

Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering
Research

Computer Science and Engineering

Report Number: WUCSE-2011-93

2011

Near Optimal Rate Selection for Wireless Control Systems

Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen

With the advent of industrial standards such as WirelessHART, process industries are now gravitating towards wireless control systems. Due to limited bandwidth in a wireless network shared by multiple control loops, it is critical to optimize the overall control performance. In this paper, we address the scheduling-control co-design problem of determining the optimal sampling rates of feedback control loops sharing a WirelessHART network. The objective is to minimize the overall control cost while ensuring that all data flows meet their end-to-end deadlines. The resulting constrained optimization based on existing delay bounds for WirelessHART networks is challenging since it is... **Read complete abstract on page 2.**

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Saifullah, Abusayeed; Wu, Chengjie; Tiwari, Paras Babu; Xu, You; Fu, Yong; Lu, Chenyang; and Chen, Yixin, "Near Optimal Rate Selection for Wireless Control Systems" Report Number: WUCSE-2011-93 (2011). *All Computer Science and Engineering Research*.

https://openscholarship.wustl.edu/cse_research/70

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

Near Optimal Rate Selection for Wireless Control Systems

Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen

Complete Abstract:

With the advent of industrial standards such as WirelessHART, process industries are now gravitating towards wireless control systems. Due to limited bandwidth in a wireless network shared by multiple control loops, it is critical to optimize the overall control performance. In this paper, we address the scheduling-control co-design problem of determining the optimal sampling rates of feedback control loops sharing a WirelessHART network. The objective is to minimize the overall control cost while ensuring that all data flows meet their end-to-end deadlines. The resulting constrained optimization based on existing delay bounds for WirelessHART networks is challenging since it is non-differentiable, non-linear, and not in closed-form. We propose four methods to solve this problem. First, we present a subgradient method for rate selection. Second, we propose a greedy heuristic that usually achieves low control cost while significantly reducing the execution time. Third, we propose a global constrained optimization algorithm using a simulated annealing (SA) based penalty method. Finally, we formulate rate selection as a differentiable convex optimization problem that provides a closed-form solution through a gradient descent method. This is based on a new delay bound that is convex and differentiable, and hence simplifies the optimization problem. We evaluate all methods through simulations based on topologies of a 74-node wireless sensor network testbed. Surprisingly, the subgradient method is disposed to incur the longest execution time as well as the highest control cost among all methods. SA and the greedy heuristic represent the opposite ends of the tradeoff between control cost and execution time, while the gradient descent method hits the balance between the two.

2011-93

Near Optimal Rate Selection for Wireless Control Systems

Authors: Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin

Corresponding Author: saifullaha@cse.wustl.edu

Web Page: <http://www.cse.wustl.edu/~saifullaha/>

Abstract: With the advent of industrial standards such as WirelessHART, process industries are now gravitating towards wireless control systems. Due to limited bandwidth in a wireless network shared by multiple control loops, it is critical to optimize the overall control performance. In this paper, we address the scheduling-control co-design problem of determining the optimal sampling rates of feedback control loops sharing a WirelessHART network. The objective is to minimize the overall control cost while ensuring that all data flows meet their end-to-end deadlines. The resulting constrained optimization based on existing delay bounds for WirelessHART networks is challenging since it is non-differentiable, non-linear, and not in closed-form. We propose four methods to solve this problem. First, we present a subgradient method for rate selection. Second, we propose a greedy heuristic that usually achieves low control cost while significantly reducing the execution time. Third, we propose a global constrained optimization algorithm using a simulated annealing (SA) based penalty method. Finally, we formulate rate selection as a differentiable convex optimization problem that provides a closed-form solution through a gradient descent method. This is based on a new delay bound that is convex and differentiable, and hence simplifies the optimization problem. We evaluate all methods through simulations based on topologies of a 74-node wireless sensor network testbed. Surprisingly, the subgradient

Type of Report: Other

Near Optimal Rate Selection for Wireless Control Systems

Abusayeed Saifullah, Chengjie Wu, Paras Babu Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen
Department of Computer Science and Engineering, Washington University in St. Louis
{saifullah, wuchengjie, pbtwari, youxu, yong.fu, lu, ychen25}@wustl.edu

Abstract—With the advent of industrial standards such as WirelessHART, process industries are now gravitating towards wireless control systems. Due to limited bandwidth in a wireless network shared by multiple control loops, it is critical to optimize the overall control performance. In this paper, we address the scheduling-control co-design problem of determining the optimal sampling rates of feedback control loops sharing a WirelessHART network. The objective is to minimize the overall control cost while ensuring that all data flows meet their end-to-end deadlines. The resulting constrained optimization based on existing delay bounds for WirelessHART networks is challenging since it is non-differentiable, non-linear, and not in closed-form. We propose four methods to solve this problem. First, we present a subgradient method for rate selection. Second, we propose a greedy heuristic that usually achieves low control cost while significantly reducing the execution time. Third, we propose a global constrained optimization algorithm using a simulated annealing (SA) based penalty method. Finally, we formulate rate selection as a differentiable convex optimization problem that provides a closed-form solution through a gradient descent method. This is based on a new delay bound that is convex and differentiable, and hence simplifies the optimization problem. We evaluate all methods through simulations based on topologies of a 74-node wireless sensor network testbed. Surprisingly, the subgradient method is disposed to incur the longest execution time as well as the highest control cost among all methods. SA and the greedy heuristic represent the opposite ends of the tradeoff between control cost and execution time, while the gradient descent method hits the balance between the two.

I. INTRODUCTION

With the advent of industrial wireless standards such as WirelessHART [1], recent years have seen successful real-world deployments of process control systems over wireless sensor-actuator networks (WSANs). In a wireless control system, the control performance not only depends on the design of control algorithms, but also relies on real-time communication over the shared wireless network. The choice of sampling rates of the feedback control loops must balance between control performance and real-time communication. A low sampling rate usually degrades the control performance while a high sampling rate may cause excessive communication delays causing degraded performance. The coupling between real-time communication and control requires a scheduling-control co-design approach to optimize the control performance subject to stringent bandwidth constraints of the wireless network.

In this paper, we address the sampling rate optimization problem for multiple feedback control loops sharing a WirelessHART network. A *feedback control loop* periodically

delivers data from sensors to the controller, and then delivers the control messages to the actuators through the network. We consider a wireless control system wherein transmissions over a multi-hop WSAN are scheduled based on fixed priorities. The objective is to determine the optimal sampling rates of the feedback control loops to minimize their total control cost, subject to the constraints that their end-to-end network delays are within their respective sampling periods. To our knowledge, this is the first work on scheduling-control co-design for WirelessHART networks.

We formulate the sampling rate optimization problem based on existing end-to-end delay bounds [2], [3] for data flows in multi-hop WirelessHART mesh networks. The resulting constrained non-linear optimization problem is challenging because the existing delay bounds are *non-differentiable* and *not in closed-form*. To address this difficult scheduling-control co-design problem in wireless control systems based on WirelessHART networks, we study and propose four methods:

- First, to handle non-differentiability and non-convexity of the delay bounds, we develop a subgradient based method to find sampling rates through Lagrangian relaxation.
- Second, we propose an efficient polynomial time greedy heuristic that usually achieves low control cost, and is suitable for large-scale WSANs and online rate selection.
- Third, we propose a global constrained optimization algorithm that adopts a penalty approach based on simulated annealing (SA).
- Finally, we derive a convex and differentiable delay bound by relaxing an existing delay bound. Then, we formulate the co-design as a differentiable convex optimization problem and, thus, provide a closed-form solution for rate selection through a gradient descent method.

We evaluate the proposed algorithms through simulations based on the real network topologies of a wireless sensor network testbed of 74 TelosB motes. The results demonstrate that, among all methods, SA achieves the least control cost while requiring the longest execution time. In contrast, the greedy heuristic runs faster but leads to higher control cost. The gradient descent method based on the new delay bound hits the balance between control cost and execution time. Interestingly, due to high nonlinearity and existence of a large number of local extrema, the subgradient method is both ineffective and inefficient.

In the rest of the paper, Section II reviews related works.

Section III presents the network model. Section IV describes the control loop model. Section V formulates the rate selection problem. Sections VI, VII, and VIII present the subgradient method, the greedy heuristic method, and the SA based penalty method, respectively, for rate selection. Section IX derives a convex delay bound and presents the gradient descent method for rate selection. Section X presents evaluation results. Section XI concludes the paper.

II. RELATED WORKS

There have been extensive studies on real-time CPU scheduling and control co-design in single-processor systems (see survey [4]). Some notable works [5]–[7] among them address rate selection under schedulability constraints. However, these works do not apply for networked control systems since network induced delays have significant effects on control performance, and the schedulability analysis through the network is usually more complicated than CPU scheduling. Following the seminal work on integrated communication and control [8], a number of works [9]–[15] have treated the co-design in networked control systems. However, these works have not been designed for wireless networks where end-to-end delay analysis introduces challenging non-linear optimization problems.

For wireless control system, a conceptual study of a wireless real-time system dedicated for remote sensor/actuator control in production automation has been presented in [16]. Wireless control co-design has been studied in [17]–[19]. But these works do not consider multi-hop wireless networks. The rate selection under schedulability constraints for multi-hop wireless sensor network (WSN) has been studied in [20], [21]. But these works consider a simplified network model where a WSN is cellular with a base station functioning as a router at the center of each cell. An inner cell is surrounded by 6 cells. The base station in a cell uses 7 orthogonal channels for communication with 6 surrounding cells, periodically enabling transmission in each direction. The utilization based analysis used for this model does not apply for common WSNs based on industrial standards such as WirelessHART. To our knowledge, there exists no utilization based schedulability analysis for multi-hop wireless networks. This lack of simple analytical model to efficiently analyze real-time performance excludes the use of scheduling-control co-design approaches developed for CPU scheduling or wired networks.

As WirelessHART networks [1], [22] are becoming the mainstream for wireless control systems in process industries, recent works have focused on control and scheduling issues in WirelessHART networks [2], [23]–[29]. However, these works have addressed either scheduling [24]–[27], [29], routing [28], delay analysis [2], or framework to model schedules [23], and have not considered the scheduling-control co-design problems such as rate selection. In contrast, we have developed the co-design approach to determine near optimal sampling rates of the feedback control loops which minimize their overall control cost and ensure their real-time schedulability. To our

knowledge, this paper is the first to address scheduling-control co-design for WirelessHART networks.

III. CONTROL NETWORK MODEL

We consider a wireless control system where feedback control loops are closed over a WirelessHART network. The WirelessHART standard [1], [22] has been specifically designed to meet the critical needs for industrial process monitoring and control. We consider a WirelessHART network consisting of a set of field devices (sensors and actuators) and one gateway. A WirelessHART network is characterized by small size and a *centralized network manager* installed in the gateway. The network manager determines the routes, and schedule of transmissions. The controllers for feedback control loops are installed in the gateway. The sensor devices deliver their sensor data to the controllers, and the control messages are then delivered to the actuators through the network.

Time is synchronized, and transmissions happen based on TDMA. A time slot is 10ms long, and allows exactly one transmission and its acknowledgement between a device pair. In a *dedicated slot*, there is only one sender for each receiver. In a *shared slot*, more than one sender can attempt to transmit to the same receiver. The network uses 16 channels defined in IEEE 802.15.4 and allows per time slot channel hopping. Each transmission in a time slot happens on a different channel. A device cannot both transmit and receive at the same time; nor can it receive from more than one sender at the same time. Two transmissions *conflict* when they involve a common node.

A directed list of paths that connect a source and destination pair is defined as a *routing graph*. For communication between a pair, transmissions are scheduled on the routing graph by allocating one link for each en-route device starting from the source, followed by allocating a second dedicated slot on the same path to handle a retransmission, and then by allocating a third shared slot on a separate path to handle another retry. This conservative practice leaves a huge number of allocated time slots unused since only one route is chosen based on network conditions, thereby degrading the schedulability. To address this, existing end-to-end delay analysis [2], [3] considers only collision-free schedule based on dedicated slots. Since delay analysis is not the focus of this paper, we use existing end-to-end delay bounds. If, in the future, any delay bound is derived by considering shared time slots, that bound can be applied to define the constraints in our co-design problem.

IV. CONTROL LOOP MODEL

The wireless control system consists of n feedback control loops, each denoted by F_i , $1 \leq i \leq n$. Associated with each control loop are a sensor node and an actuator. In each loop, the dynamics of the plant is described as a Linear Time Invariant (LTI) system and can be written as

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

where $x(t)$ is the plant state, $u(t)$ is the controller output, $y(t)$ is the system output. A , B , C are constant matrix describing

the system dynamics. Although we assume LTI in this work, the framework proposed can be extended to time-varying and/or non-linear systems. For each loop, we consider the state feedback controller:

$$u(t) = Lx(t)$$

where L is control gain designed by the control theory. The *quality of control* (QoC) of each loop is measured by the following performance index function [5]:

$$J(u) = \lim_{H \rightarrow \infty} \int_0^H (x^T(t)Qx(t) + u^T(t)Wu(t)) dt$$

Where Q and W are quadratic weight matrix representing the importance of deviation of control objective $x(t)$ and control effort $u(t)$ ¹. H is the time horizon during which the cost function is calculated. A great value of $J(u)$ thus indicates either a great deviation of the desired state or a great control effort to bring the state to its reference value. An optimal control theory, such as Least Quadratic Regulator (LQR), solves the optimization problem:

$$J^*(u) = \text{minimize } J(u)$$

to derive an optimal controller. Although $J(u)$ is often related to an optimal control problem, it can be used as a general control performance index not limited to some specific controller.

Considering the digital implementation of a control loop F_i , the optimal control performance may deviate from its continuous counterpart J_i^* respecting the sampling frequency f_i (Hz). Usually, there is complicated interaction between the deviation and the sampling frequency. However, similar to [5], the deviation with respect to the sampling frequency can be approximated as follows

$$J_i = J_{D,i}^* - J_i^* = \alpha_i e^{-\beta_i f_i} \quad (1)$$

where $J_{D,i}^*$ is the optimal control performance of the digital implementation, α_i is the magnitude coefficient, and β_i is the decay rate.

Each control loop F_i maintains a minimum required frequency of f_i^{\min} Hz and a maximum allowable frequency of f_i^{\max} Hz. To maintain an acceptable control performance, the end-to-end communication (sensor-controller-actuator) delay for every sensor data and its associated control message must remain within the sampling period T_i . For any control loop F_i , we express its sampling period T_i in terms of time slots. Since 1 slot=10ms, its sampling *rate* or *frequency* is

$$f_i = \frac{100}{T_i} \text{ Hz}$$

Transmissions are scheduled on m channels, and using *rate monotonic* policy where a loop with higher rate has higher priority, breaking ties arbitrarily. The set of control loops $F = \{F_1, F_2, \dots, F_n\}$ will always be assumed to be ordered by priorities. F_h has higher priority than F_i if and only if $h < i$. That is, each F_h , $1 \leq h \leq i - 1$, is

a higher priority loop of F_i . In a *fixed priority scheduling policy*, among all transmissions that can be scheduled in a time slot, the one belonging to the highest priority control loop is scheduled on an available channel first. The complete schedule is divided into superframes. A *superframe* represents transmissions in a series of time slots that repeat infinitely and represent the communication pattern of a group of devices. In rate monotonic scheduling, flows having the same period are assigned in the superframe of length equal to their period. We will use C_i to denote the number of transmissions (i.e., time slots) required by F_i for end-to-end communication. The end-to-end delay for F_i is denoted by R_i (time slots). The set of control loops F is *schedulable*, if $R_i \leq T_i, \forall 1 \leq i \leq n$.

V. FORMULATION OF THE RATE SELECTION PROBLEM

In this section, we formulate the rate selection problem as a constrained non-linear optimization problem. The objective is to minimize the overall control cost of the feedback control loops subject to their real-time schedulability constraints. Based on the selected rates, the control loops are scheduled using rate monotonic policy.

In order to capture the online interaction between control algorithms and the scheduler, a number of issues must be considered. It must be possible to dynamically adjust the control loop parameters, e.g., their rates, in order to compensate for changes in the workload. It can also be advantageous to view this parameter adjustment strategy in the scheduler as a controller. Control design methods must also take the schedulability constraints into account to guarantee real-time communication through the network. Besides, it should be possible to compensate for wireless deficiencies (e.g., lossy links). Briefly, there are three main factors that affect coupling between the control system and wireless network: (1) the rates of the control loops, (2) the end-to-end delays, and (3) the packet loss. As explained in Section III, a packet delivery in WirelessHART networks achieves high degree of reliability through route and spectrum diversity. As a consequence, the probability of packet loss is very low [22]. Therefore, our co-design approach focuses on rates and end-to-end delays.

We use the end-to-end delay bounds derived in [3] which are an improved and extended analysis proposed in [2]. In fact, the analysis in [3] has two ways to derive a delay bound: in pseudo-polynomial time and in polynomial time. Note that a pseudo-polynomial time bound makes the schedulability constraints extremely expensive to check at every step of optimization in the co-design, thereby making a non-linear optimization approach almost impractical. Therefore, in this section, we formulate the problem using the polynomial-time delay bounds that are somewhat less precise than pseudo-polynomial ones. In the polynomial time analysis, the worst case end-to-end delay R_i of F_i is determined as follows

$$R_i = \left\lfloor \frac{1}{m} \sum_{h=1}^{i-1} \Omega_i^h \right\rfloor + \sum_{h=1}^{i-1} \Theta_i^h + C_i \quad (2)$$

Where m denotes the total number of channels; Ω_i^h is the delay that a higher priority loop F_h causes on F_i due to *channel*

¹We assume the system converges to the origin.

contention, and is determined as follows

$$\Omega_i^h = \min \left(T_i - C_i + 1, \left\lfloor \frac{T_i + T_h - C_h}{T_h} \right\rfloor C_h \right. \\ \left. + \min \left(C_h, T_i + T_h - C_h - \left\lfloor \frac{T_i + T_h - C_h}{T_h} \right\rfloor T_h \right) \right)$$

And Θ_i^h is the delay that a higher priority loop F_h can cause on F_i due to *transmission conflict*, and is determined as follows

$$\Theta_i^h = \Delta_i^h + \left(\left\lfloor \frac{T_i}{T_h} \right\rfloor - 1 \right) \delta_i^h + \min \left(\delta_i^h, T_i - \left\lfloor \frac{T_i}{T_h} \right\rfloor T_h \right)$$

where δ_i^h denotes the maximum delay that a single transmission of F_i can suffer from F_h , and Δ_i^h denotes the total maximum delay that all transmissions of F_i can suffer from F_h , $1 \leq h < i$, due to transmission conflict. These values are calculated based on how the routes of F_i and F_h intersect each other. For any given routes of F_i and F_h , δ_i^h and Δ_i^h are constant, and their derivation can be found in [2], [3].

We now define the performance index of the control system that can describe how the control performance depends on the rates and delays of the control loops. Note that, when the controller is implemented, the system performance will deviate from the ideal value of the performance measure attained using continuous-time control, and the deviation will depend on the sampling rate. As mentioned in Equation 1 like [5], we quantify this deviation by defining the control cost for every control loop F_i by a monotonic and convex function

$$J_i = \alpha_i e^{-\beta_i f_i} \quad (3)$$

where α_i is the magnitude co-efficient, β_i is the decay rate, f_i (in Hz) is the rate of F_i . Considering w_i as the weight of F_i , for a set of chosen rates $f = \{f_1, f_2, \dots, f_n\}$, where f_i is the rate of F_i , the total control cost of the system stands

$$J(f) = \sum_{i=1}^n w_i \alpha_i e^{-\beta_i f_i} \quad (4)$$

Function $J(f)$ describes how the control performance depends on the rates (i.e., frequencies) and delays of the control loops. Namely, the higher the rates, the better the performance. However, a too high rate of some loop may cause congestion in the network, resulting in a very low rate for some other loop, thereby degrading the performance. Therefore, we choose the total control cost $J(f)$ as the *performance index*.

We can now formulate the scheduling-control co-design as a non-linear constrained optimization problem, where our objective is to determine the optimal sampling rates that minimize the total control cost. The co-design must guarantee that the end-to-end delay R_i of every loop F_i is within its deadline T_i . Besides, every control loop F_i must maintain its minimum required rate of f_i^{\min} Hz and the maximum allowable rate of f_i^{\max} Hz for an acceptable control performance. In the scheduling-control co-design, our objective thus boils down

to finding rates $f = \{f_1, f_2, \dots, f_n\}$ so as to

$$\begin{aligned} & \text{minimize } J(f) \\ & \text{subject to } R_i \leq T_i, \forall 1 \leq i \leq n \\ & \quad f_i \geq f_i^{\min}, \forall 1 \leq i \leq n \\ & \quad f_i \leq f_i^{\max}, \forall 1 \leq i \leq n \end{aligned} \quad (5)$$

where $f_i = 100/T_i$ Hz, and R_i is as defined in Equation 2, $\forall 1 \leq i \leq n$.

VI. SUBGRADIENT METHOD FOR RATE SELECTION

Subgradient based methods are an established and standard approach for nonlinear optimization. In this section, we develop a subgradient based approach to determine the sampling rates for control cost optimization in the scheduling-control co-design formulated in the previous section.

In the optimization problem defined in 5 for co-design, the objective function $J(f) : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex while the non-linear constraints $R_i \leq T_i, \forall 1 \leq i \leq n$, are not convex. This optimization problem is challenging since the constraints $R_i \leq T_i$ are not differentiable, making any traditional gradient-based optimization unsuitable. To generate approximate solutions to the primal problem defined in 5, we consider approximate solutions to its dual problem. Here, the dual problem is the one arising from Lagrangian relaxation of the inequality constraints $R_i \leq T_i$, and is given by

$$\begin{aligned} & \text{maximize } L(f, \lambda) \\ & \text{subject to } \lambda \geq 0 \end{aligned} \quad (6)$$

where $L(f, \lambda)$ is the *Lagrangian dual function* defined by

$$L(f, \lambda) = \inf \left\{ J(f) + \sum_{i=1}^n \lambda_i (R_i - T_i) \right\} \\ \text{such that } f_i^{\min} \leq f_i \leq f_i^{\max}, \forall 1 \leq i \leq n$$

Here $\lambda \in \mathbb{R}^n$ is the vector of *Lagrange multipliers*.

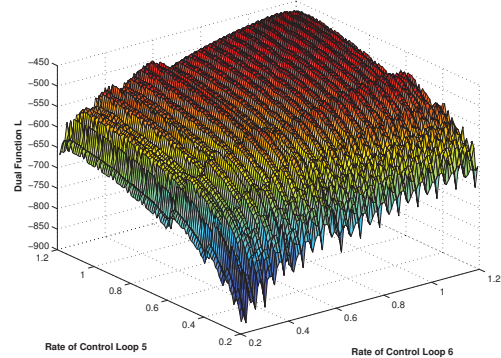


Fig. 1. Surface of the dual function in 6

Figure 1 shows the surface of the dual problem in 6 for changing the rates of 2 control loops (and keeping all other loops' rates unchanged) considering data flows of 12 control loops simulated on our WSN testbed topology (shown in Figure 4). The figure shows that $L(f, \lambda)$ is highly nonlinear

in rates. This implies the difficulty of the problem. Besides, the function $L(f, \lambda)$ is not differentiable everywhere. Therefore, traditional optimization approaches based on gradient calculation cannot be applied directly to solve it. Hence, we first adopt a subgradient optimization method to determine the rates. Note that f belongs to a finite range. Steps of the subgradient method are presented as Algorithm 1.

Algorithm 1: Subgradient Method for Rate Selection

Input: $[f_i^{\min}, f_i^{\max}], w_i, \alpha_i, \beta_i \forall F_i, 1 \leq i \leq n$;
Output: $f_i, \forall F_i$, and total control cost J ;
 $f_i \leftarrow f_i^{\min}, \forall F_i, 1 \leq i \leq n$; /* validity check */
Assign priorities using rate monotonic policy;
if $\exists F_i$ such that $R_i > T_i$ **then return unschedulable**;
Step 0: Set time $t = 0$. Choose initial Lagrange multipliers $\lambda^t = 0$. Let f^t be the primal variables corresponding to Lagrange multipliers λ^t .
while stop condition not true **do**
Step 1: Determine the rate monotonic priorities of the loops under current f . Solve the Lagrangian subproblem $L(f, \lambda)$. That is, given the dual variables λ^t , determine the primal variables f^t as follows

$$f^t \in \arg \min \left\{ J(f) + \sum_{i=1}^n \lambda_i (R_i - T_i) \right\}$$

such that $f^{\min} \leq f^t \leq f^{\max}$

This gives a subgradient $s^t = R^t - T^t$ at λ^t . If $s^t = 0$, then stop. The algorithm has converged. Current λ^t gives the optimal value of the dual, and current f^t gives an approximated value of the primal.
Step 2: Compute the Lagrange multipliers for next time as follows

$$\lambda^{t+1} = \max\{0, \lambda^t + \gamma^t s^t\}$$

where γ^t is the step size.
Step 3: Update $t = t + 1$ and go to Step 1.
end

Thus, the scheduling-control co-design defined in 5 can be solved using any existing subgradient solver (e.g., SSMS [30]). Both the speed of convergence and the quality of solution largely depend on the step size selection. As a traditional subgradient method, Algorithm 1 is guaranteed to converge under any diminishing step size or dynamically adjusted step size such as Polyak step size [31].

VII. GREEDY HEURISTIC FOR RATE SELECTION

While a subgradient method is a standard approach for non-linear optimization, it can run very slowly for many practical problems that have too many local extrema and are highly non-linear. Due to a large number of local extrema and complicated subgradient direction in our optimization problem, the subgradient based method proposed in the previous section for rate selection may turn out to be not quite efficient. Therefore, in this section, we propose a simple intuitive greedy heuristic that can scale very well. It runs in polynomial time.

The greedy heuristic starts by selecting a rate of f_i^{\min} for each control loop F_i . Note that, for valid rate ranges $[f_i^{\min}, f_i^{\max}]$, the control loops should be schedulable when each loop F_i selects a rate of f_i^{\min} . Otherwise, the test case is simply rejected since no rate selection exists that can satisfy the schedulability constraints. For valid rate ranges,

Algorithm 2: Greedy Heuristic

Input: $[f_i^{\min}, f_i^{\max}], w_i, \alpha_i, \beta_i \forall F_i, 1 \leq i \leq n$, and a step size μ ;
Output: $f_i, \forall F_i$, and total control cost J ;
 $f_i \leftarrow f_i^{\min}, \forall F_i, 1 \leq i \leq n$; /* initialize rates */
Assign priorities using rate monotonic policy;
if $\exists F_i$ such that $R_i > T_i$ **then return unschedulable**;
while true **do**
 $max \leftarrow 0$; /* maximum control cost decrease */
 $k \leftarrow null$; /* index of the best control loop */
for each $F_i, i = 1, 2, \dots, n$ such that f_i can further increase **do**
 $J_i^{\text{old}} \leftarrow w_i \alpha_i e^{-\beta_i f_i}$; /* current cost of F_i */
 $f_i \leftarrow f_i + \mu$; /* increase rate by μ */
Reassign priorities using rate monotonic policy;
if $R_j \leq T_j, \forall 1 \leq j \leq n$ **then** /* if schedulable */
 $J_i^{\text{new}} \leftarrow w_i \alpha_i e^{-\beta_i f_i}$; /* new cost of F_i */
if $J_i^{\text{old}} - J_i^{\text{new}} \geq max$ **then**
 $max \leftarrow J_i^{\text{old}} - J_i^{\text{new}}$;
 $k \leftarrow i$; /* F_k is the best candidate */
end
end
 $f_i \leftarrow f_i - \mu$; /* put back F_i 's rate */
end
if $max=0$ **then** /* no f_i can further increase */
| **return** current $f_i, \forall F_i$, and total control cost J
end
 $f_k \leftarrow f_k + \mu$; /* increase rate of loop F_k */
end

the algorithm has the highest control cost in the beginning. Therefore, it will keep decreasing the cost as long the loops are schedulable. This is done by increasing the sampling rates of the loops. The algorithm selects one control loop to increase the rate in each step, and uses a step size of μ by which the rate is increased. For loop F_i , the decrease in control cost due to an increase in current rate f_i by μ is determined as

$$w_i \alpha_i e^{-\beta_i f_i} - w_i \alpha_i e^{-\beta_i (f_i + \mu)}$$

In every step, the greedy heuristic increases the rate of the control loop that decreases the control cost most while satisfying the schedulability constraints of the loops. It keeps increasing the rates in this way as long as some loop's rate can be increased while keeping all loops schedulable. When no loop's rate can be increased anymore, the algorithm terminates, and returns the current control cost J , and the selected rates. The pseudo code of the greedy heuristic method is presented as Algorithm 2.

VIII. RATE SELECTION USING A PENALTY APPROACH WITH SIMULATED ANNEALING

The greedy heuristic proposed in the previous section can execute very fast and, in some cases, may significantly minimize the control cost. But due to complicated nonlinear constraints, in many cases, it can get stuck in local extrema and, hence, its performance (in terms of control cost) may not be guaranteed. Therefore, in this section, we explore a global optimization framework based on simulated annealing that can handle non-differentiability and escape local extrema. In particular, we propose a method that extends the standard simulated annealing through a penalty approach to address the constraints for rate selection.

Simulated annealing (SA) is a global optimization framework that is suitable for problems where gradient information is not available. It uses a global parameter called *temperature* to control the probability of accepting a new solution that is worse than the current one. The temperature decreases gradually as the algorithm gradually converges. SA is proven to be able to achieve global optimality under certain theoretical conditions. SA is particularly suitable for our problem since it does not require differentiability of functions, and it employs stochastic global exploration to escape from local minima.

However, while the original SA is designed for unconstrained optimization, our co-design problem is a constrained optimization problem. To find a feasible solution using SA for our co-design problem, we use a ℓ_1 -penalty method [32]. In this method, we introduce a new objective function

$$g = J(x) + pV(x),$$

where J is the control cost, $V = \max\{0, R_i - T_i | i = 1 \dots n\}$ is the *violation of schedulability constraints*, and $p > 0$ is the *penalty factor*. The penalty method starts with a low penalty 0.25 and an initial temperature set to $1000*n$, where n is the number of control loops.

At each iteration, we use SA to minimize g under a fixed p . If it cannot find a feasible solution with that setting, we increase the penalty p and temperature and start over the SA algorithm. Theoretically, such a penalty method can find the constrained global optimal solution when the unconstrained optimization is optimal and p is large enough. The new penalty at the i^{th} iteration is calculated by multiplying p at the $(i-1)^{th}$ iteration by four, and the new temperature is calculated by multiplying the original temperature by the iteration number i . This process is continued until we find a feasible solution or the maximum number of iteration is reached. The maximum number of iteration is currently set to 100. In all SA experiments, we set the final temperature and total number of steps to be 0.01 and 200,000, respectively.

IX. RATE SELECTION THROUGH CONVEX OPTIMIZATION

The co-design problem in 5 does not have a closed form solution. Since it is non-differentiable and non-convex, we have adopted subgradient method and simulated annealing to solve it. In this section, we derive a differentiable and convex delay bound by relaxing the pseudo-polynomial time delay bound proposed in [2], [3]. Then, we formulate the rate selection problem as a convex optimization problem. The advantage of such formulation is that it has a closed form solution, and can be solved through a gradient descent method.

For each loop F_i , we derive a differentiable and convex delay bound R_i^{cvx} as follows. Based on the pseudo-polynomial time analysis in [2], [3], if loop F_i has an end-to-end delay of x time slots, the channel contention delay Ω_i^h that a higher priority loop F_h can cause on F_i is bounded as follows

$$\Omega_i^h \leq \left\lfloor \frac{x}{T_h} \right\rfloor C_h + C_h + (C_h - 1) \leq \frac{x}{T_h} C_h + 2C_h - 1$$

Similarly, the transmission conflict delay Θ_i^h that a higher priority loop F_h can cause on F_i is bounded as follows

$$\begin{aligned} \Theta_i^h &= \Delta_i^h + \left(\left\lfloor \frac{x}{T_h} \right\rfloor - 1 \right) \delta_i^h + \min \left(\delta_i^h, x - \left\lfloor \frac{x}{T_h} \right\rfloor T_h \right) \\ &\leq \Delta_i^h + \left(\frac{x}{T_h} - 1 \right) \delta_i^h + \delta_i^h = \Delta_i^h + \frac{x}{T_h} \delta_i^h \end{aligned}$$

Note that the above upper bounds of Θ_i^h and Ω_i^h are both differentiable and continuous. If a control loop F_i has an end-to-end delay of x time slots, then using the above upper bounds of Θ_i^h and Ω_i^h , the end-to-end delay bound x can be written similar to Equation 2 as follows

$$\begin{aligned} x &= \frac{1}{m} \sum_{h=1}^{i-1} \left(\frac{x}{T_h} C_h + 2C_h - 1 \right) + \sum_{h=1}^{i-1} \left(\Delta_i^h + \frac{x}{T_h} \delta_i^h \right) + C_i \\ &= \frac{x}{m} \sum_{h=1}^{i-1} \frac{C_h}{T_h} + \frac{1}{m} \sum_{h=1}^{i-1} (2C_h - 1) + \sum_{h=1}^{i-1} \Delta_i^h + x \sum_{h=1}^{i-1} \frac{\delta_i^h}{T_h} \\ &\quad + C_i \\ \Leftrightarrow x &\left(1 - \frac{1}{m} \sum_{h=1}^{i-1} \frac{C_h}{T_h} - \sum_{h=1}^{i-1} \frac{\delta_i^h}{T_h} \right) \\ &= \frac{1}{m} \sum_{h=1}^{i-1} (2C_h - 1) + \sum_{h=1}^{i-1} \Delta_i^h + C_i \\ \text{Thus, } x &= \frac{\frac{1}{m} \sum_{h=1}^{i-1} (2C_h - 1) + \sum_{h=1}^{i-1} \Delta_i^h + C_i}{1 - \frac{1}{m} \sum_{h=1}^{i-1} \frac{C_h}{T_h} - \sum_{h=1}^{i-1} \frac{\delta_i^h}{T_h}} = R_i^{cvx} \quad (7) \end{aligned}$$

Lemma 1: For any control loop F_i , the end-to-end delay bound R_i^{cvx} derived in Equation 7 is convex in f .

Proof: Note that R_i^{cvx} is twice-differentiable. Hence R_i^{cvx} is convex iff its Hessian matrix is positive semidefinite. Let the constant (the numerator in R_i^{cvx}): $\frac{1}{m} \sum_{h=1}^{i-1} (2C_h - 1) + \sum_{h=1}^{i-1} \Delta_i^h + C_i = Q_i$. Using $T_i = 100/f_i$ the denominator of R_i^{cvx} : $1 - \frac{1}{100m} \sum_{h=1}^{i-1} f_h C_h - \frac{1}{100} \sum_{h=1}^{i-1} f_h \delta_i^h = Z_i$. Letting $\frac{C_h}{100m} + \frac{\delta_i^h}{100} = q_h$, for $h = 1, 2, \dots, i-1$, the gradient is given by

$$\nabla R_i^{cvx}(f_1, f_2, \dots, f_{i-1}) = \begin{pmatrix} \frac{Q_i}{Z_i^2} q_1 \\ \frac{Q_i}{Z_i^2} q_2 \\ \vdots \\ \frac{Q_i}{Z_i^2} q_{i-1} \end{pmatrix}$$

The Hessian matrix H is given by: $H =$

$$\begin{bmatrix} 2 \frac{Q_i}{Z_i^3} q_1^2 & 2 \frac{Q_i}{Z_i^3} q_1 q_2 & 2 \frac{Q_i}{Z_i^3} q_1 q_3 & \cdots & 2 \frac{Q_i}{Z_i^3} q_1 q_{i-1} \\ 2 \frac{Q_i}{Z_i^3} q_2 q_1 & 2 \frac{Q_i}{Z_i^3} q_2^2 & 2 \frac{Q_i}{Z_i^3} q_2 q_3 & \cdots & 2 \frac{Q_i}{Z_i^3} q_2 q_{i-1} \\ 2 \frac{Q_i}{Z_i^3} q_3 q_1 & 2 \frac{Q_i}{Z_i^3} q_3 q_2 & 2 \frac{Q_i}{Z_i^3} q_3^2 & \cdots & 2 \frac{Q_i}{Z_i^3} q_3 q_{i-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 \frac{Q_i}{Z_i^3} q_{i-1} q_1 & 2 \frac{Q_i}{Z_i^3} q_{i-1} q_2 & 2 \frac{Q_i}{Z_i^3} q_{i-1} q_3 & \cdots & 2 \frac{Q_i}{Z_i^3} q_{i-1}^2 \end{bmatrix}$$

Note that $\frac{Q_i}{Z_i} > 0$, $q_h > 0, \forall h$. Now the leading principal minors of H :

$$\begin{aligned} & \left| 2 \frac{Q_i}{Z_i} q_1^2 \right| > 0, & \begin{vmatrix} 2 \frac{Q_i}{Z_i} q_1^2 & 2 \frac{Q_i}{Z_i} q_1 q_2 \\ 2 \frac{Q_i}{Z_i} q_2 q_1 & 2 \frac{Q_i}{Z_i} q_2^2 \end{vmatrix} = 0, \\ & \begin{vmatrix} 2 \frac{Q_i}{Z_i} q_1^2 & 2 \frac{Q_i}{Z_i} q_1 q_2 & 2 \frac{Q_i}{Z_i} q_1 q_3 \\ 2 \frac{Q_i}{Z_i} q_2 q_1 & 2 \frac{Q_i}{Z_i} q_2^2 & 2 \frac{Q_i}{Z_i} q_2 q_3 \\ 2 \frac{Q_i}{Z_i} q_3 q_1 & 2 \frac{Q_i}{Z_i} q_3 q_2 & 2 \frac{Q_i}{Z_i} q_3^2 \end{vmatrix} = 0, \dots, & |H| = 0. \end{aligned}$$

Thus all leading principle minors become non-negative. Therefore, Hessian matrix H is positive semidefinite. Hence, R_i^{cvx} is convex in f . ■

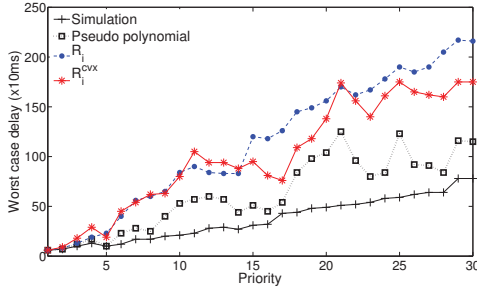


Fig. 2. End-to-end delay bounds on testbed topology

Figure 2 shows how (pessimistic) the derived convex bound R_i^{cvx} is for a test case on our WSN testbed topology (Figure 4). The simulation generates data flows for 30 control loops in the network and randomly assigns, for each loop, a harmonic period that is also a multiple of 10ms (i.e., 1 time slot) in a range $[320ms, 5120ms]$. The loops are assigned rate monotonic priority, and are sorted along the X -axis from the highest to the lowest priority. Using 12 channels, the delay bounds R_i (Equation 2), R_i^{cvx} , and the delay bound based on the pseudo-polynomial time analysis in [3] are shown in the figure for each loop F_i . The loops are scheduled up to their hyper-period, and for each loop, its maximum end-to-end delay observed in simulations (marked by ‘simulation’) is also shown. The figure indicates that R_i^{cvx} overestimates the delay at most 2 times that estimated by the pseudo-polynomial analysis. R_i^{cvx} is also highly competitive against the polynomial time delay bound R_i . Since, neither R_i^{cvx} nor R_i dominates the other, we study the results under both bounds. The advantage with R_i^{cvx} is that we can get a closed form solution through a gradient descent method.

Now we reformulate the optimization problem in 5 using above expression of R_i^{cvx} as follows. Here, we have to select rates $f = \{f_1, f_2, \dots, f_n\}$ so as to

$$\begin{aligned} & \text{minimize } J(f) \\ & \text{subject to } R_i^{\text{cvx}} \leq T_i, \forall 1 \leq i \leq n \\ & f_i \geq f_i^{\min}, \forall 1 \leq i \leq n \\ & f_i \leq f_i^{\max}, \forall 1 \leq i \leq n \end{aligned} \quad (8)$$

where $f_i = 100/T_i$ Hz, and R_i^{cvx} is as defined in Equation 7, $\forall 1 \leq i \leq n$.

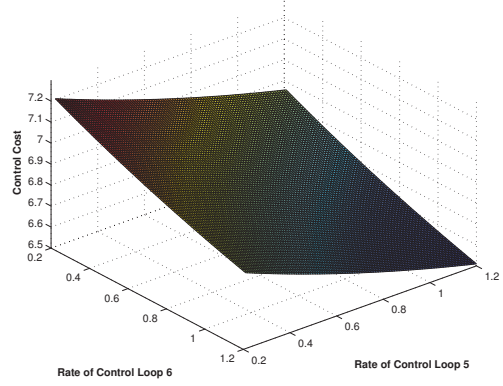


Fig. 3. Surface of the primal function of Problem in 8

The above is a convex optimization problem, and has closed-form solution. Figure 3 indicates the smoothness of the function in Problem 8 for changing the rates of 2 control loops (and keeping all other loops’ rates unchanged) considering data flows of 12 control loops simulated on a testbed topology (shown in Figure 4). To find a solution to the primal problem with 0 duality gap, we consider solutions to its dual problem. Here also, the dual problem is formed through Lagrangian relaxation of inequality constraints $R_i^{\text{cvx}} \leq T_i$, and is given by

$$\begin{aligned} & \text{maximize } L(f, \lambda) \\ & \text{subject to } \lambda \geq 0 \end{aligned}$$

Where $L(f, \lambda)$ is the Lagrangian dual function defined by

$$\begin{aligned} L(f, \lambda) &= \inf \{ J(f) + \sum_{i=1}^n \lambda_i (R_i^{\text{cvx}} - T_i) \} \\ & \text{such that } f_i^{\min} \leq f_i \leq f_i^{\max}, \forall 1 \leq i \leq n \end{aligned}$$

Here $\lambda \in \mathbb{R}^n$ is the vector of Lagrange multipliers. In the dual, $L(f, \lambda)$ is differentiable and, hence, the classical approach of maximizing the function would be the steepest descent method that computes a sequence of iterations to update the multipliers as follows

$$\lambda^{t+1} = \lambda^t + \gamma^t \nabla L(f, \lambda)$$

Note that at every step, the priorities of the control loops are updated according to rate monotonic policy based on new updated rates to calculate R_i^{cvx} . In solving the dual function, we follow the gradient at the current position, with a specified step size γ , to reach points with a higher function value. Unlike Algorithm 1, now we have unique subgradient (which is the gradient) at the current position. In our case, this evaluates to

$$\lambda^{t+1} = \lambda^t + \gamma^t (R^{\text{cvx}} - T)$$

Any traditional step size rule (either vanishing or dynamic) can be applied to reach the closed-form solution in a gradient descent way. Also, the solution can be found simply by using any standard convex optimization tool such as CVX [33].

X. EVALUATION

In this section, we evaluate the proposed algorithms for near optimal rate selection for feedback control loops in

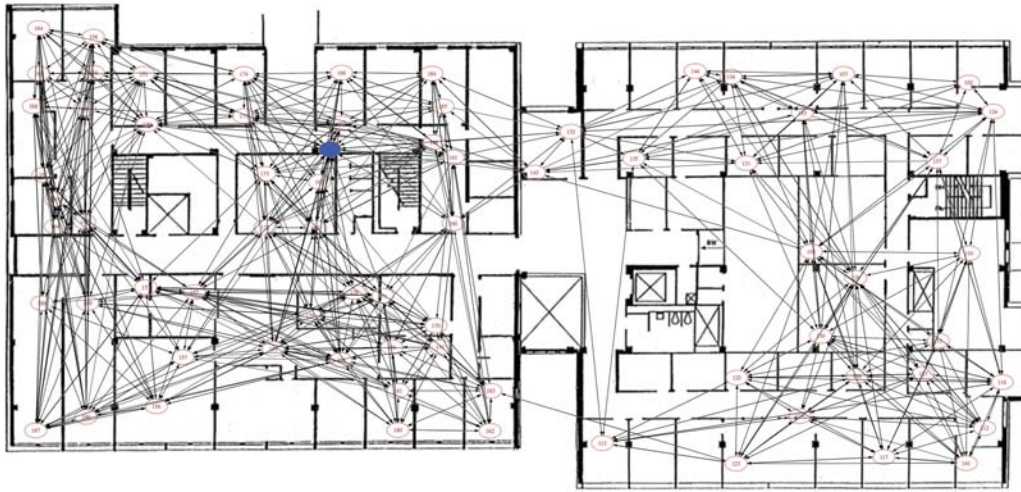


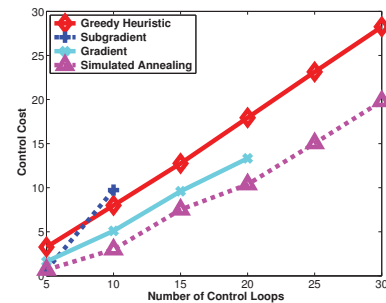
Fig. 4. Testbed topology at transmission power of -5 dBm (the gateway is colored in blue)

wireless control systems. We evaluate the algorithms through simulations based on the real topologies of a WSN testbed. Our WSN testbed is deployed in two buildings (Bryan Hall and Jolley Hall) of Washington University in St Louis [34]. The testbed consists of 74 TelosB motes each equipped with Chipcon CC2420 radios compliant with the IEEE 802.15.4 standard (WirelessHART is also based on IEEE 802.15.4).

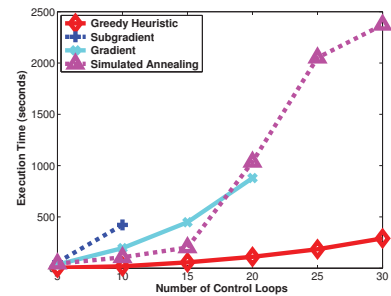
A. Simulation Setup

We simulate the networked control loops by generating data flows in our testbed topologies. The topologies are determined in the following way. Setting the same transmission power at every node, a node broadcasts 50 packets while its neighbors record the sequence numbers of the packets they receive. After a node completes sending its 50 packets, the next sending node is selected in a round-robin fashion. This cycle is repeated giving each node 5 rounds to transmit 50 packets in each round. Every link with a higher than 80% *packet reception ratio* (PRR) is considered a reliable link to derive the topology of the testbed. Figure 4 shows the network topology (embedded on the floor plans of two buildings) when each node's transmission power is set to -5 dBm. We have tested our algorithms using the topologies at 4 different transmission power levels: 0 dBm, -1 dBm, -3 dBm, -5 dBm.

The number of channels is set to 12. In each topology, the node with the highest number of neighbors is designated as the gateway. A set of nodes is considered as sources (sensors), while another set as destinations (actuators). We select the same source and destination pairs in each topology. The most reliable routes (based on PRR) are used for data flow between source and destination pairs. Each data flow is associated with a control loop. The weight of each control loop is set to 1. The decay rate (β) and magnitude coefficient (α) of the loops have been assigned according to those used for bubble control systems in [5]. The penalty based simulated annealing has been implemented based on Python Simulated Annealing Module [35]. All other algorithms have been implemented in Matlab. The tests have been performed on a Mac OS X



(a) Control cost

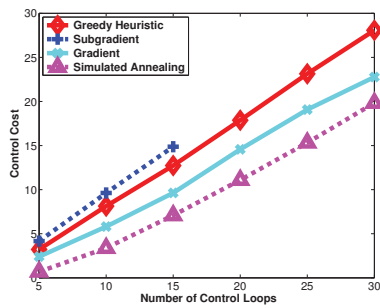


(b) Execution time

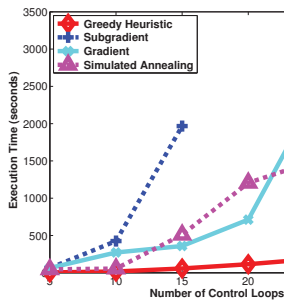
Fig. 5. Performance comparison on topology at transmission power 0 dBm machine with 2.4 GHz Intel Core 2 Duo processor.

B. Performance Study

We evaluate all 4 algorithms in terms of achieved *control cost* and *execution time*. Figure 5 shows the results for 30 control loops simulated on the testbed topology when every node's transmission power is set to 0 dBm. Figure 5(a) indicates that the control cost in the simulated annealing (SA) based penalty method is consistently a lot less than all other methods. The control cost in the gradient method is very close to that of SA for each number of loops. The control cost in the gradient method is at most 1.12 times that of SA, and is a lot less than the greedy heuristic and the subgradient method. The greedy heuristic is always achieving control cost higher than the gradient method, but less than the subgradient method



(a) Control cost



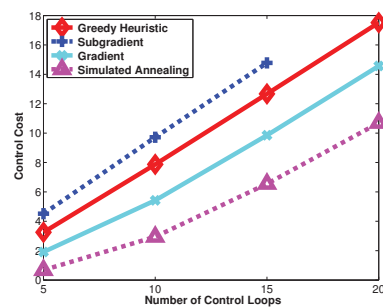
(b) Execution time

Fig. 6. Performance comparison on topology at transmission power -1 dBm

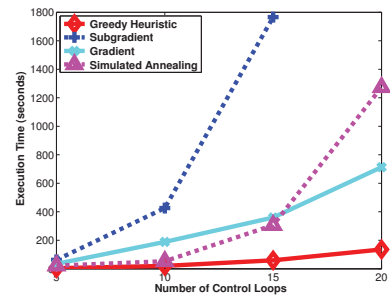
when number of loops is more than 5. The subgradient method takes a long execution time, and we were not able to get its results for more than 10 loops. For more than 20 loops, we have also observed that the gradient method takes a longer execution time (Figure 5(b)). The gradient method turns out to be a better option for a moderate number loops. According to Figure 5(b), the execution time of SA increases exponentially with the number of loops, but always remains less than the subgradient method. The greedy heuristic is a lot faster than other methods due to its polynomial time complexity.

Figure 6 shows the results for 30 control loops on the testbed topology with transmission power -1 dBm. Figure 6(a) indicates that the control cost in SA is consistently a lot less than all other methods. The control cost in the gradient method is at most 1.2 times that of SA, and is a lot less than the greedy heuristic and the subgradient method. The greedy heuristic is always achieving control cost higher than the gradient method, but less than the subgradient method. The subgradient method takes a long execution time. According to Figure 6(b), its time increases exponentially with the number of loops. The gradient method runs faster than SA when the number of loops is increased beyond 10 but does not become larger than 20. Figures 7 and 8 show similar results for the testbed topologies with transmission power -3 dBm and -5 dBm, respectively.

The results demonstrate that, among all methods, SA achieves the least control cost while requiring the longest execution time. The subgradient method turns out to be worse than all other algorithms both in terms of execution time and in terms of control cost. This is quite reasonable as our optimization problem is highly nonlinear and there exist a large number of local extrema. The subgradient direction becomes highly complicated and therefore both its execution time and control cost get worse. The greedy heuristic incurs



(a) Control cost



(b) Execution time

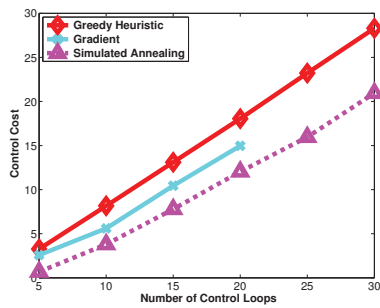
Fig. 7. Performance comparison on topology at transmission power -3 dBm

control cost at most 2.67 times that of SA, while keeping the execution time very low. The gradient based steepest descent method incurs control cost at most 1.35 times that of SA, while keeping the execution time less than SA in most cases. Therefore, to get near optimal results at the cost of longer execution time, SA turns out to be a prominent method. To get results very quickly and for scalability with a moderate control cost, the greedy heuristic turns out to be the best option. To achieve moderate control cost (not as high as greedy and not as low as SA) within a reasonable time (not as fast as greedy, not as slow as SA), the gradient descent method appears to be a promising approach for a moderate number control loops.

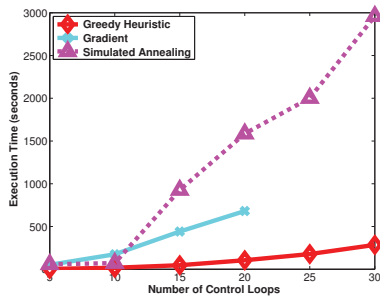
XI. CONCLUSION

Recent industrial standards such as WirelessHART have enabled real-world deployment of wireless control systems. Due to limited bandwidth in wireless sensor-actuator networks, it is important to optimize the control performance through a wireless-control co-design approach. This paper addresses the problem of determining the optimal sampling rates of feedback control loops sharing a WirelessHART network. The objective is to minimize the overall control cost while ensuring that all data flows meet their end-to-end deadlines. The resulting constrained optimization problem based on existing delay bounds for data flows in WirelessHART networks is difficult since it is non-differentiable, non-linear, and not in closed-form. We propose four approaches to solve this challenging problem: (1) a subgradient method, (2) a simulated annealing (SA) based penalty method, (3) a polynomial-time greedy heuristic method, and (4) a gradient descent method based on a new delay bound that is convex and differentiable.

We then perform a simulation study of the different approaches based on real testbed topologies and simulated



(a) Control cost



(b) Execution time

Fig. 8. Performance comparison on topology at transmission power -5 dBm

control systems. Interestingly, while subgradient methods are commonly adopted to solve non-linear constrained optimization problems, it leads to the highest control cost and significant computation times in solving our optimization problem. We found that it is due to a large number of local minima and high nonlinearity of our problem. SA consistently achieves the minimum control cost while incurring the longest execution time. Conversely, the greedy heuristic results in higher control cost using the shortest execution time. Convex optimization based on our new delay bound hits the balance between control cost and execution time for a moderate number of control loops. Our results represent a promising step towards wireless-control co-design involving complex interactions between control performance and real-time communication.

ACKNOWLEDGEMENT

This research was supported by NSF under grants CNS-1144552 (NeTS), CNS-1035773 (CPS), CNS-1017701 (NeTS), CNS-0708460 (CRI), a Microsoft Research New Faculty Fellowship, and a Sloan-Kettering Center grant.

REFERENCES

- [1] "WirelessHART specification," 2007, <http://www.hartcomm2.org>.
- [2] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "End-to-end delay analysis for fixed priority scheduling in WirelessHART networks," in *RTAS '11*.
- [3] —, "Improved end-to-end delay analysis in WirelessHART networks," Washington University in St Louis, Tech. Rep. 2011-86, 2011, <http://cse.wustl.edu/Research/Pages/technical-reports.aspx>.
- [4] F. Xia and Y. Sun, "Control-scheduling codesign: A perspective on integrating control and computing," *Dynamics of Cont., Discr. and Impulsive Syst.*, vol. 13, pp. 1352–1358, 2008.
- [5] D. Seto, J. P. Lehoczy, L. Sha, and K. G. Shin, "On task schedulability in real-time control systems," in *RTSS '96*.
- [6] E. Bini and A. Cervin, "Delay-aware period assignment in control systems," in *RTSS '08*.

- [7] L. Sha, X. Liu, M. Caccamo, and G. Buttazzo, "Online control optimization using load driven scheduling," in *CDC '00*.
- [8] Y. Halevi and H. Ray, "Performance analysis of integrated communication and control system networks," *J. Dyn. Syst. Meas. Control*, vol. 112, pp. 365 – 372, 1990.
- [9] M. Branicky, S. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *CDC '02*.
- [10] F.-L. Lian, J. Moyne, and D. Tilbury, "Network design consideration for distributed control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 2, pp. 297 –307, 2002.
- [11] F.-L. Lian, J. Yook, P. Otanez, D. Tilbury, and J. Moyne, "Design of sampling and transmission rates for achieving control and communication performance in networked agent systems," in *ACC '03*.
- [12] M. Gaid, A. Cela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: application to a car suspension system," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776 – 787, 2006.
- [13] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953 – 958, 2006.
- [14] S.-L. Dai, H. Lin, and S. S. Ge, "Scheduling-and-control codesign for a collection of networked control systems with uncertain delays," *IEEE Transaction on Control Systems Tech.*, vol. 18, no. 1, pp. 66 –78, 2010.
- [15] P. Marti, J. Yezpez, M. Velasco, R. Villa, and J. Fuertes, "Managing quality-of-control in network-based control systems by controller and message scheduling co-design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1159 – 1167, 2004.
- [16] H.-J. Korber, H. Wattar, and G. Scholl, "Modular wireless real-time sensor/actuator network for factory automation applications," *IEEE Trans. on Industrial Informatics*, vol. 3, no. 2, pp. 111–119, 2007.
- [17] X. Liu and A. J. Goldsmith, "Cross-layer design of distributed control over wireless network," in *Systems and Control: Foundations and Applications*, Birkhauser, 2005.
- [18] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *CDC '04*.
- [19] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, "Joint optimization of wireless communication and networked control systems," *Lecture Notes in Computer Science*, vol. 3355, pp. 248–272.
- [20] W. Shu, X. Liu, Z. Gu, and S. Gopalakrishnan, "Optimal sampling rate assignment with dynamic route selection for real-time wireless sensor networks," in *RTSS '08*.
- [21] X. Liu, Q. Wang, W. He, M. Caccamo, and L. Sha, "Optimal real-time sampling rate assignment for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, pp. 263–295, 2006.
- [22] D. Chen, M. Nixon, and A. Mok, *WirelessHART™ Real-Time Mesh Network for Industrial Automation*. Springer, 2010.
- [23] R. Alur, A. D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Modeling and analysis of multi-hop control network," in *RTAS '09*.
- [24] P. Soldati, H. Zhang, and M. Johansson, "Deadline-constrained transmission scheduling and data evacuation in WirelessHART networks," in *ECC '09*.
- [25] H. Zhang, F. Osterlind, P. Soldati, T. Voigt, and M. Johansson, "Rapid convergecast on commodity hardware: Performance limits and optimal policies," in *SECON '10*.
- [26] J. Pesonen, H. Zhang, P. Soldati, and M. Johansson, "Methodology and tools for controller-networking co-design in WirelessHART," in *EFTA '09*.
- [27] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in *RTSS '10*.
- [28] S. Han, X. Zhu, and A. K. Mok, "Reliable and real-time communication in industrial wireless mesh networks," in *RTAS '11*.
- [29] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Priority assignment for real-time flows in WirelessHART networks," in *ECRTS '11*.
- [30] "Subgradient solver: SSMS," http://www.searching-eye.com/sanjeevsharma/matlab_solver/subgradient_solver/.
- [31] B. Polyak, *Introduction to Optimization*, 1987.
- [32] Y. Chen and M. Chen, "Extended duality for nonlinear programming," *Comput. Optim. Appl.*, vol. 47, pp. 33–59, 2010.
- [33] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming," <http://cvxr.com/cvx/>.
- [34] Wireless sensor network testbed, <http://mobilab.wustl.edu/testbed>.
- [35] "Python simulated annealing module," <http://www-personal.umich.edu/~wagnerr/PythonAnneal.html>.