLAB/Simulink Desktop Real-time (SLDRT)* [2] and a *3-floor WSN testbed* [33], [34]. The architecture of RT-WCPS is shown in Fig. 6. Note that this figure shows the architecture of one wireless control loop. Several control loops can share the same WSN.

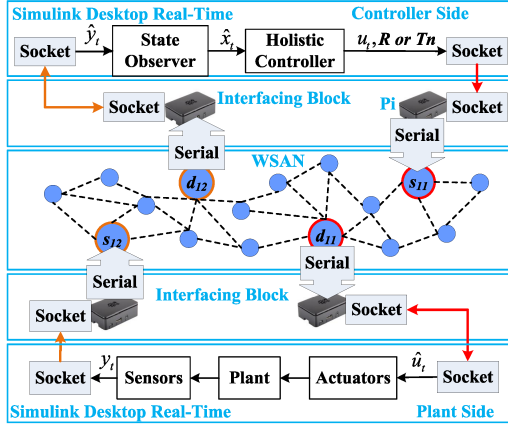


Fig. 6: Architecture of RT-WCPS.

*SLDRT* is used to simulate the physical part of the WCS: physical plants, controllers, state observers, and physical disturbance. In practice, industrial plants usually operate continuously or at very high rates. However, the wireless communication and controller execute at a relatively low rate because of the communication and computation latencies. Therefore, *SLDRT* modules are operated at different rates in our design.

The *3-floor WSN testbed* is deployed on the 3rd to 5th floors of Jolley Hall at Washington University in St. Louis, as shown in Fig. 7. It consists of 70 TelosB motes. Each mote is equipped with Chipcon CC2420 radio compliant with the IEEE 802.15.4 standard and a TI MSP430 microcontroller. 40 Raspberry Pis with a backplane network are used for the measurement and management of the wireless network [21].

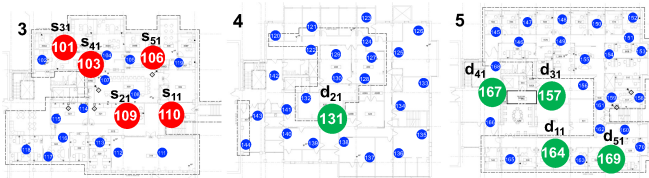


Fig. 7: 3-floor WSAW testbed in Jolley Hall of Washington University in St. Louis (from left to right, third floor to fifth floor).

The *interfaces* between SLDRT and WSAN are socket connections between the PCs that run SLDRT and the Pis, and serial connections between the Pis and the end nodes. In this way, the end nodes  $s_{ij}, d_{ij}$  of the sensing and actuation flows  $f_{ij}$  can be any nodes in the testbed.

### B. Real-time network-in-the-loop simulation

Both SLDRT and the 3-floor WSA testbed operate in real-time. To evaluate the real-time performance of the RT-WCPs, we measure the latency caused by each module. In our design, sensing and actuation flows have the same overhead induced by interfaces, since they have the same types of interfaces

between physical parts and WSN as in Fig. 6, and all data flows share the same WSN with independent interfaces.

We use the latencies of one actuation flow as an illustrative example. First, we adopt the *Precision Time Protocol (PTP)* to synchronize the PC that runs SLDRT and the Pis. *PTP* is a protocol used to synchronize clocks throughout a network. It achieves clock accuracy in the sub-microsecond range [1]. Then, we record the completion timestamps of each module on corresponding machines (1) the physical modules, (2) the actuation flow from Simulink to  $s_{11}$ , (3) the transmissions in WSA, (4) the actuation flow from  $d_{11}$  to Simulink. Finally, we draw the timeline of RT-WCS and analyze the latencies, as shown in Fig. 8. We set the sampling period to  $1s$ , which is the fastest update time supported by most industrial WSA products. From the timeline, the total overhead induced by interfaces between Simulink and the node is less than  $26\text{ ms}$  ( $2.6\%$ ). More than  $966\text{ ms}$  is reserved for communication over the WSA in each period, among which around  $175\text{ ms}$  is utilized for transmissions in this example. The results validate the real-time performance of RT-WCS. Please note that  $26\text{ ms}$  overhead is acceptable when we use RT-WCS to simulate industrial processes like oil refinery and mining, sampling periods of which are usually longer than  $1\text{ s}$  [41]. However, it is not acceptable in faster sampling period of tens of milliseconds. We will work on shortening this overhead in the future. We refer interested readers to [15], [23], [43] for network and WCS designs with tighter time requirements.

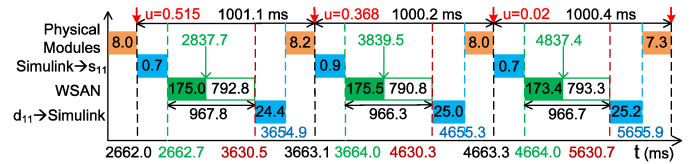


Fig. 8: Timeline of RT-WCPS

## VII. EVALUATION

In this section, we describe systematic trials of our wireless control designs using RT-WCPS. On the physical side, to represent an industrial process system, we use up to five 4-state load positioning systems that share the same WSA. On the WSA side, we evaluate the proposed network protocols over a 70-node WSA testbed [33], [34].

Because the state observer provides robust and theoretically sound protection against loss of sensing information [20], [35], [37], the WCSs are more sensitive to packet loss on the actuation side of the WSN [19]. Therefore, we focus on comprehensive actuation-network-in-the-loop simulations. We then empirically evaluate the tradeoff between rate adaptation and self-triggered control in communication cost and control performance under different operating conditions.

### A. Systems settings

1) *Physical system settings:* We run simulations of a realistic load positioning system [16], [36], which positions a load (L) using a motor with a ballscrew transmission. The motor is attached rigidly to a movable base platform (B). The load positioning is a 4-state nonlinear system as described in [16].

When the system is operated at low rates as in real industrial applications, the stiffness of the ballscrew and the potential energy stored in it are neglected in the model. The system can be simplified as a 4-state linear system [36]:

$$\dot{x}_t = A_c x_t + B_c u_t, y_t = C_c x_t, \quad (9)$$

where

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -d_L(\frac{1}{m_L} + \frac{1}{m_B}) & \frac{k_B}{m_B} & \frac{d_B}{m_B} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{d_L}{m_B} & -\frac{k_B}{m_B} & -\frac{d_B}{m_B} \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ \frac{1}{m_L} + \frac{1}{m_B} \\ 0 \\ -\frac{1}{m_B} \end{bmatrix},$$

and  $C_c = [1 \ 0 \ 0 \ 0]$ . Here,  $d_L$ ,  $m_L$ ,  $d_B$ ,  $m_B$ , and  $k_B$  are parameters of the load and base platforms, such as the mass, damping, and stiffness. The state vector is defined as  $x_t = [x_L(t) \ \dot{x}_L(t) \ x_B(t) \ \dot{x}_B(t)]^T$ , where  $x_L$  is the displacement of the load *relative* to the base platform,  $x_B$  is the *absolute* displacement of the base platform, and  $\dot{x}_L$  and  $\dot{x}_B$  are the speeds of the relative and absolute movements accordingly. We will stabilize the states of the load positioning system to the origin.

There are two kinds of plants. For the first kind, denoted as PLANT1,  $d_L = 15$ ,  $m_L = 100$ ,  $d_B = 10$ ,  $m_B = 10$ ,  $k_B = 5$ , and  $K = [-1.9393 \ -13.1373 \ 0.0842 \ -13.0264]$ . For the second kind, denoted as PLANT2,  $d_L = 10$ ,  $m_L = 15$ ,  $d_B = 3$ ,  $m_B = 5$ ,  $k_B = 2$ , and  $K = [-1.0076 \ -0.6317 \ -0.1954 \ -0.3814]$ . The second kind of plants have lower mass and damping, therefore their response time is shorter than that of PLANT1.

The rates of the controllers are fixed at 1 Hz, which are the same as the highest rate  $R$ . For simplicity, on the controller side, we discretize the continuous-time models using Euler discretization at 1 Hz. For each control loop, given the discrete-time model,  $K$ , and  $Q$ , we can get  $P$ ,  $\alpha_1$ ,  $\alpha_2$ , and  $\beta$  according to (4), (5), and (6), respectively. For all loops,  $Q = I_4$ ,  $\lambda = 0.1$ ,  $\|x_{se}\|^2 = 0.1$ , and  $\gamma = 1$ ,  $\delta = 2$ .

2) *WSAN settings*: The network protocols for *rate adaptation* and *self-triggered transmission* use Contiki [31]. The LWB operates at the rate  $R = 1$  Hz. The global static schedule has one synchronization slot, with a length of 25 ms, and 2-5 data slots, with lengths of 18 ms. 70 nodes participate in the transmissions. The synchronization packet is disseminated by the sink node (node 164) every 1 s. The synchronization packet size is 6 bytes, and the data packets are 25 bytes. Each data slot is used to transmit the control command  $u_t$  and network reconfiguration signals  $R$  or  $Tn$  of each control loop. Fig. 7 shows the source and destination pairs of five actuation flows over 3-floor WSAN. The transmission power is 0 dbm, and the retransmission number is 3.

3) *RT-WCPS settings*: We simulate the WCSs using RT-WCPS, which integrates a 70-node WSAN and SLDRT. We simulated two control loops sharing a WSAN for statistical results from Sec. VII-B to Sec. VII-E. Loop  $l_1$  controls a PLANT1. Loop  $l_2$  controls a PLANT2. The SLDRT modules of two loops are shown in Fig. 9. Each loop has its own holistic controller, and the controllers and the actuators communicate via actuation flows sharing the same WSAN. And we simulate five control loops sharing a WSAN to show the scalability

of RT-WCPS in Sec. VII-F. Loops  $l_1$ ,  $l_3$ , and  $l_5$  control 3 PLANT1s separately. Loops  $l_2$  and  $l_4$  control 2 PLANT2s.

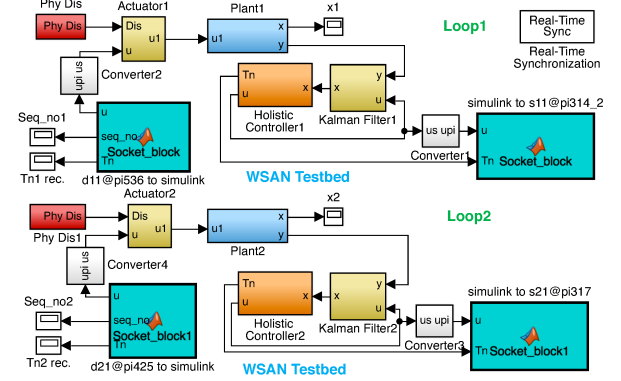


Fig. 9: SLDRT modules of RT-WCPS.

As presented in Sec. VI-A, modules in Fig. 9 operate at different rates. The physical plants run at 100 Hz. Controllers, Kalman filters, and actuators run at 1 Hz. The “worst-case-guaranteed” WSAN runs at 1 Hz, and WSAN can adjust its rates and operate self-triggered transmissions during runtime, based on control needs. In rate adaptation, we choose candidate rates: 1 Hz, 0.5 Hz, 0.25 Hz, which are reasonable rates for our load positioning systems with time constants of roughly 30 s. And they are also typical rates in industrial process control [41]. In self-trigger transmission, to provide robustness guarantees of the self-triggered control [5], we set the upper bound of the inter-transmission interval as 10 s.

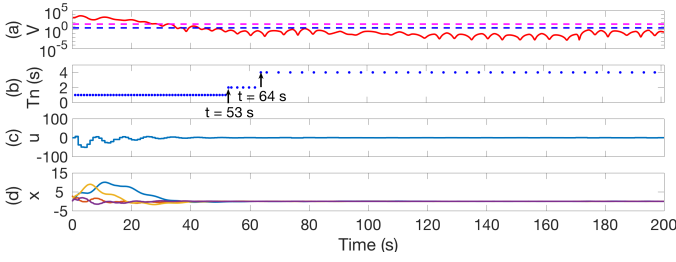
#### B. Normal network and physical conditions

We first evaluate the WCSs under normal conditions. The WSAN operated on IEEE 802.15.4’s channel 26. The average packet delivery ratio is 99.15%. And there is no physical disturbance. We present the results of five sets of network-in-the-loop simulations under the different management approaches:

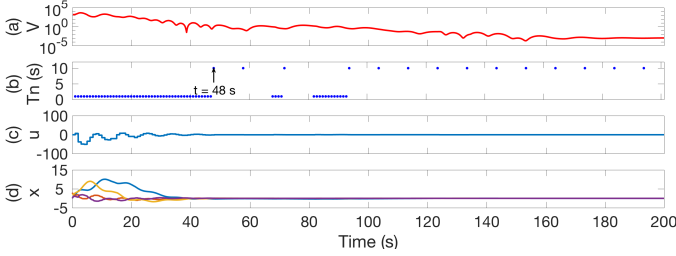
(1) *Rate adaptation (RA)*: Fig. 10A shows the response curves of loop 1. Plot (a) shows the Lyapunov function  $V(x_t)$ . The two dashed lines, from upper to lower, are the rate increase and decrease thresholds. In plot (b), each dot indicates a transmission (Tx), and the y-axis of the dot is the time of the next Tx. Plots (c) and (d) show the control command  $u_t$  and physical states  $x$ , respectively. When  $x$  is approaching the origin, as indicated by the decreases of  $V(x_t)$ , if  $V(x_t)$  is below the decrease threshold for  $\tau = 10$  s, the rate of the WSAN starts to decrease, as shown in (b). The rate changes from 1 Hz (1 Tx every 1 s), to 0.5 Hz (1 Tx every 2 s) at  $t = 53$  s, then to 0.25 Hz (1 Tx every 4 s) at  $t = 64$  s.

(2) *Self-triggered control (ST)*: Fig. 10B shows the response curve of ST. As shown in (b), since  $V(x_t)$  decreases, the inter-transmission interval changes from 1 s to 10 s at  $t = 48$  s. When  $V(x_t)$  increases at around 60 s to 90 s,  $Tn$  reduces to 1 s as soon as the timer expires.

(3) *Fixed rate time-driven control*: Existing WSANs typically employ time-drive transmissions with fixed rates, so we use three fixed rates of 1 Hz, 0.5 Hz and 0.25 Hz, denoted by 1, 2, 4 in following statistical results.



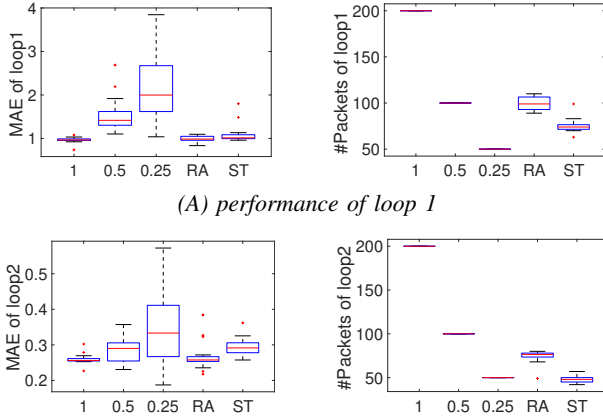
(A) time-driven control with rate adaptation



(B) self-triggered control

Fig. 10: Response curve under normal condition.

Next we run each experiment for 20 rounds with different initial values to statistically compare different management approaches. Fig. 11 shows the performances of two loops respectively. We use the mean absolute error (MAE) as the metric of control performance, and the number of packets sent through WSN as the metric of network cost. Both RA and ST can achieve similar control performances with fixed rate of 1 Hz, with a network cost (# of packets) reduction of more than 50%. Loop 2 has network cost reduction of more than 62% since it has shorter time constant. For both loops, ST is more aggressive in saving network cost than RA.



(A) performance of loop 1

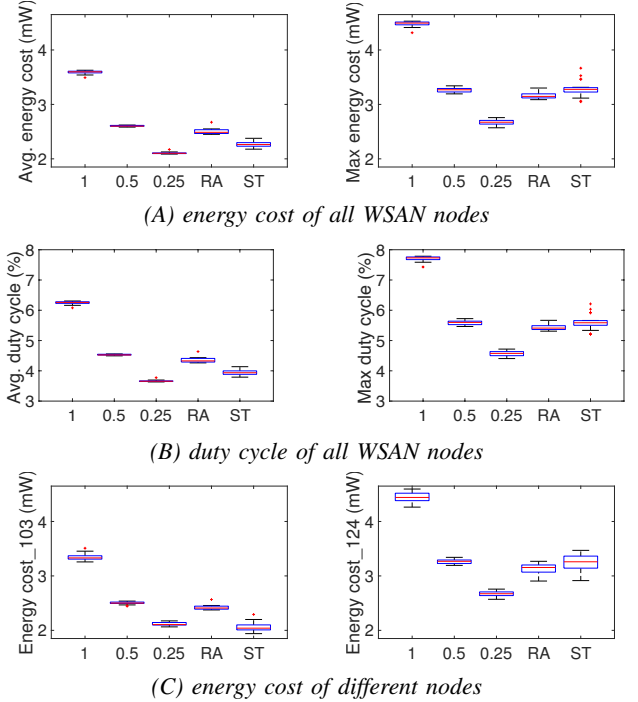
(B) performance of loop 2

Fig. 11: Performance under normal condition.

In reality, the total energy consumption of all flows, including the synchronization cost, is of interest. Therefore, we analyze power consumption over the WSN in detail. We collect the time spent in transmitting and listening per node per second using the Energest module [10] provided Contiki OS. The sum of transmitting and listening time is the radio-on time of the collection period, and the node sleeps in the rest of the period. We adopt the energy model in [39] to estimate the

energy cost based on transmitting and listening time. Fig. 12A and Fig. 12B show that the results of energy cost are consistent with that of duty cycle.

Fig. 12A shows the average and maximum energy costs of all 70 nodes. The average energy cost is consistent with the number of packets going through WSN. RA and ST save 40% energy, which is higher than energy cost of loop1 and loop2 alone in Fig. 11, since energy estimation includes the cost of sending synchronization message every second. However, in the case of the maximum energy cost, ST costs more than RA, which can be explained by the fact that the node incurs the maximum energy cost due to a high chance of packet loss. In the face of packet loss, the node with the self-triggered transmissions protocol keeps listening at a high energy cost because of its recovery mechanism. Whereas the node with the rate adaptation protocol applies its self-recovery mechanism without extra energy cost. To verify this difference, we analyze the power consumption of two nodes. Node 103 has a higher packet reception ratio than node 124. Fig. 12C shows that self-triggered transmissions are not as efficient as rate adaptation for node 124, due to its recovery mechanism.



(A) energy cost of all WSN nodes

(B) duty cycle of all WSN nodes

(C) energy cost of different nodes

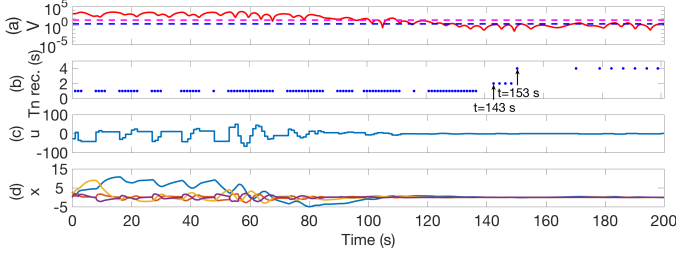
Fig. 12: Energy cost under normal condition.

### C. Network interference

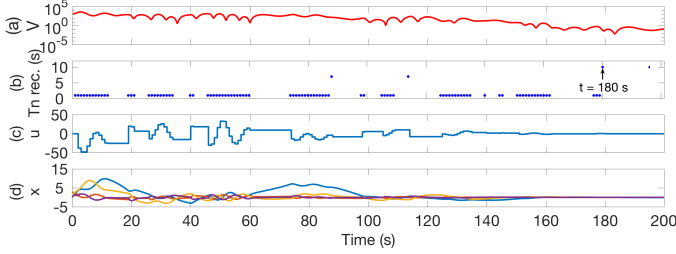
We operate WSN over channel 22 (2.460 GHz) of IEEE.802.15.4, and we introduce network interference by continuously sending jamming packets over an overlapping channel 11 (2.462 GHz) of WiFi. The average packet delivery ratio is reduced to 65.9%. Fig. 13 shows the response curves of RA and ST under network interference. In Plot (b), each dot indicates that the actuator receives a packet. Both methods stay longer at high rate than in normal condition to compensate the impact of interferences. And both the network protocols can recover from packet drops. Fig. 14 shows the statistical results



under network interference. In this case, both RA and ST guarantee the control performance, at the cost of more energy consumption than Sec. VII-B. ST consumes more energy than RA, due to its packet loss recovery mechanism.



(A) time-driven control with rate adaptation



(B) self-triggered control

Fig. 13: Response curve under network interference.

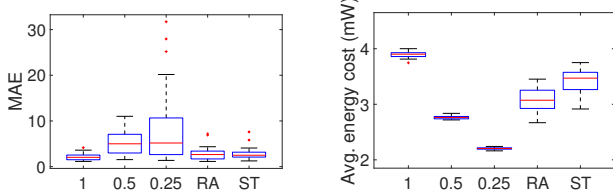


Fig. 14: Performance of double loops under network interference.

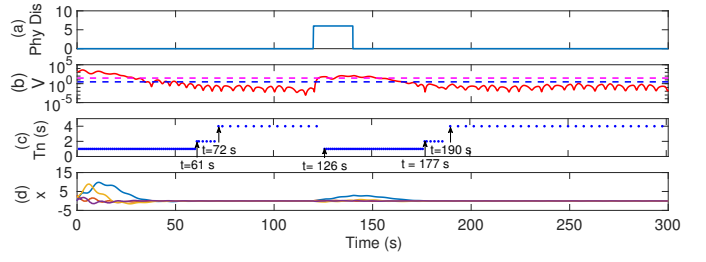
#### D. Physical disturbance

We introduce physical disturbance by adding a constant bias to actuators from 120 s to 140 s. As shown in Fig. 15, both RA and ST adapt rates to 1 Hz under the physical disturbance. However, the time ST ( $t = 130$  s) reacts to the disturbance is later than RA ( $t = 126$  s), since ST has longer  $T_n$  (10 s). Fig. 16 shows the statistical results. As shown in Fig. 16A, both RA and ST have similar MAE with a fixed rate of 1 Hz, and can save more than 30% of the energy. However, as shown in Fig. 16B, the ST performs worse than RA within the interference interval. The longer  $T_n$  (10 s) makes ST response to disturbance slower than time-driven management.

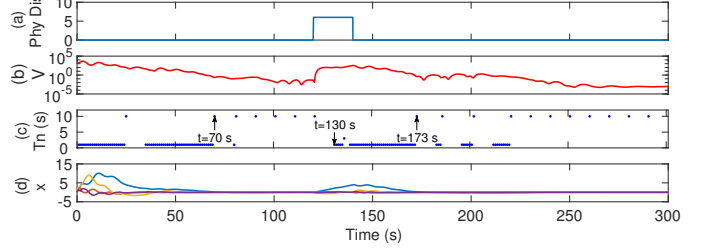
#### E. Both network and physical interferences

By introducing network interference as in Sec. VII-C to the settings in Sec. VII-D, we run experiments with both network and physical interferences. Fig. 17 shows the statistical results that both RA and ST guarantee the control performance at the costs of more energy consumption than in Sec. VII-D. ST costs more energy than RA, due to the recovery mechanism.

To summarize, in normal physical and network condition, RA and ST can achieve similar control performance to a conventional fixed rate of 1 Hz, while improving energy

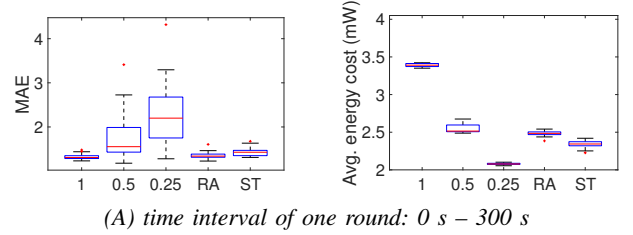


(A) time-driven control with rate adaptation

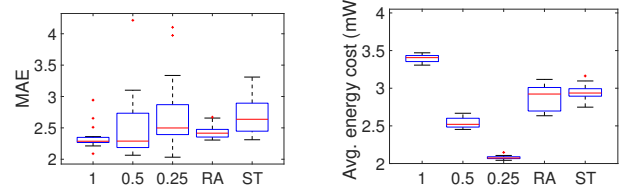


(B) self-triggered control

Fig. 15: Response curve under physical interference.



(A) time interval of one round: 0 s – 300 s



(B) time interval of physical interference: 120s – 180s

Fig. 16: Performance of double loops under physical interference.

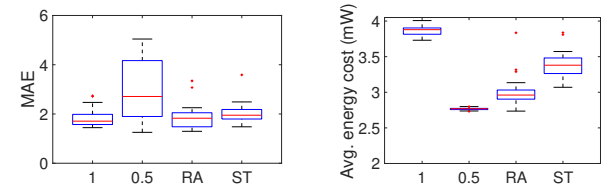


Fig. 17: Performance under network and physical interferences.

efficiency. Besides, ST is more aggressive in energy saving than RA. However, when there are interferences, RA has better performance and energy efficiency than ST, because ST has an embedded recovery mechanism, which costs more energy under packet loss, and a longer inter-transmission interval, which makes ST response slowly to disturbance.

#### F. Scalability and flexibility of RT-WCPS

Although above experimental results are based on the network-in-the-loop simulations of two control loops. RT-WCPS has the scalability to operate more control loops. In

addition, it has the flexibility that end nodes of the data flows can be any nodes in the testbed. As an example, we simulate five control loops sharing a WSN. Loops  $l_1$ ,  $l_3$ , and  $l_5$  control 3 PLANT1s. Loops  $l_2$  and  $l_4$  control 2 PLANT2s. Fig. 7 shows the source and destination pairs of five actuation flows over 3-floor WSN. Table. I shows the MAEs and energy costs in one round (200 s) of network-in-the-loop simulation under normal condition. Loops  $l_1$ ,  $l_3$ , and  $l_5$  have larger MAEs and are more sensitive to different rates than  $l_2$  and  $l_4$ , since  $l_2$  and  $l_4$  with lower mass and damping are easier and faster to stabilize. Although there is some randomness in single simulation, it is obvious that RA and ST can achieve similar control performance with fixed rate of 1 Hz, while save energy for more than 47%.

TABLE I: Performance of five-loop simulation

|           | MAE1   | MAE2   | MAE3   | MAE4   | MAE5   | Energy (mW) |
|-----------|--------|--------|--------|--------|--------|-------------|
| <b>1</b>  | 0.9666 | 0.2891 | 0.9509 | 0.2292 | 0.9630 | 5.2730      |
| <b>2</b>  | 1.2529 | 0.3158 | 1.2800 | 0.2723 | 1.6537 | 3.0461      |
| <b>4</b>  | 1.5129 | 0.3131 | 1.6886 | 0.2701 | 1.8859 | 2.0233      |
| <b>RA</b> | 0.9435 | 0.2623 | 0.9458 | 0.2987 | 0.9671 | 2.7966      |
| <b>ST</b> | 0.9764 | 0.3148 | 1.0243 | 0.3151 | 0.9943 | 2.5209      |

## VIII. CONCLUSIONS

Wireless control faces significant challenges due to data loss and energy constraints in WSNs. In this paper, we present two efficient holistic control designs for industrial process, rate adaptation (RA) and self-triggered control (ST), that can not only ensure control performance under wireless and physical interferences, but also reduce network energy consumption. Furthermore, we design two network reconfiguration mechanisms based on LWB to support RA and ST in multi-hop WSNs. In addition, we build a real-time network-in-the-loop simulation framework which integrates MATLAB/Simulink and a three-floor WSN testbed to experiment with wireless control over real-world WSNs. Our empirical studies show that both RA and ST offer advantages in control performance and energy efficiency. The advantage in energy efficiency of ST, however, diminishes under harsh physical and wireless conditions due to the cost of recovering from data loss and physical disturbances.

## ACKNOWLEDGMENT

This work was sponsored by NSF through grants 1320921 (NeTS) and 1646579 (CPS), and by the Fullgraf Foundation.

## REFERENCES

- [1] PTP protocol. <https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588>.
- [2] Simulink Desktop Real-Time. <https://www.mathworks.com/products/simulink-desktop-real-time.html>.
- [3] Saurabh Amin et al. Safe and secure networked control systems under denial-of-service attacks. In *HSCC*, 2009.
- [4] Adolfo Anta et al. To sample or not to sample: Self-triggered control for nonlinear systems. *TAC*, 2010.
- [5] José Araújo et al. System architectures, protocols and algorithms for aperiodic wireless control systems. *TII*, 2014.
- [6] Dominik Baumann et al. Evaluating low-power wireless cyber-physical systems. *arXiv*, 2018.
- [7] A Celvin et al. How does control timing affect performance. *Control Syst. Mag.*, 2003.
- [8] Burak Demirel et al. Modular Design of Jointly Optimal Controllers and Forwarding Policies for Wireless Control. *TAC*, 2014.
- [9] Mario Di Francesco et al. Reliability and energy-efficiency in ieee 802.15. 4/zigbee sensor networks: an adaptive and cross-layer approach. *IEEE J. Sel. Areas Commun.*, 2011.
- [10] Adam Dunkels et al. Software-based on-line energy estimation for sensor nodes. In *EmNetS*, 2007.
- [11] Federico Ferrari et al. Efficient network flooding and time synchronization with glossy. In *IPSN*, 2011.
- [12] Federico Ferrari et al. Low-power wireless bus. In *Sensys*, 2012.
- [13] Christoforos N Hadjicostis et al. Feedback control utilizing packet dropping network links. In *CDC*, 2002.
- [14] WPMH Heemels et al. Analysis of event-driven controllers for linear systems. *Int J Control*, 2008.
- [15] Timofei Istomin et al. Data prediction+ synchronous transmissions= ultra-low power wireless sensor networks. In *Sensys*, 2016.
- [16] Yu Jiang et al. Optimal codesign of nonlinear control systems based on a modified policy iteration method. *IEEE Trans. Neural Netw. Learn. Syst.*, 2015.
- [17] Philip Levis et al. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In *Sensys*, 2003.
- [18] Bo Li et al. Realistic Case Studies of Wireless Structural Control. In *ICCPs*, 2013.
- [19] Bo Li et al. Wireless Routing and Control: A Cyber-Physical Case Study. In *ICCPs*, 2016.
- [20] Xiangheng Liu et al. Kalman filtering with partial observation losses. In *CDC*, 2004.
- [21] Chenyang Lu et al. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proceedings of the IEEE*, 2016.
- [22] Yehan Ma et al. Holistic cyber-physical management for dependable wireless control systems. *ACM TCPS*, 2018.
- [23] Fabian Mager et al. Feedback control goes wireless: Guaranteed stability over low-power multi-hop networks. *arXiv*, 2018.
- [24] Manuel Mazo et al. On event-triggered and self-triggered control over sensor/actuator networks. In *CDC*, 2008.
- [25] Manuel Mazo et al. An ISS self-triggered implementation of linear controllers. *Automatica*, 2010.
- [26] Manuel Mazo et al. Decentralized event-triggered control over wireless sensor/actuator networks. *TAC*, 2011.
- [27] Miroslav Pajic et al. Closing the loop: A simple distributed method for control over wireless networks. In *IPSN*, 2012.
- [28] Pangun Park et al. Breath: An adaptive protocol for industrial control applications using wireless sensor networks. *TMC*, 2011.
- [29] Pangun Park et al. Wireless network design for control systems: A survey. *IEEE Communications Surveys & Tutorials*, 2017.
- [30] Abusayeed Saifullah et al. Near optimal rate selection for wireless control systems. *ACM TECS*, 2014.
- [31] Chayan Sarkar. Lwb and fs-lwb implementation for sky platform using contiki. *arXiv*, 2016.
- [32] Danbing Seto et al. On task schedulability in real-time control systems. In *RTSS*, 1996.
- [33] Mo Sha et al. Empirical study and enhancements of industrial wireless sensor-actuator network protocols. *IEEE Internet of Things Journal*.
- [34] Mo Sha et al. Implementation and experimentation of industrial wireless sensor-actuator network protocols. In *EWSN*, 2015.
- [35] Yang Shi et al. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *Int J Control*, 2010.
- [36] Vijay Shilpiekandula et al. Load positioning in the presence of base vibrations. In *ACC*, 2012.
- [37] Bruno Sinopoli et al. Kalman Filtering with Intermittent Observations. *TAC*, 2004.
- [38] Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *TAC*, 2007.
- [39] Chengjie Wu et al. Maximizing network lifetime of wireless network under graph routing. In *IoTDI*, 2016.
- [40] Wei Zhang et al. Stability of networked control systems. *IEEE Control Syst. Mag.*, 2001.
- [41] Gang Zhao. Wireless sensor networks for industrial process monitoring and control: A survey. *Network Protocols and Algorithms*, 2011.
- [42] Kemin Zhou et al. *Robust and optimal control*. Prentice hall New Jersey, 1996.
- [43] Marco Zimmerling et al. Adaptive real-time communication for wireless cyber-physical systems. *ACM TCPS*, 2017.