

Learning Markov models of fading channels in wireless control networks: a regression trees based approach

Luis Felipe Florenzan Reyes¹, Francesco Smarra¹, Yuriy Zacchia Lun², and Alessandro D’Innocenzo¹

Abstract—Finite-state Markov models are widely used for modeling wireless channels affected by a variety of non-idealities, ranging from shadowing to interference. In an industrial environment, the derivation of a Markov model based on the wireless communication physics can be prohibitive as it requires a complete knowledge of both the communication dynamics parameters and of the disturbances/interferers. In this work, a novel methodology is proposed to learn a Markov model of a fading channel via historical data of the signal-to-interference-plus-noise-ratio (SINR). Such methodology can be used to derive a Markov jump model of a wireless control network, and thus to design a stochastic optimal controller that takes into account the interdependence between the plant and the wireless channel dynamics. The proposed method is validated by comparing its prediction accuracy and control performance with those of a stationary finite-state Markov chain derived assuming perfect knowledge of the physical channel model and parameters of a WirelessHART point-to-point communication based on the IEEE-802.15.4 standard.

I. INTRODUCTION

Wireless networked control systems (WNCSs) are composed of spatially distributed sensors, actuators, and controllers communicating through wireless networks [1]. Despite their success in industrial monitoring applications, existing wireless sensor-actuator network technologies face significant challenges in supporting control systems due to their lack of real-time performance and dynamic wireless conditions in industrial plants [2]. A key challenge in WNCSs design is the channel modeling in an industrial environment because of its inherent complexity [2] [3]. These communication channels are frequently subject to time-varying fading and interference, which may lead to packet losses.

From the automatic control perspective, an example of WNCS consists of a nonlinear process with intermittent control packets due to the lossy communication channel described by the following equations:

$$\begin{cases} y(k+1) = f(y(k), u_a(k)), k \in \mathbb{N} \\ u_a(k) = \nu(k)u(k) \\ y(0) = y_0 \in \mathbb{R}^{n_y} \end{cases} \quad (1)$$

where $y(k) \in \mathbb{R}^{n_y}$ is the output of the system and $u(k) \in \mathbb{R}^{n_u}$ is the input signal. $\{\nu(k)\}_{k \in \mathbb{N}}$ is a discrete-time Boolean process modeling the packet delivery of the control signals: if the packet is correctly delivered then $\nu(k) = 1$, otherwise if it is lost then the actuator does nothing $\nu(k) = 0$. The control packet error probability (PEP) $\mathbb{P}(\nu(k) = 0)$, depends on the signal-to-interference-plus-noise-ratio (SINR) of the communication $\Gamma(k)$, i.e.,

$$\mathbb{P}(\nu(k) = 0) = R_p(\Gamma(k)), \quad (2)$$

The research leading to these results has received funding from the Italian Government under CIPE resolution n.135 (Dec. 21, 2012), project INnovating City Planning through Information and Communication Technologies (INCIPICT).

¹ Department of Information Engineering, Computer Science and Mathematics, University of L’Aquila, 67100 AQ, Italy. luisfelipe.florenzanreyes@graduate.univaq.it, francesco.smarra@univaq.it, alessandro.dinnocenzo@univaq.it
² IMT School for Advanced Studies Lucca, 55100 LU, Italy. yuriy.zacchialun@imtlucca.it

where $R_p : \mathbb{R} \rightarrow [0, 1]$ is a deterministic function defined by the communication standard. We assume full state observation with no measurement noise, and no observation packet loss. In this work, we consider a channel model based on WirelessHART [4], a on-the-market wireless communication standard specifically designed for process automation. We assume that the value of the SINR $\Gamma(k)$ is measurable and is sent to the controller via an acknowledgment (ACK). However, this ACK is available only after the current decision on the control input to apply has been made and sent through the link, since the actual success of the transmission is not known in advance. Hence, at each time k the controller receives measurements of $y(k)$ and $\Gamma(k-1)$, as depicted in Fig. 1.

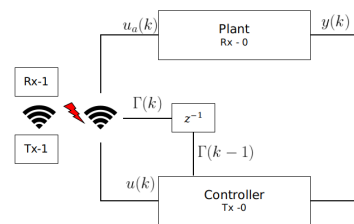


Fig. 1: Wireless networked control system

Since $\Gamma(k)$ is represented by a generic stochastic process, the obtained model may not be computationally tractable when it is derived for wireless communications in industrial environment (see Sec. II), especially if the objective is to apply optimal control algorithms (e.g. model predictive control – MPC). For this reason, a preliminary investigation of channel model abstraction is fundamental to design a controller. In the WNCS literature the packet dropouts have been modeled either as stochastic or deterministic phenomena. For what concerns stochastic models, a vast amount of research assumes memoryless packet drops, so that dropouts are realizations of a Bernoulli process. In [5], the packet delivery process is modeled as a Bernoulli random process, then an optimal controller is derived. In [6] a Markov chain model is used to derive an accurate abstraction of the WirelessHART channel, and it is proven that such model allows to characterize the stability of a WNCS for the scenarios where the simple Bernoulli-like channel models (which cannot model packet bursts) fail.

Markov models are a powerful tool for modeling stochastic random processes. They are general enough to model with high accuracy a large variety of processes and are relatively simple, allowing us to compute analytically many important parameters of the process which are very difficult to calculate for other models [7]. Hidden Markov models ([8] [9]) have been exploited to learn channels models in [3] and [10]. In [11], the authors expose the benefit of exploiting finite-state Markov chains to model the behavior of wireless fading channels. In [12], the authors derive a Markov chain that estimates the packet error probability (PEP) of an industrial wireless protocol, and then propose an optimal stochastic controller for linear systems.

Inspired by the challenges in [12] related to the derivation

of a physics-based Markov chain abstraction of the wireless channel, the main contribution of this paper is a novel data-driven methodology, based on regression trees [13], to identify such Markov chain abstraction. More precisely, we propose a novel methodology to model the PEP $\nu(k)$ on the basis of a Markov chain $\theta(k)$: each state of $\theta(k)$ is associated to a partition of \mathbb{R}^{n_y+1} consisting of rectangular sets $\{R_i\}_{i=1}^{\ell}$, each representing the range of possible values of $\Gamma(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}^{n_y}$ at time k . We construct the transition probability matrix (TPM) of $\theta(k)$ as follows:

$$\begin{aligned} p(j | i) &\doteq \mathbb{P}(\theta(k+1) = j | \theta(k) = i) \\ &= \mathbb{P}(\theta(k+1) = j | (\Gamma(k), y(k)) \in R_i) \\ &= h(\Gamma(k), y(k)), \end{aligned} \quad (3)$$

where identifying from historical data the function $h : \mathbb{R}^{n_y+1} \rightarrow [0, 1]$, which depends on the current measurements $\Gamma(k)$ and $y(k)$, is the objective of this paper. Note that, given any two states i, j of the Markov chain, $p(j | i)$ also depends on the plant output: indeed, $y(k)$ may for example determine the distances between the transmitter, the receiver and the interferer, and, as illustrated in Sec. II, this strongly affects the dynamics of $\Gamma(k)$. Finally, the PEP can be easily computed using (2), which in this paper is based on a point-to-point WirelessHART communication based on the IEEE 802.15.4 standard. Nevertheless, we remark that our data-driven methodology is independent on the transmission technology and can be replicated for any communication protocol if the corresponding wireless channel can be effectively modeled by a Markov chain.

In summary, in an industrial environment the derivation of a Markov model based on the wireless communication physics can be prohibitive as it requires a complete knowledge of both the communication dynamics parameters and of the disturbances/interferers. Our methodology has 3 main advantages: (1) it only exploits historical data, hence does not require any a-priori knowledge of the system and channel parameters. Moreover, most of physics based approaches cannot handle time-varying parameters, as the computational complexity of the obtained abstraction would be intractable, while with our methodology we consider a dependency between the parameters of the communication system and the dynamics of the plant which is still tractable in terms of computational complexity; (2) as a byproduct of leveraging our techniques in [14], and beyond the main contribution of identifying the Markov chain abstraction of $\Gamma(k)$, we also construct a switching auto-regressive exogenous (SARX) model for the nonlinear plant f in (1), i.e. our methodology does not require any linearity assumption on the plant's dynamics; (3) the obtained identified models, both for $\Gamma(k)$ and f , can be combined obtaining a Markov jump system, which can be directly used to setup a classical MPC problem that can be solved very efficiently, i.e. using quadratic programming (QP).

II. CHANNEL MODELING

In this section we first describe the channel model under consideration, then we present the Markovian model for the wireless link developed in [12], and finally we introduce an analytical model that will be useful to experiment the data-driven approach we propose in this paper.

1) *WirelessHART physical model*: We analyze the industrial environment described in Fig. 1, wherein the wireless communication is affected by interference. We study a point to point transmission based on the WirelessHART protocol, i.e. the IEEE 802.15.4-2006 defined in [15], interfered by another WirelessHART transmission. The proposed model considers the effect of path loss, the shadow fading, and the

residual power fluctuations left by the power control. The effect of multipath fading is supposed to be compensated by the aforementioned power control. We denote with $i=0$ the reference Transmitter-Receiver (Tx-Rx) link, and with $i=1$ the link between the interferer Tx and the reference Rx.

The effect of the path loss model is defined in [15]. For a system with bandwidth $W=2.4$ GHz the path loss coefficient of the link between transmitter i and the reference receiver (i.e., a mobile plant in our case) is $\alpha_i(k) = 10^{-\frac{\zeta(d_i(k))}{10}}$, where

$$\zeta(d_i(k)) = \begin{cases} 40.2 + 20 \log_{10}(d_i(k)), & \text{if } d_i(k) \leq 8, \\ 58.5 + 33 \log_{10}\left(\frac{d_i(k)}{8}\right), & \text{otherwise;} \end{cases} \quad (4)$$

and $d_i(k)$ is the length of the link i at time instant k , i.e. a distance in meters that may e.g. depend on the position of the plant (see the inverted pendulum on a cart in Sec. VI).

The shadow fading is modeled following [16] by assuming a log-normal model for each link i , which introduces a multiplicative factor $e^{\beta_i(k)}$, where $\beta_i(k)$ is a zero-mean Gaussian process with variance $\sigma_{\beta_i}^2$ and auto-covariance function $c_{\beta_i}(\tau)$, with τ being a time lapse between two consecutive (time-driven) control packets. We remark that $c_{\beta_i}(\tau)$ may also depend on the state of the plant (e.g. the speed of a cart) thus exhibiting a time-varying behavior.

For each link i the residual power control error (PCE) is also modeled as a log-normal process, $e^{\xi_i(k)}$, where $\xi_i(k)$ is a zero-mean Gaussian process with variance $\sigma_{\xi_i}^2$ and auto-covariance $c_{\xi_i}(\tau)$.

By considering the characteristics of OQPSK-DSSS (offset quadrature phase-shift keying – direct-sequence spread spectrum) modulation, as specified in [15], we can derive the power value of the SINR, $\gamma(k)$, at time k as in [12]:

$$\gamma(k) = \sqrt{\frac{P_0(k)\alpha_0^2(k)e^{\beta_0(k)+\xi_0(k)}}{\frac{N_0}{4} + \frac{8}{3G}P_1(k)\alpha_1^2(k)e^{\beta_1(k)+\xi_1(k)}}}, \quad (5)$$

where P_0 and P_1 are respectively the reference user transmission power and the interferer transmission power, N_0 is the noise spectral density, and $G = WT_s$ is the processing gain, with T_s being the symbol time. In the rest of the work we denote as $\Gamma(k) = 10 \log_{10}(\gamma(k))$ the SINR in decibel.

In WirelessHART the forward error correction is not implemented, thus even one erroneous bit leads to a corrupted WirelessHART data packet. For this reason, the packet error probability R_p is related to the SINR as follows

$$\begin{aligned} R_b(\gamma(k)) &= \frac{1}{30} \sum_{\ell=2}^{16} (-1)^\ell \binom{16}{\ell} \exp\left(20 \cdot \gamma(k) \frac{1-\ell}{\ell}\right) \\ R_p(\gamma(k)) &= 1 - (1 - R_b(\gamma(k)))^{l_f}, \end{aligned} \quad (6)$$

where l_f is the number of bits in a frame, and R_b is the bit error rate computed according to [15]. It is worth remarking that the distance Tx-Rx $d_i(k)$ influences the SINR $\gamma(k)$ that, in turn, influences the packet error rate. This will be useful for the discussion of the simulation results in Sec. VI.

The main issue with the above channel model is that, despite the ability to describe the SINR, in the majority of the control applications equation (5) is intractable, both in the continuous-time and discrete-time form. For this reason, in [12], a Markov chain model to derive a discrete-time abstraction of (5) has been proposed. We recall such technique in the following subsection as a comparison for the method we propose in this paper.

2) *Finite-state Markov chain*: It is straightforward to see that (5) can be expressed as a weighted linear combination of correlated log-normal processes: hence there is no exact explicit closed form expression of its distribution. We can use the moment matching technique [17] to approximate the probability distribution of (5) by a log-normal process, thus presenting $\Gamma(k)$ as a Gaussian process with mean $\mu_\Gamma(k)$, variance $\sigma_\Gamma^2(k)$ and auto-covariance $c_\Gamma(\tau)$, as detailed in [12]. Clearly, due to the dependence of the SINR on the state of a mobile plant through, e.g. the parameters $d_i(k)$ (which define the path loss coefficients $\alpha_i(k)$ via (4)), the moment matching approximation should be done for each relevant value of the aforementioned parameters. In the rest of this subsection we will focus on a stationary Gaussian process $\hat{\Gamma}(k)$, which is a moment matching approximation of the SINR with arbitrary values of the parameters in (5) corresponding to any given state of the plant, i.e. for $\alpha_i(k) = \hat{\alpha}_i$, $\beta_i(k) = \hat{\beta}_i(k)$, and $P_i(k) = \hat{P}_i$ in (5), with $i=0, 1$, we have that $\hat{\Gamma}(k) \approx 10 \log_{10}(\gamma(k))$.

At this point, to obtain a finite-state Markov channel abstraction, the first step is to divide the range of $\hat{\Gamma}(k)$ into several consecutive regions, each associated with a certain representative PEP. Specifically, a region r of the values of the SINR is mapped into a state s_r of the related Markov chain, and it is delimited by two thresholds ζ_r and ζ_{r+1} belonging to the set of extended reals. These SINR thresholds are determined by the chosen partitioning method [11]. In this paper, we rely on a well-known equiprobable partitioning, where the thresholds are selected in such a way that the steady state probabilities of being in any state are equal. Then, the steady state probability p_r of a state s_r is defined as the probability that the value of $\hat{\Gamma}(k)$ is between the two thresholds of the region, and it is given by

$$p_r = \int_{\zeta_r}^{\zeta_{r+1}} \phi_{\mathcal{N}}(\zeta; \mu_{\hat{\Gamma}}, \sigma_{\hat{\Gamma}}^2) d\zeta, \quad (7)$$

where $\mu_{\hat{\Gamma}}$ and $\sigma_{\hat{\Gamma}}^2$ are respectively the stationary mean value and variance of $\hat{\Gamma}(k)$, and $\phi_{\mathcal{N}}(\zeta; \mu, \sigma^2)$ is the probability density function (PDF) of a Gaussian random variable (GRV) $\mathcal{N}(\mu, \sigma^2)$. The packet delivery probability (PDP) associated to the same state within the respective region is given by

$$1 - \nu_M^{(r)} = \frac{1}{p_r} \int_{\zeta_r}^{\zeta_{r+1}} R_p(10^{\frac{\zeta}{10}}) \phi_{\mathcal{N}}(\zeta; \mu_{\hat{\Gamma}}, \sigma_{\hat{\Gamma}}^2) d\zeta. \quad (8)$$

Finally, the channel state transition probabilities are derived from integrating the joint PDF of the SINR $\hat{\Gamma}(k)$ over two consecutive packet transmissions and over the desired regions, r and q , as

$$p(q | r) = \frac{1}{p_r} \int_{\zeta_r}^{\zeta_{r+1}} \int_{\zeta_q}^{\zeta_{q+1}} \varphi_{\mathcal{N}}(s_{k-1}, s_k; \mu_{\hat{\Gamma}}, \sigma_{\hat{\Gamma}}^2, c_{\hat{\Gamma}}(\tau)) d\zeta_{k-1} d\zeta_k,$$

where $\varphi_{\mathcal{N}}(s_{k-1}, s_k; \mu_{\hat{\Gamma}}, \sigma_{\hat{\Gamma}}^2, c_{\hat{\Gamma}}(\tau))$ is the two-dimensional PDF of the Gaussian process $\hat{\Gamma}(k)$, as detailed in [12].

However, one issue related to the above technique is that in real (industrial) cases some of the channel parameters required for the above modeling are time-varying (due to their dependence on the state of the plant) and often only partially known. For these reasons, we propose a new approach to model the SINR, and thus the PEP. In this respect, a promising direction is the exploitation of historical data of the communication channel to identify a model via system identification and machine learning techniques.

3) *Auto-regressive model for channel simulation*: The application of data-driven methodologies requires the existence of a dataset containing trajectories of the SINR: in this paper we run Monte Carlo simulations of the wireless transmission model (5) and then apply our techniques to such trajectories. To this aim, in this section we illustrate the approach presented in [18], where the study of auto-regressive stochastic models for computer simulation of fading channels is addressed, to derive discrete-time trajectories of the process described in equation (5). In particular, let us consider a discrete-time Gaussian process $\{z(k)\}_{k \in \mathbb{N}}$ with auto-correlation function (ACF) $R_{zz}(n)$. We can derive an auto-regressive (AR) model of the following form that is able to generate trajectories of the process:

$$z(k) = - \sum_{n=1}^p a_n z(k-n) + w(k), \quad (9)$$

where $w(k)$ is a zero-mean white Gaussian noise process. The AR model parameters consist of coefficients $\{a_1, \dots, a_p\}$ and variance σ_p^2 of the driving noise $w(k)$. To estimate the coefficients a_j , $j = 1, \dots, p$, once the ACF R_{zz} is fixed from β_i and ξ_i , $i = 0, 1$, the relationship between R_{zz} and a_j is given as follows [19]:

$$R_{zz}(n) = \begin{cases} - \sum_{m=1}^p a_m R_{zz}(n-m), & n \geq 1 \\ - \sum_{m=1}^p a_m R_{zz}(-m) + \sigma_p^2, & k = 0 \end{cases} \quad (10)$$

Finally, coefficients a_j can be determined solving the set of Yule-Walker equations that can be easily derived from (10) (we refer the reader to [19] for further details).

The generated AR process has the following auto-correlation function:

$$\hat{R}_{zz}(n) = \begin{cases} R_{zz}(n), & 0 \leq n \leq p \\ - \sum_{m=1}^p a_m \hat{R}_{zz}[n-m], & n > p \end{cases} \quad (11)$$

The simulated process has the attractive property that its sampled ACF perfectly matches the desired sequence of ACF up to lag p . Therefore, since we know the ACF of both the residual power control error $\xi_i(k)$ and the shadowing correlation $\beta_i(k)$ we can exploit the AR model to generate sequences of $\xi_i(k)$ and $\beta_i(k)$ and then apply (5) to obtain the discrete-time trajectories of $\Gamma(k)$.

III. THE CLASSIFICATION AND REGRESSION TREE (CART) ALGORITHM

The aim of this section is to provide a short description of the classification and regression trees (CART) algorithm [13], in order to provide the basic notions to present our method in Sec. V that identifies a Markov chain wireless channel abstraction.

In a supervised framework, we consider a predictor dataset $\mathcal{P} = \{\lambda(k)\}_{k=1}^D$ and a response dataset $\mathcal{R} = \{\rho(k)\}_{k=1}^D$ of D samples each, where $\rho(k) \in \mathbb{R}$ is called response variable and $\lambda(k) \in \mathbb{R}^n$ is called predictor variable. The final goal of CART is to identify a function \mathcal{T} to estimate $\hat{\rho}(k) = \mathcal{T}(\lambda(k))$.

In the specific case of the CART algorithm [13], the dataset is partitioned into a set of hyper-rectangles R_1, \dots, R_ℓ , corresponding to the ℓ leaves of the tree. Then, $\hat{\rho}(k)$ is estimated in each leaf τ_i using a constant c_{τ_i} given by the average of the samples in the partition. Without any loss of generality we restrict our attention to recursive binary partition. Due to space limitations, we only briefly recall the partitioning algorithm of CART, and refer the reader to [13] for more details.

The CART algorithm creates the partition using a greedy algorithm to optimize the split variables and split points: starting with the whole dataset, consider a split variable j over the n available and a split point s , and define the

pair of half-planes as $R_L(j, s) = \{\lambda(k) \mid \lambda_j(k) < s\}$ and $R_R(j, s) = \{\lambda(k) \mid \lambda_j(k) \geq s\}$. Then, CART solves the following optimization problem to find the optimal j and s

$$\min_{j, s} \left[\min_{c_L} \sum_{\lambda(k) \in R_L(j, s)} (\rho(k) - c_L)^2 + \min_{c_R} \sum_{\lambda(k) \in R_R(j, s)} (\rho(k) - c_R)^2 \right], \quad (12)$$

and for any choice of j and s the inner minimization is solved by $c_L = \text{ave}(\rho(k) \mid \lambda(k) \in R_L(j, s))$ and $c_R = \text{ave}(\rho(k) \mid \lambda(k) \in R_R(j, s))$, where $\text{ave}(\cdot)$ is the arithmetic mean of the output samples. In other words, the optimal j^* and s^* minimize the sum of the quadratic prediction errors of the left and right partitions induced by the split variable and split point.

For each splitting variable, the determination of the split point s can be done very quickly and hence, by scanning through all of the inputs, the determination of the best pair (j, s) is feasible. Once the best split is found the dataset is partitioned into the two resulting regions, then the splitting procedure is repeated on each of the two regions. The process is repeated on all of the resulting regions until a stopping criterion is applied, e.g. tree size is a tuning parameter chosen to avoid overfitting and variance phenomena.

In the rest of this work we denote with \mathcal{T} the regression tree, with τ_i the i^{th} leaf of \mathcal{T} , with $|\mathcal{T}|$ the number of leaves of \mathcal{T} , with $|\tau_i|$ the number of samples in τ_i , with $c_{\tau_i} = \text{ave}(\rho(k) \mid \lambda(k) \in \tau_i)$ the prediction of leaf τ_i and, with a slight abuse of notation, with $\mathcal{T}(\lambda)$ the prediction of the regression tree, i.e.

$$\mathcal{T}(\lambda) = \sum_{\tau_i \in \mathcal{T}} \sum_{\lambda(k) \in \tau_i} \frac{\rho(k)}{|\tau_i|} I\{\lambda \in \tau_i\} = \sum_{\tau_i \in \mathcal{T}} c_{\tau_i} I\{\lambda \in \tau_i\}, \quad (13)$$

where $I\{\lambda \in \tau_i\}$ is the indicator function, which is equal to 1 if $\lambda \in \tau_i$ and 0 otherwise.

IV. SWITCHING ARX IDENTIFICATION

As discussed above, in this paper we leverage the techniques in [14] to construct a switching auto-regressive exogenous (SARX) model for the nonlinear plant f in (1) that can be directly used to setup a MPC problem, as will be done in Sec. VI. In particular, starting from a dataset $\mathcal{D} = \{(y(k), u(k))\}_{k=1}^D$ of D samples collected from the measurements of a physical system, respectively consisting of outputs $y(k) \in \mathbb{R}^{n_y}$ and inputs $u(k) \in \mathbb{R}^{n_u}$, we will derive for each $j = 0, \dots, N-1$ a model as follows:

$$x(k+j+1) = A_{\sigma_j(x(k))} x(k+j) + B_{\sigma_j(x(k))} u(k+j) + F_{\sigma_j(x(k))},$$

with $\sigma_j : \mathbb{R}^{n_x} \rightarrow \mathcal{M} \subset \mathbb{N}$ the switching signal, $x(k) \doteq [y^\top(k) \ \dots \ y^\top(k - \delta_y) \ u^\top(k-1) \ \dots \ u^\top(k - \delta_u)]^\top \in \mathbb{R}^{n_x}$ the state consisting of the regressive terms of the inputs and the outputs, $\delta_y, \delta_u \geq 0$, and $n_x = (\delta_y + 1)n_y + n_u \delta_u$.

V. MARKOV MODEL BASED ON REGRESSION TREES

In industrial environments, the derivation of Markov models based on the physics of wireless communication can be prohibitive as it requires complete knowledge of both the communication dynamics parameters and the disturbances/interferers. Furthermore, the presence of time-varying parameters can increase the complexity of the obtained model. We propose a novel methodology to derive a Markov model of the PDP based on the WNCs measurements. The approach exploits the transmissions' historical data to deal with the circumstances illustrated above. In particular, we handle the case wherein there is a dependency between the communication system's and the plant measurable outputs.

The main idea is to derive a Markovian model $\{\theta(k) \in \Theta\}_{k \in \mathbb{N}}$ abstracting the SINR stochastic process $\Gamma(k)$, with TPM $P = [p(j \mid i)]_{i,j}$ as in (3), and associate to each state $i \in \Theta$ a PDP. The learning procedure consists of three steps: (1) we grow a regression tree \mathcal{T} with predictor variables the current SINR and the plant measurements, i.e. $(\Gamma(k), x(k))$, and with response variable the SINR at the next time step, i.e. $\Gamma(k+1)$. Since the leaves of \mathcal{T} form a partition of the predictor space, we define the state-space of $\theta(k)$ associating to each element of the partition $\{R_{\tau_i}\}_{i=1}^{\ell_{\mathcal{T}}}$ a state of the Markov chain, i.e. $\Theta \doteq \{i \in \mathbb{N} : \tau_i \in \mathcal{T}\}$; (2) we grow a regression tree Π with predictor variables the current SINR, i.e. $\Gamma(k)$, and with response variable the prediction of the SINR at the next time step obtained using the regression tree \mathcal{T} , i.e. $\mathcal{T}(\Gamma(k), x(k))$. We will combine Π and \mathcal{T} to identify the TPM of $\{\theta(k)\}_{k \in \mathbb{N}}$; (3) we associate to each state of $\{\theta(k)\}_{k \in \mathbb{N}}$ a PDP.

1) *Step 1:* We grow a regression tree \mathcal{T} using as predictor dataset $\{\Gamma(k), x(k)\}_{k=1}^D = \mathcal{P}_{\mathcal{T}} \subset \mathbb{R}^{n_x+1}$, consisting of the current SINR and the plant measurable outputs, and as response dataset $\{\Gamma(k+1)\}_{k=1}^D = \mathcal{R}_{\mathcal{T}} \subset \mathbb{R}$, consisting of the SINR at the next time step. Let $\mathcal{T}((\Gamma(k), x(k)))$, with $\mathcal{T} : \mathcal{P}_{\mathcal{T}} \rightarrow \mathcal{R}_{\mathcal{T}}$, be the prediction of \mathcal{T} given the current measurements. As mentioned above, we associate to each element of the partition $\{R_{\tau_i}\}_{i=1}^{\ell_{\mathcal{T}}}$ a state of the Markov chain $\{\theta(k) \in \Theta\}_{k \in \mathbb{N}}$, i.e. $\Theta \doteq \{i \in \mathbb{N} : \tau_i \in \mathcal{T}\}$. Note that $\forall (\Gamma(k), x(k)) \in \mathbb{R}^{n_x+1}, \exists! \tau_i \in \mathcal{T} : (\Gamma(k), x(k)) \in R_{\tau_i}$, i.e. the current measurement $(\Gamma(k), x(k))$ is deterministically associated to one, and only one, discrete state of $\theta(k)$.

As each leaf τ_i contains a random subset of the SINR sequence, we assume independent and identically distributed samples belonging to the same partition element. We fit a GRV $G_{\tau_i} \sim \mathcal{N}(\mu_i, \sigma_i^2)$ to model the current level of SINR in each leaf $\tau_i \in \mathcal{T}$ of the tree:

$$\begin{aligned} \mu_i &= \frac{1}{|\tau_i|} \sum_{(\Gamma(k), x(k)) \in R_{\tau_i}} \Gamma(k), \\ \sigma_i^2 &= \frac{1}{|\tau_i| - 1} \sum_{(\Gamma(k), x(k)) \in R_{\tau_i}} (\Gamma(k) - \mu_i)^2. \end{aligned} \quad (14)$$

In Sec. V-3 we will exploit $\{G_{\tau_i}\}_{\tau_i \in \mathcal{T}}$ to estimate the PEP in each state of $\{\theta(k)\}_{k \in \mathbb{N}}$.

Following the same reasoning, we fit a GRV $G_{\tau_i}^+ \sim \mathcal{N}(\mu_{i+}, \sigma_{i+}^2)$ to model the next-step level of SINR for each leaf $\tau_i \in \mathcal{T}$ of the tree:

$$\begin{aligned} \mu_{i+} &= \frac{1}{|\tau_i|} \sum_{(\Gamma(k), x(k)) \in R_{\tau_i}} \Gamma(k+1), \\ \sigma_{i+}^2 &= \frac{1}{|\tau_i| - 1} \sum_{(\Gamma(k), x(k)) \in R_{\tau_i}} (\Gamma(k+1) - \mu_{i+})^2. \end{aligned} \quad (15)$$

In Sec. V-2 we will exploit $\{G_{\tau_i}^+\}_{\tau_i \in \mathcal{T}}$ to identify the TPM of $\{\theta(k)\}_{k \in \mathbb{N}}$.

We remark that the TPM of $\{\theta(k)\}_{k \in \mathbb{N}}$ could be identified using only \mathcal{T} , based on the number of samples that at time k stay in a leaf $\tau_i \in \mathcal{T}$, and at time $k+1$ jump to a leaf $\tau_j \in \mathcal{T}$, i.e. $p(j \mid i) = \tilde{n}(\tau_i, \tau_j) \cdot |\tau_i|^{-1}$ where

$$\begin{aligned} \tilde{n}(i, j) &\doteq \{(\Gamma(k_\epsilon), x(k_\epsilon)) \in \mathcal{P}_{\mathcal{T}} : \\ &(\Gamma(k_\epsilon), x(k_\epsilon)) \in R_{\tau_i}, (\Gamma(k_\epsilon + 1), x(k_\epsilon + 1)) \in R_{\tau_j}\}. \end{aligned} \quad (16)$$

The lack of this approach is that the tree \mathcal{T} partitions the dataset to minimize the prediction error of a deterministic estimate of $\Gamma(k+1)$, instead of minimizing the prediction error with respect to the estimate $\mathbb{E}[\mathcal{T}(\Gamma(k), x(k)) \mid \Gamma(k)]$.

We overcome this issue in the next section growing an additional regression tree Π , and combining \mathcal{T} and Π to identify the TPM to minimize the error with respect to $\mathbb{E}[\mathcal{T}(\Gamma(k), x(k)) \mid \Gamma(k)]$.

2) *Step 2*: The idea is to define a partition $\{R_{\pi_r}\}_{\pi_r \in \Pi}$ of \mathbb{R} minimizing the prediction error with respect to the estimate $\mathbb{E}[\mathcal{T}(\Gamma(k), x(k)) \mid \Gamma(k)]$, and exploit the Markov property to split the identification process of the TPM of $\{\theta(k)\}_{k \in \mathbb{N}}$ in two steps:

$$\begin{aligned} \mathbb{P}(\theta(k+1) = j \mid \theta(k) = i) &= p(j \mid i) \\ &\doteq \mathbb{P}((\Gamma(k+1), x(k+1)) \in R_{\tau_j} \mid (\Gamma(k), x(k)) \in R_{\tau_i}) \\ &= \sum_{\pi_r \in \Pi} \{\mathbb{P}((\Gamma(k+1), x(k+1)) \in R_{\tau_j} \mid \Gamma(k+1) \in R_{\pi_r}) \\ &\quad \mathbb{P}(\Gamma(k+1) \in R_{\pi_r} \mid (\Gamma(k), x(k)) \in R_{\tau_i})\}. \end{aligned} \quad (17)$$

To estimate $\mathbb{P}((\Gamma(k+1), x(k+1)) \in R_{\tau_j} \mid \Gamma(k+1) \in R_{\pi_r})$

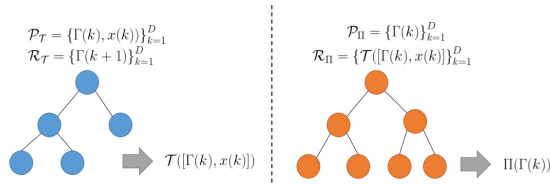


Fig. 2: TPM identification via regression trees

we define a new predictor dataset $\{\Gamma(k)\}_{k=1}^D = \mathcal{P}_{\Pi} \subset \mathbb{R}$ and a new response dataset $\{\mathcal{T}(\Gamma(k), x(k))\}_{k=1}^D = \mathcal{R}_{\Pi} \subset \mathbb{R}$. Then, we derive a new regression tree Π applying the CART algorithm to identify the function $\Pi : \mathcal{P}_{\Pi} \rightarrow \mathcal{R}_{\Pi}$. Let $\Pi(\Gamma(k))$ be the optimal estimation given by the tree Π . Define

$$\mathbb{P}((\Gamma(k+1), x(k+1)) \in R_{\tau_j} \mid \Gamma(k+1) \in R_{\pi_r}) \doteq p(\tau_j \mid \pi_r) = \frac{n(\pi_r, \tau_j)}{|\pi_r|} \quad (18)$$

where $n(\pi_r, \tau_j)$ is the number of samples that belong both to the leaf $\pi_r \in \Pi$ and the leaf $\tau_j \in \mathcal{T}$, i.e

$$n(\pi_r, \tau_j) \doteq |\{(\Gamma(k_\epsilon), x(k_\epsilon)) \in \mathcal{P}_{\mathcal{T}} : \Gamma(k_\epsilon) \in R_{\pi_r}, (\Gamma(k_\epsilon), x(k_\epsilon)) \in R_{\tau_j}\}|. \quad (19)$$

Notice that in contrast to the definition in equation (16), equation (19) does not involve the time evolution. For this reason, eq. (19) is useful to estimate the probabilities $\mathbb{P}((\Gamma(k+1), x(k+1)) \in R_{\tau_j} \mid \Gamma(k+1) \in R_{\pi_r})$.

Proposition 1: Let us assume that the dataset consists of independently drawn observations, and define the transition probabilities $p(\tau_j \mid \pi_r)$ as in Equation (18), then the CART algorithm creates the partition induced by $\{R_{\pi_r}\}_{\pi_r \in \Pi}$ optimally estimating the conditional expectation of the prediction in each leaf $\pi_r \in \Pi$, i.e. $c_{\pi_r} = \mathbb{E}[\mathcal{T}(\Gamma(k), x(k)) \mid \Gamma(k) \in R_{\pi_r}]$, $\forall \pi_r \in \Pi$.

Proof: see [20]. \blacksquare

To estimate $\mathbb{P}(\Gamma(k+1) \in R_{\pi_r} \mid (\Gamma(k), x(k)) \in R_{\tau_i}) = p(\pi_r \mid \tau_i)$ we exploit the set of GRVs defined in Equation (15) and the partition induced by the tree Π :

$$p(\pi_r \mid \tau_i) = \mathbb{P}(G_{\tau_i}^+ \in R_{\pi_r}) = \int_{R_{\pi_r}} \phi_{\mathcal{N}}(\zeta; \mu_{i^+}, \sigma_{i^+}^2) d\zeta \quad (20)$$

where $\phi_{\mathcal{N}}(\zeta; \mu, \sigma^2)$ is the PDF of the GRV $\mathcal{N}(\mu, \sigma^2)$. The TPM $P = [p(j \mid i)]_{i,j} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$ is defined by $p(j \mid i) = \mathbb{P}(\theta(k+1) = j \mid \theta(k) = i) \forall \tau_i, \tau_j \in \mathcal{T}$, as in Equation (17).

3) *Step 3*: The variable of interest in control applications is the PDP. The IEEE-802.15.4 provides the estimation of the PER given the SINR, see Equation (6). Let $\nu = (\nu_1, \dots, \nu_{|\mathcal{T}|})$ associate a PDP for each channel operating mode as follows:

$$1 - \nu_i = E[R_p(10^{G_{\tau_i}/10}) \mid \theta(k) = i] = \int_{\mathbb{R}} R_p(10^{\zeta/10}) \phi_{\mathcal{N}}(\zeta; \mu_{c_i}, \sigma_{c_i}^2) d\zeta. \quad (21)$$

Algorithms 1 and 2 summarize respectively the learning methodology and the procedure to estimate the packet delivery probability (PDP) $\hat{\nu}$ over a time horizon of length N .

Algorithm 1 Learning algorithm (off-line)

```

1: INPUT:  $\{\Gamma(k), x(k)\}_{k=1}^D$ , OUTPUT:  $\mathcal{T}, P, \nu$ 
2: function LEARN PEP
3:   DEFINE  $\mathcal{P}_{\mathcal{T}} = \{\Gamma(k), x(k)\}_{k=1}^D$ ,  $\mathcal{R}_{\mathcal{T}} = \{\Gamma(k+1)\}_{k=1}^D$ ;
4:   BUILD  $\mathcal{T}$  USING  $\mathcal{P}_{\mathcal{T}}$  TO PREDICT  $\mathcal{R}_{\mathcal{T}}$ ;
5:   DEFINE  $\mathcal{P}_{\Pi} = \{\Gamma(k)\}_{k=1}^D$ ,  $\mathcal{R}_{\Pi} = \{\Pi(\Gamma(k), x(k))\}_{k=1}^D$ ;
6:   BUILD  $\Pi$  USING  $\mathcal{P}_{\Pi}$  TO PREDICT  $\mathcal{R}_{\Pi}$ ;
7:   BUILD  $P$  BASED ON (17);
8:   COMPUTE  $\nu$  USING (21);
9:   return  $\mathcal{T}, P, \nu$ 
10: end function

```

Algorithm 2 PDP estimation (run-time)

```

1: INPUT:  $\mathcal{T}, P, \nu, (\Gamma(k), x(k)), N$ , OUTPUT:  $\hat{\nu}$ 
2: function PREDICT PDP
3:   Let  $i \in \mathbb{N}$  be the integer such that  $(\Gamma(k), x(k)) \in \tau_i$ ;
4:   for all  $j = 1, \dots, N$  do
5:      $\hat{\nu}^j = \nu P^j(i, :)$ , where  $P^j(i, :)$  is the  $i$ -th row of  $P^j$ .
6:   end for
7:   return  $\hat{\nu}$ 
8: end function

```

VI. CASE STUDY

We consider a WNCs consisting of an inverted pendulum on a cart remotely controlled over a WirelessHART link as illustrated in Sec. II. In the numerical simulations, we model the plant using the nonlinear discrete-time model of the inverted pendulum and we model the packet delivery process as in equation (6) through the SINR trajectories obtained from the AR model in Sec. II-3. We compare two implementations of Stochastic Model Predictive Control (S-MPC) [21]: one based on the physics based channel model of Sec. II-2 as in [12], and one based on the data-driven channel model of Sec. V.

1) *Plant model*: We consider the nonlinear discrete-time model of the inverted pendulum on cart described in [22], where $y(k+1) = y(k) + f(x(k), u_a(k))T_u$. $y \in \mathbb{R}^4$, $u \in \mathbb{R}$, where y_1 is the cart position, y_2 is the velocity, y_3 is the pendulum angle, y_4 is the angular velocity u is a control force applied to the cart and T_u is the sampling time.

We analyze a case wherein the model's dynamics influence the channel behavior. More in detail, we consider that a time-varying distance between the plant and the controller, d_0 in equation (4), depends on the cart's position, y_1 , i.e. $d_0(k) = y_1(k) + \bar{d}_0$, $\bar{d}_0 \in \mathbb{R}^+$. The other channel parameters are assumed constants. As consequence, the intermittent control packets can be modeled as follows:

$$u_a(k) = \nu(k)u(k), \quad \nu(k) \sim B(1, 1 - R_b(\gamma(k))) \quad (22)$$

where $u(k)$ is the input computed by the controller, $B(n, p)$ is the Bernoullian distribution with a time-varying parameter. Notice that $y_1(k)$ influences $\gamma(k)$ via equation (4).

2) *Numerical results*: The data-driven model derived in Sec. IV can be used to formalize the following:

Problem 1: Stochastic Model Predictive Control

$$\begin{aligned} \min_{u_k} \quad & \mathbb{E} \left[e_{k+N}^\top Q e_{k+N} + \sum_{j=0}^{N-1} e_{k+j+1}^\top Q e_{k+j+1} + u_{k+j}^\top R u_{k+j} \right] \\ \text{s.t.} \quad & x_{k+j+1} = A'_{i_j} x_k + \sum_{\alpha=0}^j B'_{i_j, \alpha} \hat{v}_{k+\alpha} u_{k+\alpha} + F'_{i_j}, \\ & u_{k+j} \in \mathcal{U}, \mathbb{E}[x_{k+j+1}] \in \mathcal{O}, \\ & \mathbb{E}[x_{k+N}] \in \mathcal{O}_N, x_k = x(k), j = 0 \dots, N-1, \end{aligned}$$

where $e_k = x_k - x_k^*$ is the difference between the current state of the plant and the target, $\hat{v}_{k+\alpha}$ is the PDP prediction given the current measurements, and \mathcal{O}, \mathcal{U} are polyhedra that specify the variables constraints. At each time step the optimal inputs $u_k^*, \dots, u_{k+N-1}^*$ are computed using QP, and only the first one is applied to the system, i.e. $u(k) = u_k^*$.

The plant dynamics used in the MPC solution are identified using the methodology summarized in Sec. IV and using a dataset consisting of 100 simulations, each one with time duration of 6 seconds, of the input and output of the plant with no packet losses. At any time k we can use such model and the measurement of $x_k = x(k)$ to determine the switching sequence i_0, \dots, i_{N-1} and hence the matrices $A'_{i_{j-1}, i_j}, B'_{i_{j-1}, i_j}, F'_{i_{j-1}, i_j}$. The solution of Problem 1 also requires knowledge of the initial state of \hat{v}_k and of its TPM P [21]: we derive two models of \hat{v}_k , respectively using the methodology in [12] and in of Sec. V: for both the data-driven and physics-based channel Markov models, we set the number of channel operating modes equal to 9.

The cost function in Problem 1 models a tradeoff between penalizing deviations from the desired trajectory x_k^* and minimizing the control effort. We define as control performance metric the cumulative cost of Problem 1 to compare the performance using the physics based channel model of Sec. II-2 as in [12] and the data-driven channel model of Sec. V.

Tables I and II in reference [20] show the plant and channel parameters, respectively. The minimum update period $T_u = 0.1s$ of the WirelessHART standard is too slow for several control applications, and makes the wireless link uncorrelated at the packet level. Thus, in view of showing the impact of our algorithms as a methodological enabler for the development of mobile network technologies that support much higher update rates, we consider $T_u = 0.001s$. Moreover, to emphasise the impact and improvements of stochastic vs deterministic MPC, we set $\bar{d}_0 = 14$ to consider a scenario based on significant packet loss rate, where it is evident that deterministic MPC cannot guarantee acceptable performance while stochastic MPC does.

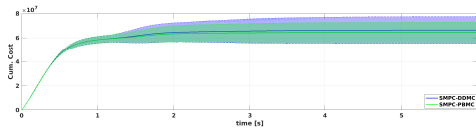


Fig. 3: Cumulative cost of the closed-loop simulation.

To statistically validate the control performance we ran Monte Carlo simulations generating 500 admissible trajectories, each with 6000 samples (corresponding to 6s). For all the simulations, the initial state is $x(0) = [2, 0, \pi, 0]^\top$ and the target state is $x^* = [5, 0, \pi, 0]^\top$. In Fig. 3 the averaged cumulative cost is displayed in solid line, and the 95.4% confidence interval is represented with a shaded area. The performance of the controllers based on physics-based and data-driven models is very close, with the advantage of the data-driven approach that no a-priori knowledge of channel model and parameters is required.

VII. CONCLUSIONS

This paper provides a novel technique to learn Markov models representing fading wireless channels. We consider a validation scenario consisting of a WNCs that exploits a WirelessHART radio link to send the optimal control inputs generated by a Stochastic MPC, and show that the control performances of our data-driven approach and of a physics-based approach based on a stationary finite-state Markov chain are extremely close: this implies that in practical applications, when assuming perfect knowledge of the channel model and parameters is not possible, the methodology presented in this paper is a valid and very effective alternative. In future work we plan to validate our techniques in an experimental setup and to consider more general communication scenarios.

REFERENCES

- [1] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 978–1013, 2018.
- [2] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1013–1024, 2016.
- [3] A. Ahlen, J. Akerberg, M. Eriksson, A. J. Isaksson, T. Iwaki, K. H. Johansson, S. Knorn, T. Lindh, and H. Sandberg, "Toward wireless control in industrial process automation: A case study at a paper mill," *IEEE Control Systems Magazine*, vol. 39, no. 5, pp. 36–57, 2019.
- [4] A. M. Deji Chen, Mark Nixon, *WirelessHART*. Springer, 2010.
- [5] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.
- [6] Y. Zaccchia Lun and A. D'Innocenzo, "Stabilizability of Markov jump linear systems modeling wireless networked control scenarios," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5766–5772.
- [7] W. Turin and R. van Nobelen, "Hidden markov modeling of flat fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, pp. 1809–1817, 1998.
- [8] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models (no. tr-97-021)," *Berkeley, CA: International Computer Science Institute and Univ. Calif. Berkeley*, 1998.
- [9] R. Lawrence and A. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [10] K. Salamati and S. Vaton, "Hidden markov modeling for network communication channels," *SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, p. 92–101, Jun. 2001. [Online]. Available: <https://doi.org/10.1145/384268.378439>
- [11] P. Sadeghi, R. Kennedy, P. Rapajic, and R. Shams, "Finite-state markov modeling of fading channels," *IEEE Signal Processing Magazine*, vol. 57, 2008.
- [12] Y. Zaccchia Lun, C. Rinaldi, A. Alrishi, A. D'Innocenzo, and F. Santucci, "On the impact of accurate radio link modeling on the performance of wirelessHART control networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2430–2439.
- [13] L. Breiman, *Classification and regression trees*. Routledge, 2017.
- [14] F. Smarra, G. D. Di Girolamo, V. De Iuliis, A. Jain, R. Mangharam, and A. D'Innocenzo, "Data-driven switching modeling for mpc using regression trees and random forests," *Nonlinear Analysis: Hybrid Systems*, vol. 36, p. 100882, 2020.
- [15] IEEE 802.15.4-2006, "IEEE Standard for Information technology – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," IEEE, Standard, September 2006.
- [16] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [17] C. Fischione, F. Graziosi, and F. Santucci, "Approximation for a sum of On-Off log-normal processes with wireless applications," *IEEE Trans. Commun.*, vol. 55, no. 9, pp. 1822–1822, Sept. 2007.
- [18] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, Jul. 2005.
- [19] S. M. Kay, *Modern spectral estimation*. Pearson Education India, 1988.
- [20] L. F. F. Reyes, F. Smarra, Y. Zaccchia Lun, and A. D'Innocenzo, "Learning Markov models of fading channels in wireless control networks: a regression trees based approach (extended version)," 2021. [Online]. Available: <http://arxiv.org/abs/2104.13280>
- [21] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2011.
- [22] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.