RESEARCH-ARTICLE

Scalable Real-Time Control in Industrial Cyber-Physical Systems

MD ASHIKUL HAQUE, Wayne State University, Detroit, MI, United States

DALI ISMAIL, Binghamton University State University of New York, Binghamton, NY, United States

ABUSAYEED SAIFULLAH, Wayne State University, Detroit, MI, United States

Published: 04 January 2024

ICDCN '24: International Conference on Distributed Computing and Networking
January 4 - 7, 2024
Chennai, India

ICDCN '24: Proceedings of the 25th International Conference on Distributed Computing and Networking (January 2024)
https://doi.org/10.1145/3631461.3631474
ISBN: 9798400716737

Scalable Real-Time Control in Industrial Cyber-Physical Systems

Md Ashikul Haque
Wayne State University
Detroit, MI, USA

Dali Isamil
Binghamton University
Binghamton, NY, USA

Abusayeed Saifullah
Wayne State University
Detroit, MI, USA

ABSTRACT

Real-time communication and control performance are the precursor of industrial cyber-physical systems that employ Wireless Networked Control System (WNCS) in critical industrial applications including process control and smart manufacturing. Control performance and real-time communication are interdependent in a WNCS. Hence, optimizing control performance under limited resource of network requires a cyber-physical codesign approach. A codesign approach needs to be online to take into account both the current network condition and the current control behavior. Leading industrial wireless standards such as WirelessHART and ISA100 adopt software-defined networking as a centralized routing mechanism. Hence, in current approach, transmission schedules of the entire network are created centrally at a network manager in advance and are then disseminated to the nodes. Control performance optimization usually requires to update sampling rates and/or priorities of the control loops, thereby requiring to re-create the schedules. Thus, it becomes highly inefficient under the current fully centralized scheduling approach. In this paper, we propose to optimize control performance in an industrial WNCS through a scheduling-control codesign based on a local and online scheduling approach proposed in a recent work. We formulate the scheduling-control codesign problem to optimize the control performance based on model predictive control theory. Unlike existing offline solution, our codesign complies with online scheduling and entails a rolling optimization to take into account the current control performance to dynamically update the rates and priorities of the control loops. We propose a highly scalable solution of the codesign problem based on a local search approach used as an anytime algorithm.

CCS CONCEPTS

• **Networks → Cyber-physical networks**.

KEYWORDS

Cyber-Physical Systems, Real-Time Control, Scalability

ACM Reference Format:
Md Ashikul Haque, Dali Isamil, and Abusayeed Saifullah. 2024. Scalable Real-Time Control in Industrial Cyber-Physical Systems . In *25th International Conference on Distributed Computing and Networking (ICDCN '24), January 04–07, 2024, Chennai, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3631461.3631474

ICDCN '24, January 04–07, 2024, Chennai, India

ACM ISBN 979-8-4007-1673-7/24/01…$15.00
https://doi.org/10.1145/3631461.3631474

1 INTRODUCTION

Real-time communication and control performance are the precursor of industrial cyber-physical systems (CPS) that employ Wireless Networked Control System (WNCS) in critical industrial applications including process control and smart manufacturing. Latency between sensing and actuation has a high impact on the stability of these control applications [29]. Specifically, large or unbounded delays may lead to unstable systems. The feedback control loops in a WNCS therefore impose real-time requirements on communication latency between sensors and actuators. However, industry settings pose a harsh environment for wireless communication causing frequent transmission failures due to mobility, physical obstacles, multi-path fading, and interference from coexisting wireless devices that make it difficult to meet these requirements [34].

With the advent of wireless control standards such as WirelessHART [9], ISA100 [8], and WIA-PA [10], WNCS has recently received a new impulse. WirelessHART was specifically designed to operate in highly unreliable wireless environments. With approximately 30 million HART devices installed across the world, it is predominantly being used worldwide for process control in harsh wireless environments with high production efficiency and robustness [1]. For example, the deployment of 800 HART devices for real-time process management at Mitsubishi Chemical plant in Kashima, Japan increased production performance saving US$20-30,000 per day that averted a US$3 million shutdown [2]. Some other notable deployments of WirelessHART include those for asset management at Monsanto [3], smart metering in Detroit Water & Sewerage [4], boiler feed water process management at Kapstone Paper [5], and smart manufacturing at Duke Energy [6].

Optimizing control performance in WirelessHART network is a highly demanding and challenging problem. Control performance and real-time communication are interdependent. Hence optimizing control performance under limited resource of network requires a cyber-physical codesign approach which is a difficult interdisciplinary research area. Besides, control signals are calculated online upon receiving the plant state from sensors. Hence, a codesign approach also needs to be online to take into account both the current network condition and the current control behavior. However, WirelessHART (like other industrial wireless standards such as ISA100) has adopted software-defined networking (SDN) as a centralized routing mechanism [9]. Hence, in current approach [8–10, 26–32], transmission schedules of the entire network are created centrally at a network manager in advance and are then disseminated to the nodes. Control performance optimization requires the sampling rates and/or priorities update, thereby requiring to re-create and re-distribute the schedules. Thus, control performance optimization becomes highly inefficient under the current fully centralized scheduling approach as it may require frequent schedule updates.

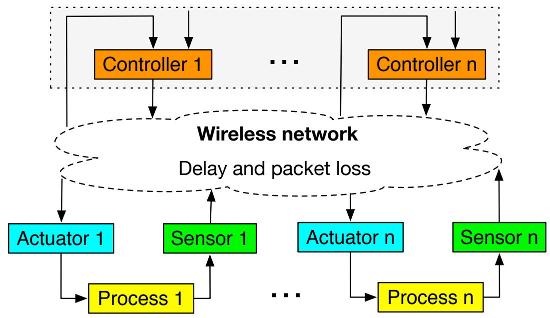

Figure 1: A WNCS architecture.

In this paper, we propose to optimize control performance through a scheduling-control codesign based on a local and online scheduling approach called DistributedHART proposed in a recent work [22]. We formulate the scheduling-control codesign problem under bandwidth constraints to optimize the control performance based on model predictive control theory. Unlike existing offline solution [27, 28], our codesign complies with online scheduling and entails a rolling optimization to take into account the current control performance to dynamically update the rates and priorities of the control loops. We propose a highly scalable solution of the codesign problem based on a *local search* approach used as an *anytime algorithm*.

In the rest of the paper, Section 2 describes the systems model along with an overview of WirelessHART. Section 3 presents related work. Section 4 gives an overview of DistributedHART and the adopted scheduling policy for our codesign. Section 5 presents the proposed scheduling-control codesign along with the proposed solution approach. Section 6 concludes the paper and also discusses future work.

2 WIRELESS NETWORKED CONTROL SYSTEM (WNCS)

In industrial CPS, a WNCS involves feedback control loops between sensors and actuators through a wireless network. Sensors measure process variables, and deliver to a controller through the network. The controller sends control commands to the actuators, which then operate the control and safety components to adjust physical processes. We consider a broad class of WNCS with applications in process control and smart manufacturing. The feedback loops are denoted by $F_1, F_2, \cdots, F_n$, where n is the total number of control loops. A simple architecture of a WNCS is shown in Fig. 1.

We consider that the control loops are periodic. A *periodic control loop* F_i has a period T_i which implies the sampling interval of its sensor/s.

Delivering process information or sensor data to the controller incurs a network latency of τ_{sc}^i. The time required to calculate the control command is τ_c^i. Delivering control commands to the actuator/s incurs a network latency of τ_{ca}^i. Thus, $\tau_{sc}^i + \tau_c^i + \tau_{ca}^i$ is called the ***end-to-end delay*** of F_i which can vary for different instances. Here $\tau_{sc}^i + \tau_{ca}^i$ is the ***network delay***. Delay impacts the control performance, and can vary due to unpredictable wireless conditions. However, for the stability of a control loop, the end-to-end delay has to be bounded by a value called the ***deadline*** D_i of F_i. The deadline is equal to the period, i.e., $D_i = T_i$.

A number of wireless standards in the 2.4GHz band such as WirelessHART [9], WIA-PA [10], ISA100 [8] designed on top of IEEE 802.15.4 [7] evolved in recent years for control. Our network is modeled based on the key features of these legacy standards, particularly WirelessHART that is predominantly used in industrial wireless control [1]. It is a multi-hop mesh network of a Gateway, field devices, and multiple access points. A *network manager* and the controllers are at the *Gateway*. The *field devices* are wirelessly networked sensors and actuators. Each node is equipped with a *half-duplex* omnidirectional radio transceiver that cannot both transmit and receive together and can receive if only one sender transmits to it at a time. *Access points* provide redundant paths between the wireless network and the Gateway. Very large networks are organized in a hierarchical structure connecting multiple smaller ones, each with its own network manager.

Transmissions are scheduled based on a multi-channel TDMA (time division multiple access) protocol. Time is globally synchronized. Each time slot is of fixed length. A transmission and its acknowledgment (ACK) needs one time slot. For transmission between a receiver and its sender, a time slot can be either dedicated or shared for the link between the sender and the receiver. In a ***dedicated slot***, only one sender is allowed to transmit to a receiver. In a ***shared slot***, multiple senders can attempt to send to a common receiver. Note this notion of 'shared' and 'dedicated' is a local concept; at the same slot some transmissions can be scheduled as dedicated slot while some are scheduled as shared slot. The network uses the channels defined in IEEE 802.15.4.

For enhanced reliability, the network adopts ***graph routing*** [9]. A ***routing graph*** is a directed list of paths between two nodes. Packets from all sensor nodes are routed to the Gateway through an ***uplink graph*** denoted by UG. For each F_i, there is a ***downlink graph*** DG_i from the Gateway through which it delivers control messages to F_i's actuator. In each routing graph, a node can have multiple neighbors, say *parents*, to which it can transmit and retransmit a packet multiple times to be delivered to the corresponding destination. Typically, a packet at node a with two parents b and c is first scheduled along link $a \longrightarrow b$ on a dedicated slot. If a does not receive an ACK from b, it retransmits to b again on a second dedicated slot. If that also fails, a transmits it to c on a shared slot (allowing others to attempt to transmit to c in this slot). WirelessHART (like other industrial wireless standards such as ISA100) adopts SDN as a centralized routing mechanism, where the graph routes are determined at the control plane centrally.

A WNCS has to satisfy three key requirements: **(a) real-time:** for stability, each F_i must satisfy $\tau_{sc}^i + \tau_c^i + \tau_{ca}^i \leq D_i$; **(b) reliability** of communication; **(c) control performance** optimization.

3 RELATED WORK

Scheduling-control codesign was studied for wireless CPS in [17–19, 21, 25, 36, 37] (also see survey [35]) considering single-hop networks. Scheduling-control codesign for multi-hop wireless networks was studied in [12, 33]. The work in [33] considers scheduling-control co-design for only a single control loop. This work in [12] does not involve multi-path graph routing or multi-channel. Its delay bounds based on virtual link capacity margin cannot be applied to our network model even under single channel or single path routing. Its communication protocol does not employ any priority-based real-time scheduling policy. In contrast, our work focuses on scheduling-control codesign based on current industrial wireless standards that exploit the spectrum and spatial diversity for real-time communication in highly unreliable environments. In addition, our approach is compliant with online real-time scheduling and entails a rolling optimization based on local search to take into account the current control behavior and to dynamically update the sampling rates and priorities of the control loops.

A scheduling-control codesign problem of sampling rate selection for centralized scheduling was studied in [27, 28]. That work considered a simplified network model without graph routing. Also, its control performance metric did not take into account the network delay impact. It was designed as an offline solution for fully centralized scheduling. It does not take into account the current control behavior which requires an online codesign.

4 TRANSMISSION SCHEDULING

In industrial wireless control applications based on WirelessHART, transmission schedules of the entire network are created centrally at a network manager in advance and are then disseminated to the nodes. To make reliable and real-time communication in highly unreliable wireless environments, a common feature is to adopt a high degree of redundancy at the expense of scalability. To circumvent transmission failure, a packet is scheduled on multiple time slots, on multiple links in multi-path graph routing, and on multiple channels using a TDMA-based MAC protocol based on both dedicated and shared time slots. While they are crucial to real-time control for handling worst-case scenarios, the centralized scheduling with high redundancy in current approach causes a huge waste of time, bandwidth, energy, and memory of the nodes even under a small network, let alone large-scale ones. As the hyperperiod grows exponentially with the number and periods of the control loops, the schedule length grows exponentially, hindering the network's scalability. Storing the schedules in the memories of the nodes also introduces a big overhead. In addition, if the sampling rate or priority of a control loop changes, the centralized approach needs to re-create and re-distribute the entire schedule. To overcome these limitations, DistributedHART was proposed in [22, 23]. In our proposed scheduling-control codesign, we adopt scheduling based on DistributedHART.

Here, we provide an overview of DistributedHART [22]. Its essence is to adopt local (node-level) scheduling through a time window allocation among the nodes denoted by set V. This is done after the routes are generated. A node schedules its transmissions locally and online using a real-time scheduling policy. Dedicated slot transmissions are scheduled in its assigned window while shared slot ones may be scheduled outside. ***Time window allocation*** is a function $w : V \longrightarrow \{1, 2, 3, \cdots, \}$ with the windows numbered as $1, 2, 3, \cdots$, each of **length** ℓ, where ℓ consists of one or more time slots. Every node will be assigned one time window. Two nodes are ***in conflict*** if and only if a transmission from one causes radio interference at the receiver of a simultaneous transmission from the other. To minimize such conflicts, DistributedHART first performs a ***receiver based channel allocation*** $g : V \longrightarrow M$ where each node is assigned a fixed channel from a set of channels M to receive message; its neighbors use this channel to send to it. After this, the time window allocation is done such that all nodes having the same window are conflict-free. To minimize network latencies of the control loops, the objective is to minimize the number of time windows, called the *worst case chromatic number* $|w(V)|$. Every node can autonomously schedule the current packets that it has in its assigned time window using the corresponding receiver's channel. It does not need to create any schedule in advance. After every ***cycle*** of $|w(V)|\ell$ time slots, the windows repeat. We refer to [22], for a detailed description including the handling of shared and dedicated slots of DistributedHART. Network latency bound for each flow under DistributedHART has been derived in [23].

We adopt fixed priority scheduling which is widely used in practice, e.g., in CPU scheduling [14] and Control-Area Networks [15]. For priority assignment, we consider ***Deadline Monotonic (DM)*** policy. DM assigns priorities to control loops based on their relative deadlines; the loop with the shortest deadline being assigned the highest priority. In our model, rate monotonic and DM are the same as deadline equals period. Each node adopts the DM policy locally. Namely, when it has multiple packets, it first transmits the one with the shortest deadline. Thus, schedule updates upon any rate or priority change can be handled locally. This also enables aperiodic events with their own deadlines. Management communication can run in parallel with the highest priority.

5 SCHEDULING-CONTROL CODESIGN TO OPTIMIZE CONTROL PERFORMANCE

Control performance and real-time communication are interdependent in industrial CPS. Hence, we will optimize control performance through a scheduling-control codesign under limited bandwidth. The **challenges** in this codesign are many. Since control signals are calculated online upon receiving the plant state from sensors, such a codesign needs to be online to take into account both the current network condition and the current control behavior. Hence, it has to be highly efficient which involves challenging interdisciplinary research. Besides, the control behavior needs to be modeled to take into account the impact of network delay. Furthermore, control performance optimization requires the sampling rates and/or priorities update, thereby requiring to re-generate transmission schedules. Thus, control performance optimization becomes highly inefficient under current fully centralized scheduling approach. We address these challenges by proposing a scheduling-control codesign framework that complies with online scheduling (DistributedHART) and entails a rolling optimization to take into account the current control performance to dynamically update the rates and priorities of the control loops.

5.1 Controller Design

We design and analyze the codesign framework based on ***Model Predictive Control (MPC)*** theory. MPC is a widely used advanced control methodology (e.g., in process and petrochemical industries) due to its ability to handle multivariable complex systems [20]. Another advantage is its ability to compute outputs $y(k)$ and control signals $u(k)$ in advance to avoid sudden changes in control signal and undesired effect of delay in system response. We assume that the plant is a linear time-invariant discrete-time system as follows.

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k)$$

where $x(k)$ is the plant state, and $u(k)$ and $y(k)$ are input and output vectors, respectively, at time k; A, B, C are system matrices. For a control loop F_i, we consider finding the optimal control sequence on the horizon of prediction by minimizing the objective function given by

$$\begin{aligned} J_i(k) = &\sum_{j=1}^{N_p} Q(j)[y(k+j|k) - r(k+j)]^2 + \sum_{j=0}^{N_c-1} R(j)[\delta u(k+j)]^2 \\ &\textbf{\textit{subject to}} \\ &\delta u^{\min} \le \delta u \le \delta u^{\max}; \\ &(y-r)^{\min} \le (y-r) \le (y-r)^{\max}; \\ &u^{\min} \le u \le u^{\max} \end{aligned} \tag{1}$$

where $\delta u(k) = u(k) - u(k-1)$; $Q(j), R(j) > 0$ are weighting matrices, N_p is the ***prediction horizon*** and N_c is the ***control horizon*** (in number of samples); $r(k+j)$ is the reference trajectory; the notation $y(k+j|k)$ means that the value of y at time $k+j$ depends on the condition at time k; For any variable v, $v^{\min}$ and $v^{\max}$ denote its minimum and maximum range, respectively. These constraints are designed to control overshoots. The first term in the cost function represents ***tracking error***, and the second term represents ***control penalty***.

5.2 Scheduling-Control Codesign Formulation

Our proposed codesign leverages sampling rate selection of the feedback loops due to its significant impact on control and real-time performance [12, 27, 28]. A low sampling rate degrades the control performance while a high one may cause excessive communication delays degrading it as well. In addition, in our DM scheduling model, any sampling rate update may change the scheduling priorities of the control loops.

To comply with the online scheduling, we propose a scheduling-control codesign approach that runs at the network manager and entails a rolling optimization to take into account the current control behavior and to dynamically update the sampling rates as well as the scheduling priorities of the control loops. To take into account the current control behavior, we consider minimizing the ***predictive tracking error*** preliminarily defined as follows. We consider a large horizon of length H in time from the current time. Control loop F_i with sampling period T_i (rate $\varphi_i = \frac{1}{T_i}$) will have at most $\eta_i = \lfloor \frac{H}{T_i} \rfloor$ samples in this horizon. At the beginning of this time horizon, suppose the most recent sample for F_i is the k-th sample. Therefore, the output at the j-th sampling instant in the horizon, where $1 \le j \le \eta_i$, will be: $y(k+j|k)$. We use ξ_i to indicate **predictive tracking error for** F_i and define it as the root mean square of the predictive errors of all sampling instants in the

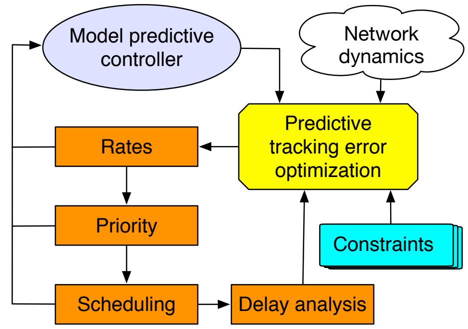

Figure 2: Scheduling-control codesign.

horizon of length H. Clearly, ξ_i is a function of sampling rate φ_i. ξ_i is calculated in the beginning of the horizon as follows.

$$\xi_i(\varphi_i, H) = \sqrt{\frac{1}{\eta_i}\sum_{j=1}^{\eta_i}\left(y(k+j|k) - r(k+j)\right)^2}, \text{ where } \eta_i = \left\lfloor \frac{H}{T_i} \right\rfloor \tag{2}$$

A high sampling rate makes the prediction better and reduces ξ_i. However, in the presence of multiple control loops sharing the same network, sampling rate of one control loop impacts that of the others. Hence, our objective is to minimize the ***overall predictive tracking error*** considering all n control loops while ensuring the stability of the system. Each sampling rate φ_i has a minimum and a maximum range, denoted by $\varphi_i^{\min}$ and $\varphi_i^{\max}$, respectively, that are required for **stability**. For stability, the end-to-end latency between sensing and actuation must not exceed the sampling period (deadline). These constraints also cover the network **bandwidth constraints** as they determine the schedulability. Hence, we preliminarily formulate the codesign as the following non-linear optimization problem: *Select sampling rates* $\varphi_1, \varphi_2, \cdots, \varphi_n$ *to*

$$\begin{aligned} &\textbf{\textit{minimize}} \sum_{i=1}^{n} \omega_i \xi_i \\ &\textbf{\textit{subject to}}\ \ \varphi_i^{\min} \le \varphi_i \le \varphi_i^{\max} \\ &\qquad \hat{\tau}_{sc}^i + \hat{\tau}_c^i + \hat{\tau}_{ca}^i \le T_i,\ \varphi_i = \frac{1}{T_i} \end{aligned} \tag{3}$$

where, for control loop F_i, ω_i is the weight, $\hat{\tau}_{sc}^i + \hat{\tau}_{ca}^i$ is a network delay bound detailed in [23], and $\hat{\tau}_c^i$ is the maximum value of τ_c^i. Note that even if the network does not change, we have to observe the control behavior and readjust the sampling rates. Hence, the above codesign runs as a rolling optimization problem after every H interval. It is also triggered by some network dynamics such as channel blacklisting. Fig. 2 shows a block diagram of our codesign framework.

Taking into Account Delay Effects on Tracking Error:

While the end-to-end delay bounds need to meet deadlines, it is also critical to consider the impact of network delay on the predictive tracking error. Let $\tau_{sc,k}$ be the network latency for the k-th sample to arrive from sensor to controller. When the controller calculates the predictive output upon receiving the k-the sample, it does not have the state information of exactly that time because the sensor data was sampled $\tau_{sc,k}$ time before. For ease of notation, let $\tilde{k}$ be the absolute time when the k-the sample was received at the controller. To consider the impact of this delay, we determine the output of k-the sample time as part of the predictive output sequence. Thus taking this into account in Equation (2), the predictive tracking error can be rewritten as

$$\xi_i(\varphi_i, H) = \sqrt{\frac{1}{\eta_i + 1} \sum_{j=0}^{\eta_i} \left(y(k+j|\tilde{k} - \tau_{sc,k}) - r(k+j)\right)^2}$$

5.3 Solution Approach

The problem (3) runs as a rolling optimization problem at the network manager. In a hierarchical structure that spans multiple smaller networks each with a separate network manager, the manager in each smaller network runs this optimization. Its running interval depends on how dynamic the control behavior and network condition are. As illustrated in Fig. 2, the optimization of the control cost formulated above involves a complex interaction between wireless scheduling and control, making a complicated co-design problem. We can use the delay bounds $\hat{\tau}^i_{sc} + \hat{\tau}^i_{ca}$ derived in [23] or derive tighter bounds. As observed before [27, 28], the delay bounds $\hat{\tau}^i_{sc} + \hat{\tau}^i_{ca}$ can be highly non-linear and non-differentiable, making such an optimization problem non-differentiable. The delay bounds derived in [23] are also non-linear and non-differentiable. We propose to solve the co-design problem using ***local search algorithm*** [13]. Local search is a general framework for solving hard nonlinear optimization problems. It moves from one solution to another in a controlled and intelligent way. It is a general framework for solving hard nonlinear optimization problems. The idea underlying local search is to explore the search space of partial and full solutions by applying local changes, until a target solution is found or the stopping condition is satisfied.

5.3.1 Rationale. Given an optimization problem $\mathbb{P}$, a local search algorithm typically consists of four components. 1) ***State space***: the state space $\mathbb{S}$ of $\mathbb{P}$ is the set of all possible variable assignments (including partial assignments); 2) ***Neighbors:*** for each variable assignment $s \in \mathbb{S}$, its neighbor set $Neigh(s)$ is a subset of $\mathbb{S}$; 3) ***Heuristic:*** the heuristic is a function $f : \mathbb{S} \mapsto \mathbf{R}$. It gives a quantitative estimation on how good a (partial) variable assignment $s \in \mathbb{S}$ is; 4) ***Pruning:*** the pruning condition decides if a state is worth inspecting or not. Intuitively, any local search forms a *decision tree*. Nodes in the decision tree are variable assignments in the state space and a node's children are its neighbors. The heuristic is used to evaluate the quality of branches and the pruning condition is used to prune certain branches of the decision tree.

There are some major advantages of using local search over traditional approaches such as mixed-integer programming or global optimization frameworks such as simulated annealing. First, local search can handle both discrete and continuous variables in a unified fashion without requiring continuity or differentiability of functions. Second, it can be very efficient when guided by proper heuristics and pruning bounds. The heuristic and pruning condition provide mechanisms that allow us to leverage domain knowledge such as delay bounds that can be derived based on the end-to-end delay analysis we developed for WNCS [30]. Third, it can be used as an ***anytime approach*** that first finds a feasible solution, and then improves as time allows. Finally, local search supports quick adaptation to network dynamics as local adaptation can be done efficiently on the search tree. It is particularly suitable for problem (3) as the solution from the previous horizon provides a good initial solution. Methods like integer programming often require to be rerun from scratch when the system changes. A key innovation of our approach is to develop and leverage efficient end-to-end delay bounds that can be used to steer local search to near-optimal solutions efficiently.

5.3.2 Neighbor Generation. We will leverage efficient neighborhood structure to move to a better and feasible solution. For example, when some attribute of a control loop changes, which also needs to change some attributes of other loops to meet real-time requirements, we can only limit exploring its lower priority loops since it does not affect any higher priority loop's schedulability. When some link breaks, a neighborhood solution would be to simply try with an alternative link or path instead of exploring all links in the network. In particular, to balance between solution quality and execution cost, we propose to use several existing neighborhood structures such as *genetic local search (GLS) neighborhood* [24, 38] and *cyclic exchange neighborhood* [11]. GLS combines the advantages of local search and genetic algorithm for faster solution, and does not require us to examine all neighborhood solutions. The approach has also been shown to be highly effective in multi-objective optimization [16]. In the neighborhood structure of GLS, adjacent or randomly selected elements are exchanged for mutation and local search. A *cyclic exchange neighborhood* consists of a sequence of elements being transferred among a family of subsets. A neighbor is obtained by performing a cyclic exchange without explicitly enumerating and evaluating all neighbors in the neighborhood. We propose to adopt these methods for very fast local search in our optimization. Further, we plan to exploit the gradient of the smooth approximation of the Lagrangian function [27] which can enhance the above neighbor generation schemes so that they focus on regions that are aligned with the gradient.

5.3.3 Heuristic and Pruning. We will use delay bounds to devise novel pruning techniques and good heuristics that improve the efficiency of local search methods for solving the codesign optimization problem. The work in [31] has shown how to compute an upper bound and a lower bound of delay, and how to discard redundant branches using those bounds in local search for optimal or near-optimal priority assignment. Using a similar spirit, we will devise efficient pruning methods and heuristics to guide the local search. To make our framework highly efficient, we will adopt *anytime local search* that first finds a feasible solution and then improves the performance as more computing time is allowed. For this, we set the heuristic function into a weighted sum $(\beta J + h)$ of the feasibility heuristic h and objective J. When we have no initial solution, β is set to zero to let the local search find a feasible solution. Then, β is gradually increased so that the search is oriented towards high-quality solutions to optimize the control performance. This approach allows us to achieve a good time-quality trade-off for real-time controlled operations.

5.4 Rate Distribution

In our DM scheduling model, sampling rate update changes the scheduling priorities of the control loops. For every source node whose rate gets updated, the controller generates a new control message with the information of the new rate, and sends to it. This is propagated through management communication that can be periodic or aperiodic and have the highest priority. Management communication can run in parallel with other control loops. The rate (as 1/period) is written in the packet by each source (sensor) node. The nodes on its route can schedule the packets using new priorities.

6 CONCLUSION AND FUTURE WORK

In this paper, we have proposed a scalable scheduling-control codesign approach to optimize control performance in industrial cyber-physical systems (CPS) for various applications including process control and smart manufacturing. For industrial CPS,

we have proposed a scheduling-control codesign approach that leverages a local and online scheduling named DistributedHART. We have formulated the codesign problem to optimize the control performance based on model predictive control theory. Unlike existing offline solution, our codesign complies with online scheduling and entails a rolling optimization to take into account the current control performance to dynamically update the rates and priorities of the control loops. We have proposed a highly scalable solution of the codesign problem based on a local search approach. In the future, we shall complete the proposed local search based solution of the codesign, implement it on a physical testbed, and evaluate it considering real-industrial use cases.

ACKNOWLEDGMENTS

The work was supported by NSF through grants CNS-2301757, CAREER- 2306486, CNS-2306745, and by ONR through grant N00014-23-1-2151.

REFERENCES

[1] [n. d.]. http://it.hartcomm.org/hcp/tech/wihart/wireless_overview.html.
[2] [n. d.]. http://it.hartcomm.org/hcp/tech/applications/applications_success_mitsubishi_chemical.html.
[3] [n. d.]. http://it.hartcomm.org/hcp/tech/applications/applications_success_monsanto.html.
[4] [n. d.]. http://it.hartcomm.org/hcp/tech/applications/applications_success_detroitwater.html.
[5] [n. d.]. http://it.hartcomm.org/hcp/tech/applications/applications_success_Kapstone.html.
[6] [n. d.]. http://it.hartcomm.org/hcp/tech/applications/applications_success_duke.html.
[7] [n. d.]. IEEE 802.15.4. http://standards.ieee.org/about/get/802/802.15.html.
[8] [n. d.]. ISA100: Wireless Systems for Automation. http://www.isa.org/MSTemplate.cfm?MicrositeID=1134&CommitteeID=6891.
[9] 2007. WirelessHART. https://www.fieldcommgroup.org/technologies/hart.
[10] 2011. I.E.C.C. IEC/PAS 62601: IndustriaL Communication Networks – Fieldbus Specifications – WIA-PA Communication Network and Communication Profile. International Electrotechnical Commission: Worcester, MA, USA.
[11] Ravindra K. Ahujaa, James B. Orlinb, and Dushyant Sharmac. 2000. Very large-scale neighborhood search. *Intl. Trans. in Op. Res.* (2000), 301–317.
[12] Jia Bai, Emeka P. Eyisi, Fan Qiu, Yuan Xue, and Xenofon D. Koutsoukos. 2012. Optimal Cross-Layer Design of Sampling Rate Adaptation and Network Scheduling for Wireless Networked Control Systems. In *Proceedings of the IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*. 107–116.
[13] Roberto Battiti, Mauro Brunato, and Franco Mascia. 2008. *Reactive Search and Intelligent Optimization.* Springer.
[14] G. C. Buttazzo. 2005. *Hard Real-Time Computing Systems.* Springer. 2nd edition.
[15] RobertI. Davis, Alan Burns, ReinderJ. Bril, and JohanJ. Lukkien. 2007. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Systems* 35, 3 (2007), 239–272.
[16] H. Ishibuchi and T. Murata. 1998. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 28, 3 (aug 1998), 392 –403.
[17] H.-J. Korber, H. Wattar, and G. Scholl. 2007. Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications. *IEEE Transactions on Industrial Informatics* 3, 2 (2007), 111–119.
[18] Xiangheng Liu and A. Goldsmith. 2004. Wireless network design for distributed control. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*. 2823–2829.
[19] Xiangheng Liu and Andrea J. Goldsmith. 2005. Cross-layer design of distributed control over wireless network. In *Proceedings of the Systems and Control: Foundations and Applications, Birkhauser.*
[20] J M Maciejowski. 2002. *Predictive Control With Constraints.* Prentice Hall.
[21] Venkata Prashant Modekurthy and Abusayeed Saifullah. 2019. Online Period Selection for Wireless Control Systems. In *2019 IEEE International Conference on Industrial Internet (ICII)*. 170–179.
[22] Venkata Prashant Modekurthy, Abusayeed Saifullah, and Sanjay Madria. 2019. DistributedHART: A Distributed Real-Time Scheduling System for WirelessHART Networks. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 216–227.
[23] Venkata P. Modekurthy, Abusayeed Saifullah, and Sanjay Madria. 2021. A Distributed Real-Time Scheduling System for Industrial Wireless Networks. 20, 5, Article 46 (Jul 2021), 28 pages.
[24] T. Murata, H. Ishibuchi, and M. Gen. 1998. Neighborhood structure for genetic local search algorithms. In *Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems (KES '98).*
[25] P. Park, J. Araújo, and K.H. Johansson. 2011. Wireless networked control system co-design. In *Networking, Sensing and Control (ICNSC), 2011 IEEE International Conference on.* IEEE, 486–491. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5874926
[26] Abusayeed Saifullah, Dolvara Gunatilaka, Paras Tiwari, Mo Sha, Chenyang Lu, Bo Li, Chengjie Wu, and Yixin Chen. 2015. Schedulability Analysis under Graph Routing in WirelessHART Networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*. 165–174.
[27] Abusayeed Saifullah, Chengjie Wu, Paras Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen. 2012. Near Optimal Rate Selection for Wireless Control Systems. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 231–240.
[28] Abusayeed Saifullah, Chengjie Wu, Paras Tiwari, You Xu, Yong Fu, Chenyang Lu, and Yixin Chen. 2013. Near Optimal Rate Selection for Wireless Control Systems. *ACM Transactions on Embedded Computing Systems* 13, 4s (2013), 128:1–128:25. Special Issue on Real-Time and Embedded Systems.
[29] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. 2010. Real-Time Scheduling for WirelessHART Networks. In *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*. 150–159.
[30] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. 2011. End-to-End Delay Analysis for Fixed Priority Scheduling in WirelessHART Networks. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 13–22.
[31] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. 2011. Priority Assignment for Real-time Flows in WirelessHART networks. In *Proceedings of the IEEE Euromicro Conference on Real-Time Systems (ECRTS)*. 35–44.
[32] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. 2014. End-to-end communication delay analysis in industrial wireless networks. *IEEE Trans. Comput.* 64, 5 (2014), 1361–1374.
[33] F. Smarra, A. D'Innocenzo, and M. D. Di Benedetto. 2012. Optimal co-design of control, scheduling and routing in multi-hop control networks. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. 1960–1965.
[34] J. Song, A. K. Mok, D. Chen, and M. Nixon. 2006. Challenges of Wireless Control in Process Industry. In *Proceedings of the Workshop on Research Directions for Security and Networking in Critical Real-Time and Embedded Systems (CRTES)*. 1–4.
[35] Yuan-Qing Xia, Yu-Long Gao, Li-Ping Yan, and Meng-Yin Fu. 2015. Recent progress in networked control systems — A survey. *International Journal of Automation and Computing* 12, 4 (01 Aug 2015), 343–367.
[36] Lin Xiao, Mikael Johansson, Haitham Hindi, Stephen Boyd, and Andrea Goldsmith. 2005. Joint Optimization of Wireless Communication and Networked Control Systems. *Switching and Learning in Feedback Systems* 3355 (2005), 248–272.
[37] Hao Xu and S. Jagannathan. 2013. *Distributed Joint Optimal Network Scheduling and Controller Design for Wireless Networks.* Springer New York, 147–162.
[38] Takeshi Yamada and Ryohei Nakano. 1996. Scheduling by genetic local search with multi-step crossover. 1141 (1996), 960–969.