Control-Aware Scheduling for Low Latency Wireless Systems with Deep Learning

Mark Eisen† Mohammad M. Rashid† Dave Cavalcanti† Alejandro Ribeiro*

***Abstract*—We consider the problem of scheduling transmissions over low-latency wireless communication links to control various control systems. Low-latency requirements are critical in developing wireless technology for industrial control, but are inherently challenging to meet while also maintaining reliable performance. An alternative to ultra reliable low latency communications is a framework in which reliability is adapted to control system demands. We formulate the control-aware scheduling problem as a constrained statistical optimization problem in which the optimal scheduler is a function of current control and channel states. The scheduler is parameterized with a deep neural network, and the constrained problem is solved using techniques from primal-dual learning, which have a necessary model-free property in that they do not require explicit knowledge of channels models, performance metrics, or system dynamics to execute. The resulting control-aware deep scheduler is evaluated in empirical simulations and strong performance is shown relative to other model-free heuristic scheduling methods.**

***Index Terms*—wireless control, low-latency, deep learning, primal-dual**

I. Introduction

The recent advances in wireless technology and automation have given rise to efforts in integrating wireless communications in autonomous environments, particularly in industrial control settings where the scale of wired networks is proving increasingly costly [1]. The analysis of control systems operating over wireless communication links is thus an integral apart in enabling these wireless industrial automation applications. However, the performance specifications of Tactile Internet applications demands the design of wireless networks that can meet both the high reliability and low latency demands of the system [1]–[3]. Ultra reliable low latency communications (URLLC) is inherently challenging as the physical medium of wireless communication trades off reliability and latency, making it hard to meet both demands.

One promising direction in enabling low latency communications involves specific developments in radio resource allocation, or scheduling. For low latency applications, traditional delay-aware schedulers [4]–[6] have been employed, in addition to more recent URLLC techniques based on various forms of diversity [2], [7], [8]—all of which are agnostic to the control system. However, due to the physical limitations of the wireless channel, it is often necessary to use information from the control system to make proper use of scheduling resources in meeting latency requirements. While there exist numerous ways in which control system information is incorporated into "control-aware" scheduling methods [9]–[14], these are agnostic to latency requirements of the system. More recent work [15] looks at heuristic based scheduling methods that are both control and latency aware, but whose practical use in low latency systems is limited both by its computational complexity at every scheduling cycle and reliance on explicit knowledge of the communication model and control dynamics.

Such existing methods, however, rely on accurate system knowledge, including plant dynamics and communication network parameters. Most practical systems present in modern industrial environments do not have accurate models available, inspiring the use of machine learning approaches to make intelligent scheduling and resource allocation decisions in wireless control systems without requiring model knowledge. This leads to a natural use of learning models, such as deep neural networks (DNNs), for designing schedulers. Most DNN training techniques are not immediately usable here as the latency requirements of the system pose constraints in the optimization problem [16]. Recent advancements apply techniques from both reinforcement learning and deep learning for control-aware scheduling in simple systems [12]–[14] and traditional wireless systems with latency constraints [17], [18]. Learning-based scheduling policies are well suited for URLLC and control as the computational complexity at each scheduling round is very low and can furthermore be implemented model-free when system dynamics and communication models are unknown. Our contributions namely consist of

1) formulating a statistical learning problem for control-aware, low latency scheduling as a function of both channel *and* control states,
2) parameterizing the scheduling policy with a deep neural network (DNN), and
3) utilizing a *model-free*, primal-dual learning framework to find control-aware scheduling policies.

This paper is organized as follows. We discuss the wireless control system in which state information is communicated to the control over a wireless channel as a switched dynamical system (Section II). We formulate the optimal scheduling problem that minimizes a control cost under latency constraints (Section II-A) and parameterize the optimal policy with a deep neural network (Section II-B). The constrained

Supported by Intel Science and Technology Center for Wireless Autonomous Systems. At the time of this work, the authors were with (†)Intel Corporation and the (∗)Department of Electrical and Systems Engineering, University of Pennsylvania. Email: mark.eisen@intel.com, mamun.rashid@intel.com, dave.cavalcanti@intel.com, aribeiro@seas.upenn.edu.

978-1-7281-7440-2/20/$31.00 ©2020 IEEE

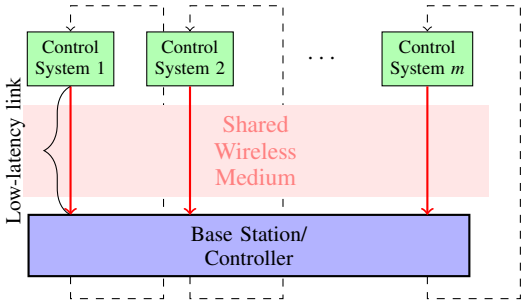

Fig. 1: A series of independent wireless control systems send state information over a shared wireless medium to a base station, where control information is fed back to the systems. The uplink transmissions (red arrow) is subject to latency constraint $t_{\max}$.

learning problem is solved using a so-called primal-dual learning method (Section III). We further discuss ways in which the primal-dual method can be approximated without explicit model knowledge (Section III-A). The performance of the learned control-aware scheduling method is analyzed in a numerical simulation and compared against existing baseline scheduling methods (Section IV) to show benefits of model-free learning and the consideration of control and channel conditions in making scheduling decisions.

II. Wireless Control Systems

We consider a series of m control systems—each a wireless device or plant—operating over a shared wireless channel as shown in Figure 1. The state of system i at control cycle index k is given by the variable $\mathbf{x}_i^k \in \mathbb{R}^p$. At each control/scheduling cycle, the sensor measures the state $\mathbf{x}_i^k$ and transmits it over a wireless channel to a common base station (BS) that is co-located with the controller. Given the state information, the controller determines the necessary control input which is fed back to the system. This is referred to as the closed-loop configuration of the control cycle. Given the noisy nature of the wireless channel, there is the potential for the communications packet containing the state information to be dropped, resulting in an open-loop configuration of the control cycle. We may model the linear dynamics of the wireless control system for system i as

$$\mathbf{x}_i^{k+1} = \begin{cases} \hat{\mathbf{A}}_i \mathbf{x}_i^k + \mathbf{w}^k & \text{if packet received} \\ \mathring{\mathbf{A}}_i \mathbf{x}_i^k + \mathbf{w}^k & \text{otherwise} \end{cases}, \quad (1)$$

where $\hat{\mathbf{A}}_i \in \mathbb{R}^{p\times p}$ is the closed loop gain, $\mathring{\mathbf{A}}_i \in \mathbb{R}^{p\times p}$ is the open loop gain, and $\mathbf{w}^k \in \mathbb{R}^p$ is zero-mean i.i.d. disturbance process with covariance $\mathbf{W}$. The closed loop and open loop gains may reflect, e.g., controlled dynamics using accurate and estimated state information, respectively. We assume that the closed loop gains are preferable to the open loop gain, i.e. $\lambda_{\max}(\hat{\mathbf{A}}_i) < \lambda_{\max}(\mathring{\mathbf{A}}_i)$. Further note this model restricts its attention to wireless connections in uplink of the control loop, while downlink is assumed to occur over an ideal channel—i.e. no packet drops.

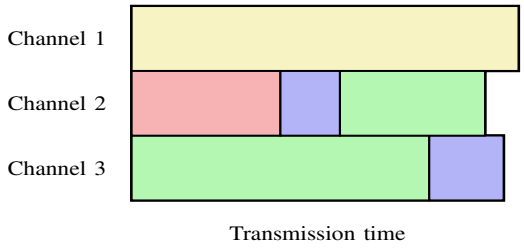

Fig. 2: Example of scheduling of $m = 4$ users colored in yellow, red, blue and green across $n = 3$ channels using TDMA in each channel.

Given this dynamical model of the wireless control systems, the communications goal is to allocate radio resources among the various systems to maintain strong performance across all the systems. To do so, we present a generic frequency and time division multiplexing scheduling architecture with which the BS allocates scheduling resources to the systems. A scheduling window occupies the uplink of a single cycle in the control loop in which each system has a single packet containing state information to transmit. For URLLC systems, the total length of this scheduling window is subject to a tight low-latency bound $t_{\max}$.

We assume that transmissions are scheduled by the BS across n available channels occupying different (possibly non-consecutive) frequency bands. Each channel is subject to continuous time division multiple access (TDMA), meaning that multiple transmissions in the same channel will occur in sequence. For full generality, we assume that a single device may be schedule the same packet in multiple channels in a single cycle to add redundancy and improve chance of success. Denote by $\boldsymbol{\varsigma}_i \in \{0,1\}^n$ a binary vector whose jth element $\varsigma_{i,j}$ is 1 if the ith device transmits in the jth channel, and 0 otherwise. Further denote for each device a data rate selection $\mu_i \in [\mu_{\min}, \mu_{\max}]$. These two scheduling parameters together define the scheduling decision made for the ith system. An illustration of $m = 4$ users making multiple transmission across $n = 3$ channels is shown in Figure 2.

The achieved communications performance by a given scheduling decision can be formulated as follows. We first define $\mathbf{h}_i^k \in \mathbb{R}_+^n$ to be the set of fading channel states experienced by device i at cycle k, where the j element $h_{i,j}^k$ is the fading channel gain in channel j. We assume that these channel conditions do not change over the course of a scheduling window. In any given channel with fading state $\mathbf{h}$, we define a function $q(h,\mu)$ that returns the packet delivery rate (PDR), or the probability of successful transmission of the packet, when transmitting with data rate μ. Likewise, we define a function $\tau(\mu)$ that returns the transmission time to transmit a packet of fixed length with data rate μ. These two functions play a critical role in designing low-latency wireless control systems, as they allow us to explore the trade-off between PDR and transmission time and the resulting effect on control system performance. We may consider that the functions $q(h,\mu)$ and $\tau(\mu)$ both get smaller as we increase

data rate μ, i.e.

$$\mu' > \mu \implies q(h,\mu) \leq q(h,\mu'), \quad \tau(\mu') \leq \tau(\mu). \quad (2)$$

Thus, by increasing the data rate we may reduce the transmission time to satisfy latency constraints, but at the cost of control system performance, as characterized by the switched dynamics in (1).

Remark 1: The communication architecture utilized here has a generic form that assumes both continuous time division and simultaneous transmission in independent, unsynchronized channels. We present the architecture in this form both for the purposes of a more tractable mathematical model as well as its generalization of the architectures used in, e.g., Bluetooth or centralized scheduled WiFi. Note that common OFDMA architectures, such as 5G [19] and next-generation WiFi IEEE 802.11ax [20], do not conform precisely to this architecture although it can be adapted as such with slight modifications. We leave the consideration of a synchronized, OFDMA architecture as a point of future work.

A. Optimal scheduling design

We are interested in designing scheduling policies that optimize control performance, subject to the strict low latency constraints of the system. To do so, we first formulate the global control-based performance given a scheduling decision. Collect in the matrix $\boldsymbol{\Sigma} \in \{0,1\}^{n\times m}$ all of the channel transmission vectors $\boldsymbol{\varsigma}_i$ for $i = 1,\dots,m$ and collect in the vector $\boldsymbol{\mu} \in [\mu_{\min},\mu_{\max}]^m$ the data rates μ_i for $i = 1,\dots,m$. Given that a device may transmit in multiple channels within a single scheduling cycle, the probability of successful transmission can be given as the probability that the transmission was successful in at least one channel, i.e.

$$\tilde{q}(\mathbf{h}_i,\boldsymbol{\varsigma}_i,\mu_i) := 1 - \prod_{j=1}^{n} \left(1 - \varsigma_{i,j} q(h_{i,j},\mu_i)\right). \quad (3)$$

The total delivery rate in (3) can be viewed as the probability of receiving the packet and experiencing the closed loop dynamics in (1). Now, to evaluate the performance of a given system at a particular state $\mathbf{x}$, define a quadratic Lyapunov function $L_i(\mathbf{x}) := \mathbf{x}^T\mathbf{P}_i\mathbf{x}$ with some positive definite matrix $\mathbf{P}_i \in \mathbb{R}^{p\times p}$. Such a function can be used to evaluate performance or stability of the control system. Because the control system evolves in a random manner, the cost of a given scheduling decision $\{\boldsymbol{\varsigma}_i,\mu_i\}$ for the ith system can be formulated as the *expected future Lyapunov cost* under such a schedule. As the probability of closing the loop in (1) is given by $\tilde{q}(\mathbf{h}_i^k,\boldsymbol{\varsigma}_i,\mu_i)$, we may write this expected future cost as

$$\begin{aligned} J_i(\mathbf{x}_i,\mathbf{h}_i,\boldsymbol{\varsigma}_i,\mu_i) :=\ & \mathbb{E}\left[L_i(\mathbf{x}_i^{k+1}) \mid \mathbf{x}_i^k = \mathbf{x}_i, \mathbf{h}_i^k = \mathbf{h}_i\right] \quad (4)\\ =\ & \tilde{q}(\mathbf{h}_i,\boldsymbol{\varsigma}_i,\mu_i)(\hat{\mathbf{A}}_i\mathbf{x}_i)^T\mathbf{P}_i(\hat{\mathbf{A}}_i\mathbf{x}_i) \ + \\ & (1-\tilde{q}(\mathbf{h}_i,\boldsymbol{\varsigma}_i,\mu_i))(\mathring{\mathbf{A}}_i\boldsymbol{\xi})^T\mathbf{P}_i(\mathring{\mathbf{A}}_i\mathbf{x}_i) \\ & + \ \mathrm{Tr}(\mathbf{P}_i\mathbf{W}_i). \end{aligned}$$

Observe that the local control cost for the ith system $J_i(\mathbf{x}_i^k,\mathbf{h}_i^k,\boldsymbol{\varsigma}_i,\mu_i)$ is a function of both the system *states*—the fading channel $\mathbf{h}_i^k$ and control state $\mathbf{x}_i^k$—and the scheduler *actions*—channel selection $\boldsymbol{\varsigma}_i$ and data rate μ_i. The objective is to choose the actions $\boldsymbol{\varsigma}_i$ and μ_i that minimizes the cost relative to states $\mathbf{h}_i^k$ and $\mathbf{x}_i^k$.

In addition to minimizing a control cost, we must make scheduling decisions that respect the low-latency requirements of the system. To formulate this constraint, consider the *total* time of a global scheduling decision $\boldsymbol{\Sigma},\boldsymbol{\mu}$ of channel j as the sum of all active transmissions, i.e.

$$\tilde{\tau}_j(\boldsymbol{\Sigma},\boldsymbol{\mu}) := \sum_{i=1}^{m} \varsigma_{i,j}\tau(\mu_i). \quad (5)$$

Combining all the local costs for systems $i = 1,\dots,m$ in (4) with the a constraint on the latency costs for all channels $j = 1,\dots,n$ in (5), we may define the optimal scheduling design problem. Because we are interested in long-term, or average, performance across random channels and control states, we optimize with respect to expected costs and probabilistic constraints. Collect all channel vectors $\mathbf{h}_i$ in a matrix $\mathbf{H} \in \mathbb{R}_+^{n\times m}$ and states $\mathbf{x}_i$ in a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$. Consider a scheduling policy $\mathbf{p}(\mathbf{H},\mathbf{X}) := \{\boldsymbol{\Sigma},\boldsymbol{\mu}\}$ that, given a set of channel states $\mathbf{H}$ and control states $\mathbf{X}$, returns a schedule defined by the channel selection matrix $\boldsymbol{\Sigma}$ and data rate selection vector $\boldsymbol{\mu}$. The optimal low-latency constrained scheduling policy for the wireless control systems is the one which solves the program

$$\begin{aligned} J^* := \min_{\mathbf{p}(\mathbf{H},\mathbf{X})}\ & \mathbb{E}_{\mathbf{H},\mathbf{X}}\left[\sum_{i=1}^{m} J_i(\mathbf{x}_i,\mathbf{h}_i,\boldsymbol{\varsigma}_i,\mu_i)\right], \quad (6)\\ \text{s.t.}\quad & \mathbb{P}_{\mathbf{H},\mathbf{X}}\left(\tilde{\tau}_j(\boldsymbol{\Sigma},\boldsymbol{\mu}) \leq t_{\max}\right) \geq 1-\delta \quad j = 1\dots,n,\\ & \mathbf{p}(\mathbf{H},\mathbf{X}) := \{\boldsymbol{\Sigma} \in \{0,1\}^{n\times m}, \boldsymbol{\mu} \in [\mu_{\min},\mu_{\max}]^m\}. \end{aligned}$$

In (6), we minimize the average cost over the distribution of channel and control states, subject to the condition that the probability of violating the latency constraint over the distribution of states is less than some small value δ. Because each channel's transmission time varies, we impose this constraint independently for *each channel*. The above scheduling problem can be viewed as a constrained statistical learning problem [16]. While such a problem characterizes the optimal scheduling decision for the latency-constraint wireless control system, finding solutions to such a problem is a significant challenge. This is due to a number of complexities in (6), namely: (i) it requires functional optimization, (ii) it contains explicit constraints, and (iii) we typically do not have analytic forms for the functions and distributions in (6). The first of these complexities can be resolved using a standard technique in statistical learning, discussed next in Section II-B. The latter two of these complexities are discussed and resolved later in Sections III and III-A, respectively.

B. Deep learning parameterization

The scheduling problem in (6) is computationally challenging because it requires finding a policy—or *function*—$\mathbf{p}(\mathbf{H},\mathbf{X})$. In statistical learning, or regression, problems the

regression function is replaced by some given parameterization $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$ that is defined with some finite dimensional parameter $\boldsymbol{\theta} \in \mathbb{R}^q$. There exist a wide variety of choices of this parameterization, but in modern machine learning problems the *deep neural network (DNN)* is commonly employed. This is due to the fact the DNN can be shown both empirically and analytically to contain strong representative power and generalization ability, meaning that it can approximate almost any function well. A DNN is defined as a composition of L layers, each of which consisting of a linear operation followed by a point-wise nonlinearity—also known as an activation function. More specifically, the layer l is defined by the linear operation $\mathbf{W}_l \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\sigma_l : \mathbb{R}^{q_l} \to \mathbb{R}^{q_l}$. Common choices of activation functions σ_l include a sigmoid function or a rectifier function (commonly referred to as ReLu).

Given an input from the $l-1$ layer $\mathbf{w}_{l-1} \in \mathbb{R}^{q_{l-1}}$, the resulting output $\mathbf{w}_l \in \mathbb{R}^{q_l}$ is then computed as $\mathbf{w}_l := \sigma_l(\mathbf{W}_l \mathbf{w}_{l-1})$. The full DNN-parameterization of the scheduling policy is then defined as an L-layer DNN whose input at the initial layer is the concatenation of states $\mathbf{w}_0 := [\text{vec}(\mathbf{H}); \text{vec}(\mathbf{X})]$, i.e.

$$\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta}) := \sigma_L(\mathbf{W}_L(\sigma_{L-1}(\mathbf{W}_{L-1}(\dots(\sigma_1(\mathbf{W}_1 \mathbf{w}_0)))))). \tag{7}$$

The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^q$ that defines the DNN is then the entries of $\{\mathbf{W}_l\}_{l=1}^L$ with $q = \sum_{l=1}^{L-1} q_l q_{l+1}$. Further note that we can easily construct an activation function at the final layer σ_L—or the *output layer*—such that the outputs $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$ are in the space $\{0,1\}^{n \times m} \times [\mu_{\min}, \mu_{\max}]$ that contains possible schedules. With this DNN parameterization, the control-aware scheduling problem can be rewritten as

$$\begin{aligned} J^*_{\boldsymbol{\phi}} := \min_{\boldsymbol{\theta} \in \mathbb{R}^q} \mathbb{E}_{\mathbf{H},\mathbf{X}} \left[\sum_{i=1}^m J_i(\mathbf{x}_i, \mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i) \right], \\ \text{s.t.} \quad \mathbb{P}_{\mathbf{H},\mathbf{X}} \left(\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \leq t_{\max} \right) \geq 1 - \delta \qquad \forall j, \\ \boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta}) := \{\boldsymbol{\Sigma} \in \{0,1\}^{n \times m}, \boldsymbol{\mu} \in [\mu_{\min}, \mu_{\max}]^m\}. \end{aligned} \tag{8}$$

Observe in (8) that the optimization is performed over $\boldsymbol{\theta}$ rather than the scheduling policy directly. In other words, we look for the interlayer weights that define a DNN that minimizers the total control cost while satisfying the latency constraints. We proceed then to discuss a learning method that can find solutions to the constrained optimization problem in (8).

III. Primal-Dual Learning

Finding the DNN layer weights $\boldsymbol{\theta}$ that provide good solutions to (8) requires the solving of a constraint learning problem. The standard approach of gradient-based optimization methods cannot be applied directly here due to the presence of the latency constraints. To proceed then, we must formulate an unconstrained problem that captures the form of (8). A naive penalty-based reformulation will introduce a similar but fundamentally different problem, so we thus opt for constructing a Lagrangian dual problem. For notational convenience, moving forward we employ the following shorthands for the state variables, aggregate Lyapunov function, latency constraint functions, respectively:

$$\mathbf{w} := [\text{vec}(\mathbf{H}); \text{vec}(\mathbf{X})], \tag{9}$$

$$f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) : = \sum_{i=1}^m J_i(\mathbf{x}_i, \mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i), \tag{10}$$

$$g_j(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) := \mathbb{I}\left[\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \leq t_{\max}\right] - (1 - \delta) \tag{11}$$

We introduce the nonnegative dual variables $\boldsymbol{\lambda} \in \mathbb{R}^n_+$ associated with the vector of constraint functions $\mathbf{g}(\mathbf{p}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) := [g_1(\cdot); \dots; g_n(\cdot)]$, and form the Lagrangian as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}) := \mathbb{E}_{\mathbf{w}} \left[f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) - \boldsymbol{\lambda}^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) \right]. \tag{12}$$

The Lagrangian in (12) penalizes constraint violation through the second term. Note, however, that the penalty is scaled by the dual parameter $\boldsymbol{\lambda}$. The so-called Lagrangian dual problem is one in which both the primal variable $\boldsymbol{\theta}$ is simultaneously minimized while the dual parameter $\boldsymbol{\lambda}$ is maximized. Such a problem can be written with the saddle point formulation

$$D^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}). \tag{13}$$

The dual optimum $D^*_{\boldsymbol{\phi}}$ is the best approximation of the form in (12) we can have of $J^*_{\boldsymbol{\phi}}$. In fact, under some standard assumptions on the problem and assuming a sufficiently dense DNN architecture, we can formally bound the difference between $D^*_{\boldsymbol{\phi}}$ and J^* to be proportional to the approximation capacity of the DNN $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$—see [16] for details on this result. Thus, we may say that, up to some approximation, solving the unconstrained problem in (13) is equivalent to solving the constrained problem in (8).

With the unconstrained saddle point problem in (13), we may perform standard gradient-based optimization methods to obtain solutions. The max-min structure necessitates the use of a *primal-dual* learning method, in which we iteratively update both the primal and dual variable in (12) to find a local stationary point of the KKT conditions of (8). Consider a learning iteration index $t = 0, 1, \dots$ over which we define a sequence of primal variables $\{\boldsymbol{\theta}_t\}$ and dual variables $\{\boldsymbol{\lambda}_t\}$. At index t, we determine the value of next primal iterate $\mathbf{x}_{t+1}$ by adding to the current iterates the corresponding partial gradients of the Lagrangian in (12) $\nabla_{\boldsymbol{\theta}} \mathcal{L}$, i.e.,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{w}} \left[f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) - \boldsymbol{\lambda}_t^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) \right], \tag{14}$$

where we introduce $\alpha_t > 0$ as a scalar step size. We subsequently perform a corresponding partial gradient update to compute the dual iterate $\boldsymbol{\lambda}_{t+1}$, i.e.

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t - \beta_t \mathbb{E}_{\mathbf{w}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_{t+1}), \mathbf{w}) \right]_+, \tag{15}$$

with associated step size $\beta_t > 0$. Observe in (15) that we additionally project onto the positive orthant to maintain the nonnegative constraint on $\boldsymbol{\lambda}$. The gradient primal-dual updates in (14) and (15) successively move the primal and dual variables towards maximum and minimum points of the Lagrangian function, respectively.

A. Model-free updates

The updates in (14)-(15) cannot, in general, be applied exactly. To see this, observe that computing the gradients in (14) requires computing the gradient of $J_i(\cdot)$—which depends on PDR function $\tilde{q}(\cdot)$ and system dynamics—and the gradient of an indicator of transmission length function $\tilde{\tau}(\cdot)$. In practical systems, we do not typically have easily available analytic forms for these functions to take gradients. Furthermore, both the updates in (14) and (15) require to take the expectation over the distribution of states $\mathbf{x}$ and $\mathbf{h}$. These, too, are often unknown in practice. However, there exist standard ways of approximating the updates with stochastic, *model-free* updates that do not require such knowledge. Most popular among these is the policy gradient approximation [21].

To compute a policy gradient update, we consider the scheduling parameters $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are drawn stochastically from a distribution with given form $\pi_{\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta})}$ whose parameters are given by the output of the DNN $\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta})$—e.g. the mean and variance of a normal distribution. Using such a stochastic policy, it can be shown that an unbiased estimators of the gradients in (14) and (15) can be formed as,

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{w}} f(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}),\mathbf{w}) = f(\hat{\mathbf{p}}_{\boldsymbol{\theta}},\hat{\mathbf{w}})\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}) \quad (16)$$

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{w}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}),\mathbf{w}) = \mathbf{g}(\hat{\mathbf{p}}_{\boldsymbol{\theta}},\hat{\mathbf{w}})\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}(\hat{\mathbf{p}}_{\boldsymbol{\theta}})^T \quad (17)$$

$$\widehat{\mathbb{E}_{\mathbf{w}}}\mathbf{g}(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}),\mathbf{w}) = \mathbf{g}(\hat{\mathbf{p}}_{\boldsymbol{\theta}},\hat{\mathbf{w}}), \quad (18)$$

where $\hat{\mathbf{w}}$ is a sampled state and $\hat{\mathbf{p}}_{\boldsymbol{\theta}}$ is a sample drawn from the distribution $\pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}$. In practice, we may reduce the variance of these unbiased estimates by taking B samples and averaging. Note that the updates here only require taking gradients of the log likelihoods rather than of the functions themselves. This implies we can perform the learning process without explicitly knowing, e.g., system dynamics, performance metrics, state distributions. Thus, we can replace the updates in (14) and (15) with their model free counterparts by substituting the gradient estimates in (16)-(18). The complete primal-dual learning algorithm is summarized in Algorithm 1. We conclude with a brief remark on state sampling.

Algorithm 1 Model-Free Primal-Dual Learning

1: **Parameters:** Policy model $\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta})$ and distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}$

2: **Input:** Initial states $\boldsymbol{\theta}_0, \boldsymbol{\lambda}_0$

3: **for** $t = 0, 1, 2, \ldots$ **do** {main loop}

4: Draw samples $\{\hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}\}$, or in batches of size B

5: Compute policy gradients [c.f. (16)-(18)]

6: Update primal and dual variables

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{w}}\left[f(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}_t),\mathbf{w}) - \boldsymbol{\lambda}_t^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}_t),\mathbf{w})\right], [cf.(14)]$$

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t - \beta_t \widehat{\mathbb{E}_{\mathbf{w}}}\mathbf{g}(\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta}_{t+1}),\mathbf{w})\right]_+ [cf.(15)]$$

7: **end for**

Remark 2: In the gradient estimations in, e.g. (16), we sample both the control states $\mathbf{x}$ and channel states $\mathbf{h}$. This assumes that such samples can be drawn i.i.d. While this may generally be true for the channel states $\mathbf{h}$, it will not be generally be true for the control states $\mathbf{x}$ in practice, due to the fact that the states evolve based on the switched dynamics in (1), which itself depends on the scheduling actions taken. A more precise way to model the statistics of the control states would be with a Markov decision process (MDP). The generalization of the presented techniques for this setting make up what is known as *reinforcement learning* algorithms. In this work, we nonetheless assume that *during training* that $\mathbf{x}$ can also be drawn i.i.d. from an approximate distribution. As shown in the proceeding section, we demonstrate that the learned policy is still effective when the states evolve via (1) as would be seen in practice.

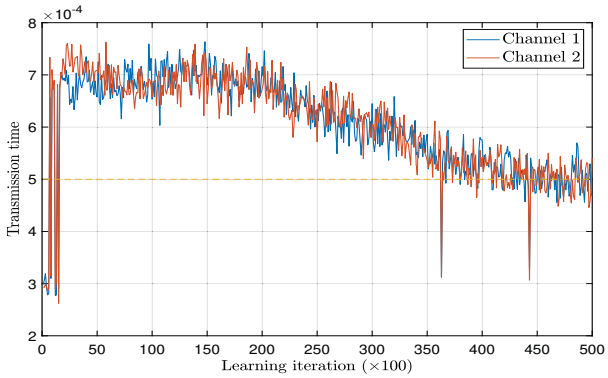

Fig. 3: Convergence of transmission time for a low-latency, control aware scheduling policy over the learning process. The DNN parameterized scheduling policy obtains feasible latency-contained schedules ($t_{\max} = 5 \times 10^{-4}$ shown in dashed red line) on both channels.

IV. Simulation Results

We perform a series of simulations on latency-constrained wireless control systems to evaluate the performance the learning method in and the resulting control-aware scheduling policies. We generate a series $m = 9$ systems with closed-loop gains $\hat{\mathbf{A}}_i \sim \text{Uniform}(0.85, 0.95)$ and open-loop gains $\mathring{\mathbf{A}}_i \sim \text{Uniform}(1.01, 1.2)$. The variance for all system noise $\mathbf{w}_i$ is set to be $W = 1$. All such systems send their state information over a shared wireless channel with $n = 2$ independent channels with a total latency constraint of $t_{\max} = 0.5$ ms. A latency bound of this order is typical of industrial control systems such as printing machines and presses [1]. We further assume that the states of the systems are confined to the box $[-10, 10]$. In simulations, we utilize a DNN with 2 layers of size 2000 and 1000, with ReLU activation functions. For the policy distribution $\pi_{\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta})}$, we utilize a Beta distribution scaled between $[1.6, 9.0]$ to select $\boldsymbol{\mu}$ and a Bernoulli distribution to select $\boldsymbol{\Sigma}$.

With the scheduling architecture given in Figure 2 for $n = 2$ channels and $m = 9$ systems, at a control scheduling interval each system is given a data rate μ_i and a set of channels to

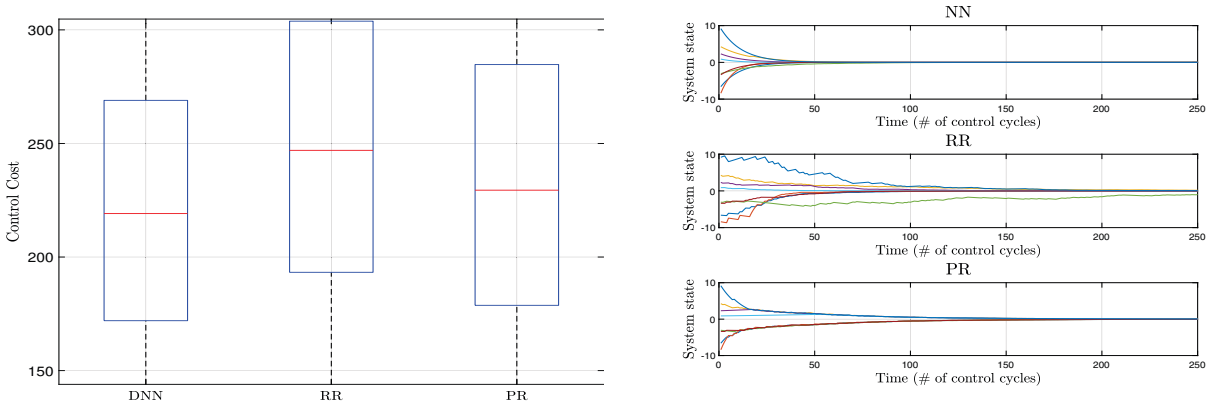

Fig. 4: (Left) A boxplot of medians and quartiles of control costs obtained by proposed DNN-based scheduling and model-free heuristics. In terms of the quadratic system cost, the DNN outperforms both baselines. In (Right) we simulate the control system using the learned scheduler and two baseline model-free heuristic scheduler. The NN policy stabilizes most of the systems faster than RR and PR.

transmit on. In our simulations, we use the modulation and coding schemes (MCS) of the next-generation IEEE 802.11ax Wi-Fi protocol as a representative architecture for data rate selection and packet error rate computation. As such, the continuous data rates μ_i are selected in an interval of $[1.6, 10]$ and rounded down to the nearest discrete MCS selection given in 802.11ax—see [20] for details on the MCS tables given in this protocol. The corresponding transmission time $\tau(\mu)$ is then calculated assuming a fixed packet size of 100 bytes and the packet delivery rate $q(h, \mu)$ is computed using the associated AWGN error curve (scaled by the effective SNR given channel conditions).

In Figure 3 we show the training process for the control-aware scheduler using the primal-dual scheduler given in Algorithm 1. In particular, we show the transmission time utilized over the 2 channels by the NN-based scheduling policy over the course of $50,000$ learning iterations. As can be seen, the policies converge to scheduling decisions that respect latency requirements for both channels after $50,000$ iterations, showcasing the capability of a neural network to learn latency restrictive policies.

We proceed to compare the performance of the learned policy in terms of the control metric in (4) against other scheduling heuristics. We compares against a standard, control-agnostic round-robin scheduling policy (RR) and a control-aware priority ranking (PR) heuristic in which transmissions are scheduled iteratively for systems based on control state value. These methods are chosen as they can be implemented *model-free*, to make a reasonable comparison against the model-free DNN, and are those utilized in modern practical systems. We point out that both of the scheduling policies used fixed target PDRs of 0.95 to determine data rate selection. In the left image of Figure 4 we show a box plot of the quadratic control costs obtained by each of these methods over 1000 different randomly drawn plant and channel states. It can be observed that, in terms of this cost, the DNN outperforms both baselines. Thus, when considering a specific control-aware cost to optimize, designing scheduling algorithms directly with respect to this cost can benefit the performance of the system.

Alternatively to the quadratic cost shown in the left image, we may observe the end system performance of each of the scheduling methods by looking at the state evolution of each of the 9 plants using the respective schedulers—see Remark 2. In the right image of Figure 4 we show evolution of the 9 systems under each method. It can be observed that, while all systems stabilize using each of the three schedulers, the DNN is overall able to draw the plant states to zero faster than the other methods, with one exception. Together, the results in both figures of Figure 4 demonstrate an improved performance relative to existing baselines. This can be attributed to the fact that DNN has been model-free trained to adapt to both changing channel conditions *and* the individual dynamics and states of each of the systems, which allows it to make more efficient scheduling policies with regards to the varying system dynamics and latency constraints. In these results we observe that it is indeed advantageous to incorporate control system knowledge in the scheduling decision to promote good performance.

V. Conclusion

We consider the setting of scheduling for low-latency wireless control systems. To handle the challenge of achieving high reliability performance with limited scheduling resources, we formulate a control-aware scheduling problem in which reliability is adapted to control and channel states. This problem takes the form of a constrained statistical learning problem, in which solutions can be found by parameterized the scheduling policy with a deep neural network and finding optimal weights with a primal-dual learning algorithm that can be implemented without system or dynamical models. Numerical simulations

showcase DNN-based scheduling policies that outperform baseline scheduling procedures.

REFERENCES

[1] Shehzad A Ashraf, Ismet Aktas, Erik Eriksson, Ke Wang Helmersson, and Junaid Ansari, "Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G," in *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*. IEEE, 2016, pp. 1–8.

[2] Petar Popovski, Jimmy J Nielsen, Cedomir Stefanovic, Elisabeth de Carvalho, Erik Strom, Kasper F Trillingsgaard, Alexandru-Sabin Bana, Dong Min Kim, Radoslaw Kotaba, Jihong Park, et al., "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, 2018.

[3] Mehdi Bennis, Mérouane Debbah, and H Vincent Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *arXiv preprint arXiv:1801.01270*, 2018.

[4] Chengjie Wu, Mo Sha, Dolvara Gunatilaka, Abusayeed Saifullah, Chenyang Lu, and Yixin Chen, "Analysis of edf scheduling for wireless sensor-actuator networks," in *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*. IEEE, 2014, pp. 31–40.

[5] Songwu Lu, Vaduvur Bharghavan, and Rayadurgam Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on networking*, vol. 7, no. 4, pp. 473–489, 1999.

[6] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Phil Whiting, and Rajiv Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications magazine*, vol. 39, no. 2, pp. 150–154, 2001.

[7] Vasuki Narasimha Swamy, Sahaana Suri, Paul Rigge, Matthew Weiner, Gireeja Ranade, Anant Sahai, and Borivoje Nikolić, "Cooperative communication for high-reliability low-latency wireless control," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4380–4386.

[8] Muhammad Ikram Ashraf, Chen-Feng Liu, Mehdi Bennis, Walid Saad, and Choong Seon Hong, "Dynamic resource allocation for optimized latency and reliability in vehicular networks," *IEEE Access*, vol. 6, pp. 63843–63858, 2018.

[9] Anton Cervin and Toivo Henningsson, "Scheduling of event-triggered controllers on a shared network," in *Proc. of the 47th IEEE Conf. on Dec. and Control (CDC)*, 2008, pp. 3601–3606.

[10] Duo Han, Junfeng Wu, Huanshui Zhang, and Ling Shi, "Optimal sensor scheduling for multiple linear dynamical systems," *Automatica*, vol. 75, pp. 260–270, 2017.

[11] Konstantinos Gatsis, Miroslav Pajic, Alejandro Ribeiro, and George J. Pappas, "Opportunistic control over shared wireless channels," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3140–3155, December 2015.

[12] Burak Demirel, Arunselvan Ramaswamy, Daniel E Quevedo, and Holger Karl, "Deepcas: A deep reinforcement learning algorithm for control-aware scheduling," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 737–742, 2018.

[13] Alex S Leong, Arunselvan Ramaswamy, Daniel E Quevedo, Holger Karl, and Ling Shi, "Deep reinforcement learning for wireless sensor scheduling in cyber-physical systems," *arXiv preprint arXiv:1809.05149*, 2018.

[14] Vinicius Lima Silva, Mark Eisen, Konstantinos Gatsis, and Alejandro Ribeiro, "Optimal resource allocation in wireless control systems via deep policy gradient," *arXiv preprint arXiv:1910.11900*, 2019.

[15] Mark Eisen, Mohammad M Rashid, Konstantinos Gatsis, Dave Cavalcanti, Nageen Himayat, and Alejandro Ribeiro, "Control aware radio resource allocation in low latency wireless control systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7878–7890, 2019.

[16] Mark Eisen, Clark Zhang, Luiz FO Chamon, Daniel D Lee, and Alejandro Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, 2019.

[17] Anis Elgabli, Hamza Khan, Mounssif Krouka, and Mehdi Bennis, "Reinforcement learning based scheduling algorithm for optimizing age of information in ultra reliable low latency networks," *arXiv preprint arXiv:1811.06776*, 2018.

[18] Chengjian Sun and Chenyang Yang, "Unsupervised deep learning for ultra-reliable and low-latency communications," *arXiv preprint arXiv:1905.13014*, 2019.

[19] Mamta Agiwal, Abhishek Roy, and Navrati Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[20] Jianhan Liu, R Porat, N Jindal, et al., "Ieee 802.11 ax channel model document," *Wireless LANs, Rep. IEEE 802.11–14/0882r3*, 2014.

[21] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.