

# Predictive Closed-Loop Remote Control Over Wireless Two-Way Split Koopman Autoencoder

Abanoub M. Girgis<sup>1</sup>, Member, IEEE, Hywoon Seo<sup>2</sup>, Member, IEEE, Jihong Park<sup>3</sup>, Senior Member, IEEE, Mehdi Bennis<sup>1</sup>, Fellow, IEEE, and Jinho Choi<sup>3</sup>, Senior Member, IEEE

(Invited Paper)

**Abstract**—Real-time remote control over wireless is an important yet challenging application in fifth-generation and beyond due to its mission-critical nature under limited communication resources. Current solutions hinge on not only utilizing ultrareliable and low-latency communication (URLLC) links but also predicting future states, which may consume enormous communication resources and struggle with a short prediction time horizon. To fill this void, in this article we propose a novel two-way Koopman autoencoder (AE) approach wherein: 1) a sensing Koopman AE learns to understand the temporal state dynamics and predicts missing packets from a sensor to its remote controller and 2) a controlling Koopman AE learns to understand the temporal action dynamics and predicts missing packets from the controller to an actuator co-located with the sensor. Specifically, each Koopman AE aims to learn the Koopman operator in the hidden layers while the encoder of the AE aims to project the nonlinear dynamics onto a lifted subspace, which is reverted into the original nonlinear dynamics by the decoder of the AE. The Koopman operator describes the linearized temporal dynamics, enabling long-term future prediction and coping with missing packets and closed-form optimal control in the lifted subspace. Simulation results corroborate that the proposed approach achieves a 38× lower mean squared control error at 0-dBm signal-to-noise ratio (SNR) than the nonpredictive baseline.

**Index Terms**—6G, autoencoder (AE), beyond fifth-generation (5G), Koopman theory, remote control, split learning.

## I. INTRODUCTION

ONE OF the visions of beyond fifth-generation (5G) and 6G communication systems [2], [3], [4] is to leverage sensing, communication, and connectivity in a closed-loop

Manuscript received 13 January 2022; revised 7 April 2022 and 4 July 2022; accepted 30 August 2022. Date of publication 14 September 2022; date of current version 21 November 2022. This work was supported in part by the Academy of Finland 6G Flagship under Grant 318927 and Project SMARTER; in part by Project EU-ICT IntelloT and EU-CHISTERA LeadingEdge; in part by Project CONNECT, Infotech-NOOR, and NEGEIN; and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2022R1F1A1075078. This article was presented in part at the 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2021, pp. 1191–1196 [DOI: 10.1109/PIMRC50174.2021.9569357]. (Corresponding author: Hywoon Seo.)

Abanoub M. Girgis and Mehdi Bennis are with the Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland (e-mail: abanoub.pipaoy@oulu.fi; mehdi.bennis@oulu.fi).

Hywoon Seo is with the Department of Electronics and Communications Engineering, Kwangju University, Seoul 01897, South Korea (e-mail: hywoonseo@kw.ac.kr).

Jihong Park and Jinho Choi are with the School of Information Technology, Deakin University, Geelong, VIC 3220, Australia (e-mail: jihong.park@deakin.edu.au; jinho.choi@deakin.edu.au).

Digital Object Identifier 10.1109/JIOT.2022.3206415

integrated manner. The basic principle of closed-loop remote control has been studied in the literature of wireless networked control systems (WNCSs) [5]. Nonetheless, the feasibility of these methods becomes questionable in the presence of mission-critical remote control applications, including remote surgery [6], industrial Internet of Things (IIoT) control in a smart factory [7], autonomous vehicular platooning [8], low-earth-orbit (LEO) satellite maneuvering to avoid space debris [9], and so forth. These remote control applications not only impose extremely stringent communication latency requirements but also are unable to afford missing any single control command.

To address these concerns, one could exploit wireless communication that guarantee extremely high reliability with low latency. Indeed, supporting mission-critical applications over wireless links have been investigated in the context of ultrareliable and low-latency communication (URLLC) [10], [11]. However, URLLC is originally intended to support short packets [12], in which the guaranteed reliability and latency are challenged by the long packets transmission requirements in the aforementioned mission-critical applications. Furthermore, even one-way URLLC requires an enormous amount of bandwidth to reserve dedicated channels while consuming a significant amount of energy to increase transmit power [13]. This challenge is aggravated when supporting closed-loop control applications where errors may propagate during the two-way closed-loop interaction.

On the other hand, by leveraging recent advances in data-driven machine learning (ML), one could learn to predict future states (e.g., position, velocity, and temperature) based on previous data samples. This enables to carry out the proactive decision-making for reduced control latency or equivalently for improved reliability by communicating the same messages multiple times [14]. Time-series prediction falls in this category, in which recurrent neural networks (RNNs), such as long short-term memory (LSTM) [15] and gated recurrent unit (GRU) [16] are widely used for this purpose. While effective, their prediction accuracy often decreases sharply with the forecast time horizon, limiting their applicability only to short-term prediction such as forecasting the next video frame within a few milliseconds [17]. Such a time horizon may be too short to evaluate an optimal control action, particularly under nonlinear and complex state dynamics.

To fill this void, in this article, we propose a novel ML-based closed-loop remote control framework inspired by

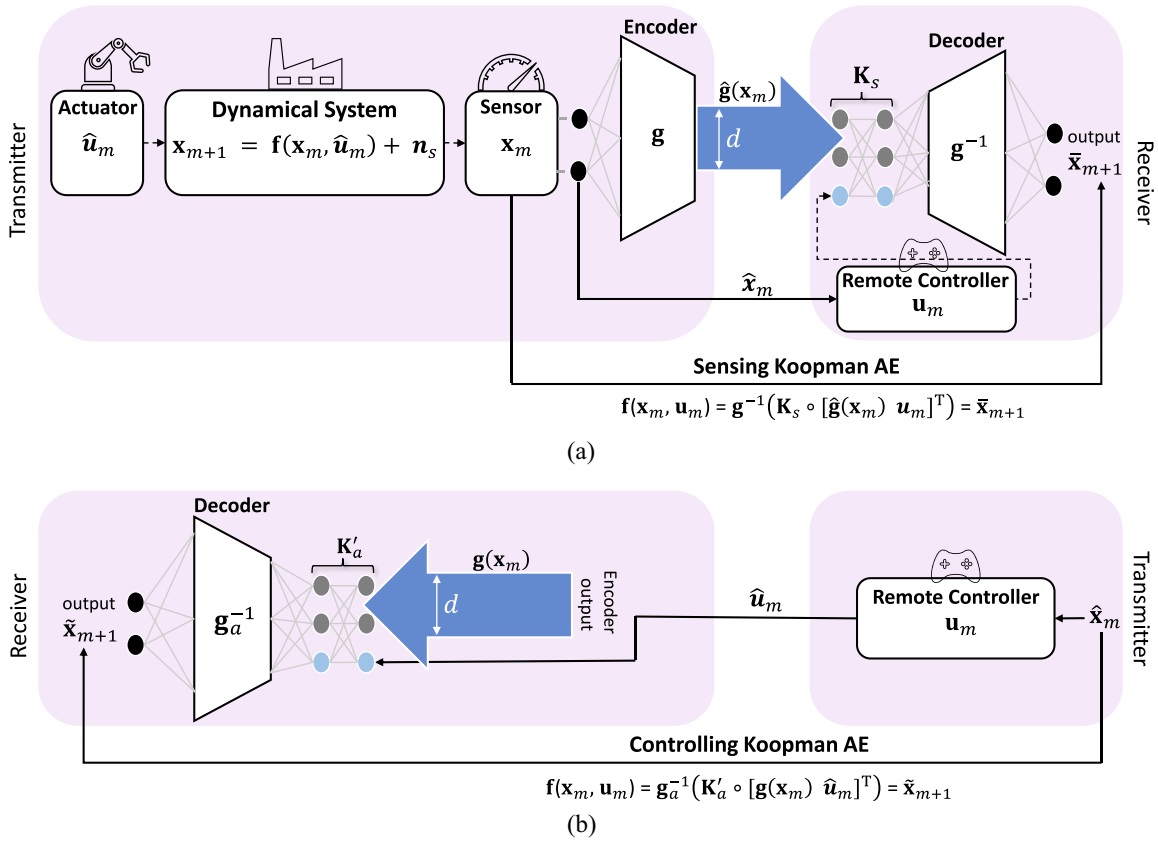


Fig. 1. Schematic representation of WNCS via two-way split Koopman AE architecture. (a) Wireless split sensing Koopman AE. (b) Wireless split controlling Koopman AE.

the Koopman theory [18], [19] and split learning [20], [21] using an autoencoder (AE) architecture, i.e., Koopman AE. In particular, we consider an AE split into its encoder and decoder parts, whereby the smashed data or latent representations propagate from the encoder to the decoder while the prediction errors or gradients propagate back in the reverse direction [20], [21]. The encoder part stored at the transmitter learns to map the input data into its lower-dimensional latent representation which is then reconstructed by the decoder stored at the receiver. This mapping reduces the communication payload size in the case of discovering the Koopman invariant subspace that ensures long-term prediction accuracy and control system stability.<sup>1</sup>

Subsequently, rather than predicting individual future states, we learn the temporal state dynamics as a whole. However, discovering the dynamics in the raw state space is difficult notably when the dynamics are nonlinear. Instead, inspired by the Koopman operator theory [18], [19], we aim to identify the dynamics in a lifted subspace in which the latent dynamics are linearized such that multiplying the Koopman operator  $M$  times yields the state prediction in  $M$  time slots. This provides an additional benefit of applying a closed-form optimal control derived from the linearized subspace, thereby reducing the control overhead in terms of computation.

<sup>1</sup>Ideally, according to Koopman theory, the lower-dimensional mapping in the Koopman operator amounts to lossless compression. However, the Koopman AE-based approach might lose system reliability depending on how low dimensional space the input data are mapped into [22].

Additionally, the control errors are reduced compared to the linear approximation method around an equilibrium point that becomes vacuous when the initial condition goes far from the equilibrium point. Recent works have demonstrated that the matrix form of the Koopman operator can be found in the hidden layers of an AE [23], which coincides with the aforementioned split AE architecture.

Last, in the closed-loop remote control, we avoid error propagation by additionally learning the temporal action dynamics as illustrated in Fig. 1. Consequently, in the forward link, any missing state reception due to poor channel conditions or excessive latency can be predicted by the remote controller that understands the state dynamics. Likewise, in the reverse link, any missing control action command receptions can be replaced by the actions predicted by the actuator that understands the action dynamics.

#### A. Backgrounds and Related Works

In the literature of remote control over wireless links, one central question is how to cope with communication imperfections on the control system operations. These works are often termed communication and control co-design (CoCoCo) that deal with the tradeoff between wireless resource consumption and control stability [24], [25]. In this direction, dynamic sensor scheduling approaches were proposed to cope with time-varying control and channel states, thereby improving control stability and communication efficiency [26], [27]. In [28] and [29], a channel state information (CSI) aware

scheduling and the power allocation method were introduced to minimize the total power consumption while guaranteeing a target control performance under limited communication resources. In order to meet stringent latency and reliability requirements in the context of URLLC and time-sensitive control systems, control dynamics and CSI aware resource allocation and scheduling solutions were developed in [30] and [31], which can partly relax URLLC requirements without compromising the control performance. In our prior work [25], a predictive and control-aware scheduler was proposed, in which the future states and actions are locally predicted using a Gaussian process regression approach (GPR) at the controller and actuator, respectively. Nevertheless, these works only consider linear control systems, questioning their feasibility when the system dynamics are nonlinear.

Nonlinear control system dynamics have been extensively studied in the field of control theory. A periodic event-triggered control is proposed in [32], [33], and [34] to stabilize nonlinear control systems under communication constraints. According to this scheme, the event-triggered condition is only updated at some sampling instances to preserve control stability and reduce the number of transmissions. However, the previously mentioned scheme in [32], [33], and [34] suffers from high computational complexity due to the system's nonlinearity. Furthermore, it suffers from long-term savings of the wireless communication resources as it fails to identify the nonlinear system dynamics at the controller. In [35] and [36], a model predictive control approach is proposed for nonlinear control systems to compensate for two-channel packet loss. However, [35] and [36] assumes that the accurate nonlinear system dynamics are given at the remote controller to compute the control action based on a nonlinear optimization problem. Hence, this approach suffers from impracticality due to the previously specified nonlinear dynamics and inaccuracy due to the uncertainty inherited in practical systems. In addition, it has a high computational burden due to the nonconvex optimization problem. Finally, this approach requires transmitting a finite number of predicted control actions in one packet to the actuator, affecting the transmission delay.

To make the nonlinearity amenable to stability analysis, the Jacobian linearization is a well-known method that provides a linear approximation around the equilibrium point using the Taylor series expansion [37], [38]. However, such an approximation becomes vacuous when the states of interest are far from the equilibrium point [37], [38]. As an alternative, Koopman operator methods have been suggested in [18] and [39], which represent finite-dimensional and nonlinear system states using infinite-dimensional and linear forms via linearly evolving functions of states, termed observables. Such Koopman representations can be obtained using the dynamic mode decomposition (DMD) algorithm as shown in open-loop [40] and closed-loop [41] scenarios. The main challenge in this direction is to find the minimum number of observables that sufficiently describe given nonlinear system dynamics [40], [41]. In [23], a deep learning-based method was proposed for providing finite-dimensional Koopman representations using an AE, i.e., Koopman AE. Leveraging and extending this method that originally ignores wireless

connectivity, in this work, we propose a two-way Koopman AE framework using a pair of two AE architectures to cope with wireless communication outages that may propagate control errors in closed-loop systems, hindering control stability.

## B. Contributions and Organization

We propose a split learning of Koopman operator for real-time remote control systems. The major contributions of this article can be summarized as follows.

- 1) For predictive closed-loop remote control, we propose a two-way split Koopman AE architecture comprising: a) *a sensing Koopman AE* from a sensor to its remote controller in the forward link and b) *a controlling Koopman AE* from the controller to an actuator co-located with the sensor in the reverse link (see Fig. 1).
- 2) To train the proposed architecture, for the sensing Koopman AE, we first derive the temporal state evolution with the Koopman operator (see (16) in Section III-B), and provide the training loss function (see (26) in Section III-C). For the controlling Koopman AE, we derive the temporal action evolution with the Koopman operator (see (27) in Section III-E), and provide the training loss function (see (29) in Section III-E).
- 3) By simulation, in an inverted cart-pole system, we corroborate that the proposed two-way split Koopman AE is robust to consecutive packet losses. At 0-dBm signal-to-noise ratio (SNR) in both forward and reverse links, we observe that the mean squared control error (MSCE) of the proposed architecture is  $38\times$  lower than that of the remote control without prediction (see Fig. 8 in Section IV).
- 4) Given the linearized temporal state evolution via the Koopman operator, we derive a closed-form optimal control action (see (19) in Section III-C) and observe that the proposed approach has better control performance than the standard Jacobian linearization technique for nonlinear control (see Fig. 9 in Section IV).
- 5) Finally, with an extensive set of simulation results, we show the impact of SNR, latent representation dimension, number of control trajectories, and training data set size on the state/action prediction error (see Tables I and II), Koopman AE convergence (Fig. 4), and training completion time (Figs. 5–7 in Section IV).

Note that our preliminary work has shown the effectiveness of the one-way sensing Koopman AE in a remote monitoring scenario [1]. Extending this to the closed-loop remote control scenario under study is a daunting task mainly due to the error propagation within the loop. We address this challenge by developing a two-way Koopman AE framework that additionally introduces another Koopman AE for control action prediction. Furthermore, compared to the preliminary version, in this article, we provide a more extensive set of simulation results, advocating the feasibility of the Koopman AE framework under a wide range of future prediction

intervals, different control system parameters, and different communication costs in terms of SNR and payload sizes.

The remainder of this article is organized as follows. In Section II, we specify the WNCSS architecture, including the models for control and communication systems. In Section III, we present the proposed two-way split Koopman AE architecture for predictive and linear control. In Sections IV and V, we present the simulation results and conclude this article.

## II. SYSTEM MODEL

Consider a closed-loop WNCSS which comprises a *plant* that exhibits nonlinear dynamics, a *sensor* that samples the plant's state, a *controller* that computes optimal control actions based on the sampled states, and an *actuator* that applies the control action command on the plant received from the controller. Specifically, as illustrated in Fig. 1, we assume that the sensor and actuator are co-located and share the same transceiver. On the other hand, the controller is located away from them with a computational capability to produce control action commands to ensure system stability. On this account, the controller operates remotely over wireless channels, by receiving the sensors' sampled states via the forward link and feeding back the computed optimal control action commands to the actuators via the reverse link.

### A. Control System Architecture

Consider that the control system operates at a fixed *control period*  $\tau_o > 0$  seconds to keep the plant stable over time. Hence, the sensor samples the state of the plant at a fixed sampling rate  $1/\tau_o$  and transmits the samples to the controller over wireless links. The  $m$ th sampled state of the plant at time  $t = m\tau_o$  is denoted by a  $p$ -dimensional vector  $\mathbf{x}_m = [x_{m,1} \ x_{m,2} \ \dots \ x_{m,p}] \in \mathbb{R}^p$ . On the other hand, an optimal control action command computed at the controller based on the  $m$ th sample  $\mathbf{x}_m$  is denoted by a  $q$ -dimensional vector  $\mathbf{u}_m = [u_{m,1} \ u_{m,2} \ \dots \ u_{m,q}] \in \mathbb{R}^q$ . Moreover, the state evolution of the plant's nonlinear dynamics at time  $t = (m+1)\tau_o$  is given as

$$\mathbf{x}_{m+1} = \mathbf{f}(\mathbf{x}_m, \mathbf{u}_m) + \mathbf{n}_s \quad (1)$$

$$= \mathbf{f}_s(\mathbf{x}_m) + \sum_{i=1}^q \mathbf{f}_{u,i}(\mathbf{x}_m)u_{m,i} + \mathbf{n}_s \quad (2)$$

where  $\mathbf{n}_s \in \mathbb{R}^p$  is a  $p$ -dimensional random system noise vector at  $t = (m+1)\tau_o$ . The entries of the noise vector are assumed to be independently and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance  $N_s$  for all  $m \in \mathbb{Z}_+$ . The function  $\mathbf{f} : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^p$  is a nonlinear state transition function of the plant's current state and control action command that steers the plant's state forward in time. Specifically, we introduce a model that disambiguates the state dynamics from the effect of the control actuation by reformulating the nonlinear dynamics into a control-affine nonlinear system [42], [43] as (2), where  $\mathbf{f}_s(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is the unforced system dynamics and  $\mathbf{f}_{u,i}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^p$  is a state dependant control coupling term for  $u_{m,i}$  for all  $i \in \{1, \dots, q\}$ .

The time duration between  $t = m\tau_o$  and the time point at which an actuator receives the estimated control action

command  $\hat{\mathbf{u}}_m$  corresponding to the plant's state  $\mathbf{x}_m$  is referred to as *control loop duration* and denoted by

$$\tau_m = \tau_{m,\text{comm}} + \tau_{m,\text{comp}} \quad (3)$$

for all  $m \in \mathbb{Z}_+$ . This control loop duration incorporates the communication delay  $\tau_{m,\text{comm}}$  through forward and reverse links and computing delay  $\tau_{m,\text{comp}}$  for obtaining the optimal control action command. On the one hand, in the Jacobian baseline, this calls for the forward link to deliver the sampled states to the controller, while the reverse link is also needed for the controller to deliver the obtained control action command to the actuator. On the other hand, in the proposed approach explained in the following section, the system state can be predicted using the trained sensing Koopman AE, thereby, only requiring the reverse link as long as the state prediction is accurate.

### B. Wireless Communication Model

Suppose the forward and reverse links follow a time division multiple access (TDMA) approach and assume that channel reciprocity holds between forward and reverse channels under the assumption that the sensor and actuator are co-located. Moreover, we consider a Rayleigh block fading channel model, where the coherence time is strictly longer than the control loop time. The path-loss of both forward and reverse links is given as

$$\text{PL}_{\text{dB}}(D) = \text{PL}_{\text{dB}}(D_0) + 10\eta \log_{10}\left(\frac{D}{D_0}\right) \quad (4)$$

where  $D$  is the distance from sensor and actuator to the controller,  $\text{PL}_{\text{dB}}(D_0)$  denotes the path-loss at the reference distance  $D_0$ , and  $\eta \geq 2$  is the path-loss exponent. Hence, the received SNR at the remote controller is given as

$$\text{SNR}_m = 10^{-\frac{\text{PL}_{\text{dB}}(D_0)}{10}} \frac{P_t |h_m|^2}{N_c} \left(\frac{D_0}{D}\right)^\eta \quad (5)$$

where  $h_m$  is the Rayleigh fading channel gain from the sensor to the controller (or the controller to the actuator) at the  $m$ th control loop duration, where  $|h_m|^2$  is exponentially distributed with parameter  $\lambda = 1$  for all  $m \in \mathbb{Z}_+$ ,  $P_t$  is the transmission power used at both communication ends, and  $N_c$  is the channel noise power. In addition, we suppose that the channel gains are i.i.d. over time. Having (5), the transmission rate of both forward and reverse link communications in the control system during the  $m$ th control loop can be expressed as

$$R_m = W \log_2(1 + \text{SNR}_m) \quad (6)$$

where  $W$  is the communication bandwidth.

To stabilize the control system, the control loop duration should be at least shorter than the control period, that is  $\tau_m < \tau_o$  for all  $m \in \mathbb{Z}_+$ . Denote by  $L_m$  the overall communicating information bit length in the  $m$ th control loop duration, such that  $(L_m/R_m)$  indicates the communication time. Then, the outage probability of the  $m$ th control loop can be expressed as

$$\epsilon_m = \mathbb{P}[\tau_{m,\text{comm}} > \tau_o - \tau_{m,\text{comp}}] \quad (7)$$

$$= 1 - \exp\left[-10^{\frac{\text{PL}_{\text{dB}}(D_0)}{10}} \frac{N_c D_0^\eta}{P_t} \left(2^{\frac{L_m}{(\tau_o - \tau_{m,\text{comp}})W}} - 1\right)\right] \quad (8)$$

where (8) holds from the cumulative distribution function (c.d.f.) of the exponential random variable. Note that the control outage probability for each control loop is affected by the communication payload size  $L_m$  and computation time  $\tau_{m,\text{comp}}$  while the other communication parameters in (8) are given. Therefore, to reduce the outage probability, the control system should be designed to operate at small communication and computation costs.

### III. TWO-WAY SPLIT KOOPMAN AUTOENCODER ARCHITECTURE FOR PREDICTIVE LINEAR CONTROL

Toward reducing communication and computation costs for closed-loop remote control, we propose a two-way Koopman AE architecture comprised of *sensing and controlling Koopman AEs*. To this end, this section first revisits fundamentals of the Koopman operator theory. Then, a split learning method for obtaining a sensing Koopman AE that linearizes the nonlinear state dynamics enabling future state prediction at the controller is proposed. Moreover, armed with a well-trained sensing Koopman AE, a linear quadratic regulator (LQR) is applied over the linearized Koopman subspace to obtain the optimal control action command at a small computational cost. In addition, we propose a split learning-based controlling Koopman AE that predicts future control action commands at the actuator.

#### A. Preliminary: Koopman Operator for Closed-Loop Control

Typically, a data sample observed from a closed-loop controlled dynamical system can be seen as a function of a system state and control action. Define functions  $g : \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}$ , dubbed *observables* which span an infinite-dimensional Hilbert space  $\mathcal{H}$ . The Koopman operator  $\mathcal{K} : \mathcal{H} \rightarrow \mathcal{H}$  is a linear operator that acts on  $\mathcal{H}$  [44], [45] such that

$$\mathcal{K}g(\mathbf{x}_m, \mathbf{u}_m) = g(\mathbf{f}(\mathbf{x}_m, \mathbf{u}_m), \mathbf{u}_{m+1}) \quad (9)$$

for some observable  $g$ . Note that by ignoring the additive system noise in (1), the expression (9) can be rewritten as

$$\mathcal{K}g(\mathbf{x}_m, \mathbf{u}_m) = g(\mathbf{x}_{m+1}, \mathbf{u}_{m+1}) \quad (10)$$

and in other words, the Koopman operator in (10) enables to express the linear evolution of the nonlinear system dynamics in the function space  $\mathcal{G}$ . However, to obtain the Koopman operator, representation and computation issues may arise, since the operator acts on the infinite-dimensional space  $\mathcal{H}$ .

One approach to resolve such problems is to find an invariant subspace spanned by a finite set of functions. That is, finding a span of  $d$  different functions  $\{g_1, g_2, \dots, g_d\}$ , such that a function  $g$  in this subspace

$$g(\mathbf{x}_m, \mathbf{u}_m) = a_1 g_1(\mathbf{x}_m, \mathbf{u}_m) + \dots + a_d g_d(\mathbf{x}_m, \mathbf{u}_m) \quad (11)$$

for some real-valued coefficients  $a_1, \dots, a_d \in \mathbb{R}$ , is still in the same subspace after acted on by the Koopman operation

$$\mathcal{K}g(\mathbf{x}_m, \mathbf{u}_m) = b_1 g_1(\mathbf{x}_m, \mathbf{u}_m) + \dots + b_d g_d(\mathbf{x}_m, \mathbf{u}_m) \quad (12)$$

for some real-valued coefficients  $b_1, \dots, b_d \in \mathbb{R}$ . Note that the Koopman operator is linear and thus allows eigendecomposition, and any finite set of eigenfunctions of the Koopman

operator will span a Koopman invariant subspace [46]. Therefore, by introducing a finite-dimensional Koopman matrix representation  $\mathbf{K} \in \mathbb{R}^{d \times d}$  for a given Koopman invariant subspace, we can obtain a global linearization expression of a nonlinear system dynamics in (1) as

$$\mathbf{g}(\mathbf{x}_{m+1}, \mathbf{u}_{m+1}) = \mathbf{K}\mathbf{g}(\mathbf{x}_m, \mathbf{u}_m) \quad (13)$$

where  $\mathbf{g}$  is the  $d$ -dimensional vector of which the elements are the Koopman eigenfunctions. Note that for a given current system state  $\mathbf{x}_m$  and its computed control action command  $\mathbf{u}_m$ , if the concatenated eigenfunctions  $\mathbf{g}$ , its element-wise inverse  $\mathbf{g}^{-1}$ , and the Koopman matrix  $\mathbf{K}$  are known, the future system state  $\mathbf{x}_{m+1}$  and control action command  $\mathbf{u}_{m+1}$  can be readily obtained. However, since discovering the Koopman eigenfunctions from finite samples of a dynamical system are still challenging, it calls for an AE-based deep learning-based approach as will be explained in the following sections.

In the meantime, since the control system exhibits control-affine nonlinear dynamics in (2) in this article, the global linearization expression (13) can be recast as [46]

$$\begin{bmatrix} \mathbf{g}_s(\mathbf{x}_{m+1}) \\ \mathbf{g}_u(\mathbf{x}_{m+1}, \mathbf{u}_{m+1}) \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{g}_s(\mathbf{x}_m) \\ \mathbf{g}_u(\mathbf{x}_m, \mathbf{u}_m) \end{bmatrix} \quad (14)$$

where  $\mathbf{g}_s$  is the  $d$ -dimensional vector of Koopman eigenfunctions that depend only on the system states,  $\mathbf{g}_u$  is the  $f$ -dimensional vector of Koopman eigenfunctions that depend on both system states and control action commands. For the purpose of utilizing linear control theory, it is required to ensure that the linear evolution of the Koopman eigenfunctions is explicitly related to the control action commands; hence, we simplify  $\mathbf{g}_u$  in (14) into  $\mathbf{g}_u(\mathbf{x}_m, \mathbf{u}_m) \approx \mathbf{u}_m$ , and thus, (14) becomes [46], [47], [48]

$$\begin{bmatrix} \mathbf{g}_s(\mathbf{x}_{m+1}) \\ \mathbf{u}_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{g}_s(\mathbf{x}_m) \\ \mathbf{u}_m \end{bmatrix} \quad (15)$$

where  $\mathbf{g}_s$  and  $\mathbf{g}$  are used interchangeably thereafter,  $\mathbf{K}_{11} \in \mathbb{R}^{d \times d}$  is the state-transition matrix in the Koopman matrix  $\mathbf{K} \in \mathbb{R}^{d+q} \times \mathbb{R}^{d+q}$ ,  $\mathbf{K}_{12} \in \mathbb{R}^{d \times q}$  is the control action matrix of the state dynamics,  $\mathbf{K}_{21} \in \mathbb{R}^{q \times d}$  is the state-dependent control action matrix, and  $\mathbf{K}_{22} \in \mathbb{R}^{q \times q}$  is the control action matrix of the control dynamics. Moreover, we define  $\mathbf{K}_s = [\mathbf{K}_{11} \ \mathbf{K}_{12}]$  as the state Koopman matrix representing the state dynamics, while  $\mathbf{K}_a = [\mathbf{K}_{21} \ \mathbf{K}_{22}]$  as the action Koopman matrix representing the control action dynamics. Note that the simplification of (14) in (15) comes at the cost of less accurate approximation of the system dynamics. For instance, if the term  $\sin(x_{m,1})\mathbf{u}_m$  appears in the system dynamics, it will be approximated in the Koopman model as  $c_1\mathbf{u}_m$ , where  $c_1 \in \mathbb{R}$  is a constant.

#### B. Sensing Koopman Autoencoder for State Prediction

The first component of the two-way Koopman AE architecture is the *sensing Koopman AE*, which plays two roles: 1) linearizing the nonlinear system dynamics and 2) predicting the future system states based on the linearized system dynamics. Note that linearization of nonlinear system dynamics helps reduce the complexity in terms of minimizing the computational time required to analyze the system and calculate its

optimal control action command, while state prediction helps reduce the communication cost [49]. The proposed sensing Koopman AE is a tripartite neural network that consists of an *encoder* that is related to the concatenated Koopman eigenfunctions  $\mathbf{g}$ , two fully connected hidden layers that constitute a finite-dimensional matrix representation of the Koopman operator  $\mathbf{K}$ , and the *decoder* which is an inverse function vector of the encoder  $\mathbf{g}^{-1}$ . Since we consider a remote control scenario, the Koopman AE is split into two parts, where the encoder is located at the sensor and actuator side, while the Koopman hidden layers and the decoder are situated at the controller side. Thus, the sensing Koopman AE is trained through split learning [20], [50] which will be detailed in Section III-D.

Once the sensing Koopman AE is trained, the sensor sends an encoded representation  $\mathbf{g}(\mathbf{x}_m)$  of the sampled system state  $\mathbf{x}_m$  and the controller obtains an estimated latent state representation  $\hat{\mathbf{g}}(\mathbf{x}_m)$  and concatenates it with the computed control action command  $\mathbf{u}_m$  as  $\mathbf{y}_m = [\hat{\mathbf{g}}(\mathbf{x}_m) \ \mathbf{u}_m]^T$ . Then, by multiplying  $\mathbf{y}_m$  with the sensing Koopman operator matrix  $\mathbf{K}_s$ , the remote controller obtains a linear state representation evolution as

$$\hat{\mathbf{g}}(\mathbf{x}_{m+1}) = \mathbf{K}_{11}\hat{\mathbf{g}}(\mathbf{x}_m) + \mathbf{K}_{12}\mathbf{u}_m \quad (16)$$

where  $\mathbf{K}_{11}$  and  $\mathbf{K}_{12}$  are the Koopman submatrices mentioned in (15). Moreover, by passing  $\mathbf{y}_m$  through the trained sensing Koopman matrix and the decoder, the remote controller obtains the predicted future system state as  $\bar{\mathbf{x}}_{m+1} = \mathbf{g}^{-1}(\mathbf{K}_s\mathbf{y}_m)$ . Moreover, by multiplying the sensing Koopman matrix  $m'$  times, the controller can predict the future system states at  $t = (m + m')\tau_o$  as

$$\bar{\mathbf{x}}_{m+m'} = \mathbf{g}^{-1}(\mathbf{K}_s^{m'}\mathbf{y}_m) \quad (17)$$

for all target prediction depth  $m' \in \mathbb{Z}_+$  which will be detailed in Section III-C.

To this end, the proposed real-time remote control of a closed-loop control system, aided by the Koopman AE, consists of two-phases. In the first phase, the remote controller receives the measured system states in real time from the sensor, computes its control action command, and sends it back to the actuator. Meanwhile, the sensor and controller train the sensing Koopman AE and the first phase is maintained until it is well-trained. In the second phase, the remote controller sends back a control signal informing the sensor to stop sending the system states once it is well-trained, and since the sensing Koopman AE is trained, the remote controller can predict the future system states.

### C. Linear Control Over Koopman Subspace

Having a closed-loop Koopman AE, an optimal control action can be computed over the linearized Koopman subspace by readily applying the linear control theory. That is, formulating an infinite-horizon discrete-time LQR over the linearized Koopman subspace rather than over the nonlinear system dynamics. Following [51], the control objective is a quadratic cost function given as:

$$\mathcal{J}(\hat{\mathbf{g}}(\mathbf{x}_m), \mathbf{u}_m) = \frac{1}{2} \sum_{m=0}^{\infty} \hat{\mathbf{g}}(\mathbf{x}_m)^T \mathbf{Q}_g \hat{\mathbf{g}}(\mathbf{x}_m) + \mathbf{u}_m^T \mathbf{R} \mathbf{u}_m \quad (18)$$

where  $\mathbf{Q}_g \in \mathbb{S}_+^{d \times d}$  is a positive semi-definite weight matrix of the latent state representation deviation cost, and  $\mathbf{R} \in \mathbb{S}_+^{q \times q}$  is a positive-definite weight matrix of the control action cost. Then, the optimization problem of an infinite-horizon discrete-time LQR to calculate the optimal control action is formulated as

$$\text{Minimize}_{\mathbf{u}_m} \mathcal{J}(\hat{\mathbf{g}}(\mathbf{x}_m), \mathbf{u}_m) \quad (19a)$$

$$\text{subject to: } \hat{\mathbf{g}}(\mathbf{x}_{m+1}) = \mathbf{K}_{11}\hat{\mathbf{g}}(\mathbf{x}_m) + \mathbf{K}_{12}\mathbf{u}_m. \quad (19b)$$

The problem in (19) is a convex optimization problem with a quadratic cost function and linear constraint, hence, the optimal control action can be readily obtained as

$$\mathbf{u}_m = -\mathbf{K}_{\text{LQR}} \hat{\mathbf{g}}(\mathbf{x}_m) \quad (20)$$

where  $\mathbf{K}_{\text{LQR}} = (\mathbf{R} + \mathbf{K}_{12}^T \mathbf{P} \mathbf{K}_{12})^{-1} \mathbf{K}_{12}^T \mathbf{P} \mathbf{K}_{11}$  is the feedback gain matrix, and  $\mathbf{P} = \mathbf{K}_{11}^T \mathbf{P} \mathbf{K}_{11} - \mathbf{K}_{11}^T \mathbf{P} \mathbf{K}_{12} (\mathbf{K}_{12}^T \mathbf{P} \mathbf{K}_{12} + \mathbf{R})^{-1} \mathbf{K}_{12}^T \mathbf{P} \mathbf{K}_{11} + \mathbf{Q}_g$  is a unique positive-definite matrix which satisfies the discrete-time algebraic Riccati equation (DARE) [52]. The Koopman-based controller outperforms the locally linearized-based controller in terms of control performance. This is because the locally linearized-based controller performs optimally near the equilibrium point, but the performance becomes poor for the points that are located far from the equilibrium point. In contrast, since the Koopman operator globally linearizes the system dynamics, the controller over the linearized Koopman subspace performs optimally for all points.

### D. Split Learning-Based Koopman Autoencoder

An optimization problem for obtaining the closed-loop Koopman AE in the considered WNCS can be formulated as

$$\text{minimize}_{\mathbf{K}, \mathbf{g}, \mathbf{g}^{-1}} \frac{1}{M_d} \sum_{m'=1}^{M_d} \left\| \hat{\mathbf{x}}_{m+m'} - \sum_{l=0}^{M_d-1} \mathbf{g}^{-1}(\mathbf{K}_s^{M_d-l} \mathbf{y}_{m+l}) w_l \right\|_2^2 \quad (21a)$$

$$\text{subject to: } \bar{\mathbf{x}}_m = \mathbf{g}^{-1}(\mathbf{y}_m) \quad (21b)$$

where  $\hat{\mathbf{x}}_{m+m'}$  is the estimated system state at time  $m+m'$ ,  $M_d$  is the target prediction depth considered in the training that represents the Koopman operator prediction time horizon checked in the training, and  $w_l$  is a weight parameter of the  $l$ th predicted system state at the controller side. On the one hand, when  $l=0$  and  $w_l=1$ , we call it a *special case* of the optimization problem in (21) suggested in [23] and [53]. On the other hand, when  $l>0$  and  $w_l=1/M_d$ , we call it a *general case* of the optimization problem in (21) proposed in this article. The *general case* of the optimization problem in (21) utilizes all the received latent state representations to predict the future system states compared to the *special case*. Since it is analytically hard to solve the formulated optimization problem in (21) for long-term target prediction depth, we solve it by exploiting deep learning over three separated unconstrained optimization subproblems formulated as follows.

- 1) A *reconstruction loss* is formulated to accurately reconstruct the system states. This loss measures the mean squared error (MSE) between the estimated system states  $\hat{\mathbf{x}}_m$  and the decoded estimated state representation

adjoined with the control action command  $\mathbf{g}^{-1}(\mathbf{y}_m)$  given as

$$\mathcal{L}_1 = \frac{1}{M_s} \sum_{m=1}^{M_s} \|\hat{\mathbf{x}}_m - \mathbf{g}^{-1}(\mathbf{y}_m)\|_2^2 \quad (22)$$

where  $M_s$  denotes the length of the received time-series signals in the first phase of the remote control of a closed-loop control system.

- 2) A *linear Dynamics loss* is formulated to obtain the linear finite-dimensional sensing Koopman matrix  $\mathbf{K}_s$  that ensures the linearity in the Koopman invariant subspace. This loss measures the MSE between the estimated state representation of the future system state adjoined with its future control action commands as  $\mathbf{y}_{m+m'}$  and the weighted sum of the multiplication of the sensing Koopman matrix with different target prediction depth during training and the estimated state representation of the received system states adjoined with the calculated control action commands  $\sum_{l=0}^{M_d-1} (\mathbf{K}_s^{M_d-l} \mathbf{y}_{m+l}) w_l$ , i.e.,

$$\mathcal{L}_2 = \frac{1}{M_d} \sum_{m'=1}^{M_d} \|\mathbf{y}_{m+m'} - \sum_{l=0}^{M_d-1} (\mathbf{K}_s^{M_d-l} \mathbf{y}_{m+l}) w_l\|_2^2. \quad (23)$$

Note that the linear latent state representation evolution in (23) for  $l=0$  is represented as  $\mathbf{K}_s^{M_d} \mathbf{y}_m = \mathbf{K}_{11}^{M_d} \mathbf{g}(\mathbf{x}_m) + \sum_{m'=1}^{M_d} \mathbf{K}_{11}^{m'-1} \mathbf{K}_{12} \mathbf{u}_{m+m'-1}$ .

- 3) A *prediction loss* is formulated to accurately predict the future system states. The prediction loss measures the MSE between the estimated future system states  $\hat{\mathbf{x}}_{m+m'}$  and decoded weighted sum of the multiplication of the sensing Koopman matrix with different target prediction depth considered in training and the estimated state representation of the received system states adjoined with the calculated control action commands  $\mathbf{g}^{-1}(\sum_{l=0}^{M_d-1} (\mathbf{K}_s^{M_d-l} \mathbf{y}_{m+l}) w_l)$ , i.e.,

$$\mathcal{L}_3 = \frac{1}{M_d} \sum_{m'=1}^{M_d} \|\hat{\mathbf{x}}_{m+m'} - \bar{\mathbf{x}}_{m+m'}\|_2^2 \quad (24)$$

where the future predicted system state  $\bar{\mathbf{x}}_{m+m'}$  in (24) is given as  $\bar{\mathbf{x}}_{m+m'} = \sum_{l=0}^{M_d-1} \mathbf{g}^{-1}(\mathbf{K}_s^{M_d-l} \mathbf{y}_{m+l}) w_l$ .

- 4) A *representation cost loss* is formulated to guarantee that the state representation deviation cost is equivalent to the system state deviation cost in the Jacobian linearization method. This loss measures the MSE between the state representation deviation cost  $\tilde{\mathbf{Q}}_g$  and the system state deviation cost  $\mathbf{Q}_x$ , given as

$$\mathcal{L}_4 = \frac{1}{M_s} \sum_{m=1}^{M_s} \|\hat{\mathbf{x}}_m \mathbf{Q}_x \hat{\mathbf{x}}_m - \hat{\mathbf{g}}(\mathbf{x}_m) \tilde{\mathbf{Q}}_g \hat{\mathbf{g}}(\mathbf{x}_m)\|_2^2. \quad (25)$$

Here, we set  $\mathbf{Q}_g = [\tilde{\mathbf{Q}}_g]_+$  as the final estimator, where  $[\cdot]_+$  truncates the nonpositive eigenvalues to zero to ensure the state representation deviation cost is a positive semi-definite weight matrix.

Overall, the total weighted-sum loss function to train the closed-loop Koopman AE for predicting the future system

states and calculating the optimal control action is given as

$$\mathcal{L} = c_1 \mathcal{L}_1 + c_2 \mathcal{L}_2 + c_3 \mathcal{L}_3 + c_4 \mathcal{L}_4 \quad (26)$$

for some positive coefficients  $c_1, c_2, c_3, c_4 \in \mathbb{Z}_+$ . The weights of the sensing Koopman AE can be trained based on the stochastic gradient descent (SGD) and backpropagation via feedback channels from the controller to the sensor. An early stopping strategy is utilized to avoid model overfitting, which enhances both the prediction accuracy and communication efficiency. Note that the sensor must consistently sample the system states and send them along with the latent state representations to the controller over wireless fading channels during the first phase of remote control to compute the total weighted-sum loss at the remote controller. Otherwise, if the system states are missed due to adverse channel conditions in the forward link, the remote controller predicts the future system states based on the last observed system states.

### E. Controlling Koopman Autoencoder for Control Prediction

To compensate for the missing control action command in case of adverse channel conditions in the reverse link, we propose controlling Koopman AE to linearize the action dynamics and predict the future control action commands at the actuator side. The proposed controlling Koopman AE is composed of three parts of the encoder that is located at the sensor side is the same as the encoder utilized in the sensing Koopman AE  $\mathbf{g}$ , the Koopman hidden layers at the actuator side  $\mathbf{K}'$ , and the decoder are located at the actuator side  $\mathbf{g}_a^{-1}$ . After the controlling Koopman AE is well-trained, if the remote controller sends a control action command  $\mathbf{u}_m$ , the actuator obtains a noisy control action command  $\hat{\mathbf{u}}_m$ , applies it on the plant, and concatenates it with the latent state representations  $\mathbf{g}(\mathbf{x}_m)$  as follows  $\mathbf{z}_m = [\mathbf{g}(\mathbf{x}_m) \quad \hat{\mathbf{u}}_m]^T$ . Then, by passing  $\mathbf{z}_m$  through the finite-dimensional controlling the Koopman matrix  $\mathbf{K}'_a$ , the linear control action command evolution at the actuator is given as

$$\hat{\mathbf{u}}_{m+1} = \mathbf{K}'_{21} \mathbf{g}(\mathbf{x}_m) + \mathbf{K}'_{22} \hat{\mathbf{u}}_m \quad (27)$$

where  $\mathbf{K}'_{21} \in \mathbb{R}^{q \times d}$  and  $\mathbf{K}'_{22} \in \mathbb{R}^{q \times q}$  are the trained Koopman submatrices at the actuator side representing the nonlinear control action dynamics.

Since the objective of the controlling Koopman AE is to predict the future control action commands for a long time, it is required to have the measurement functions, its inverse functions, and the finite-dimensional Koopman matrix. Therefore, we train the controlling Koopman AE through an optimization problem formulated as

$$\underset{\mathbf{K}', \mathbf{g}, \mathbf{g}_a^{-1}}{\text{minimize}} \quad \frac{1}{M_d} \sum_{m'=1}^{M_d} \|\mathbf{x}_{m+m'} - \tilde{\mathbf{x}}_{m+m'}\|_2^2 \quad (28a)$$

$$\text{subject to: } \tilde{\mathbf{x}}_m = \mathbf{g}_a^{-1}(\mathbf{z}_m) \quad (28b)$$

where  $\tilde{\mathbf{x}}_{m+m'} = \sum_{l=0}^{M_d-1} \mathbf{g}_a^{-1}(\mathbf{K}'_a{}^{M_d-l} \mathbf{z}_{m+l}) w'_l$  is the future predicted system states at the actuator side, and  $w'_l$  is the  $l$ th weight parameter of the predicted system state at the actuator. The problem in (28) is similar to the problem of the sensing Koopman AE in (21) except that the controlling Koopman

AE receive noisy control action commands and the original system states along with their state representations. As a result, we minimize a total weighted-sum of three separated loss functions similar to the sensing Koopman AE given as

$$\mathcal{L}' = c'_1 \mathcal{L}'_1 + c'_2 \mathcal{L}'_2 + c'_3 \mathcal{L}'_3 \quad (29)$$

where  $c'_1$ ,  $c'_2$ , and  $c'_3$  are positive weighting hyperparameters of the controlling Koopman AE. The loss functions  $\mathcal{L}'_1$ ,  $\mathcal{L}'_2$ , and  $\mathcal{L}'_3$  of the controlling Koopman AE in (29) are similar to the loss functions  $\mathcal{L}_1$ ,  $\mathcal{L}_2$ , and  $\mathcal{L}_3$  of the sensing Koopman AE, respectively, except that  $\hat{\mathbf{x}}_m$  is substituted by  $\mathbf{x}_m$  and  $\mathbf{y}_m$  by  $\mathbf{z}_m$ .

#### IV. SIMULATION RESULTS AND DISCUSSION

In this section, we investigate the performance of both sensing Koopman AE and controlling Koopman AE over wireless fading channels in an inverted cart-pole system. This control system is nonlinear and multidimensional. For a given time, the inverted cart-pole system is described by a 4-D state vector  $\mathbf{x} = [x \ v \ \theta \ \omega]^T$ , where  $x$  and  $v$  represent the horizontal position and velocity of the cart, respectively. The terms  $\theta$  and  $\omega$  are the vertical angle and angular velocity of the pendulum, respectively. The control action is described as a horizontal force applied on the cart  $u$ . Accordingly, the full nonlinear system dynamics are described as follows:

$$\begin{aligned} \frac{dx}{dt} &= v \\ \frac{d\theta}{dt} &= \omega \\ \frac{dv}{dt} &= \frac{-m_p^2 L^2 v \ c\theta \ s\theta + m_p L^2 (m_p L \omega^2 s\theta - \delta v) + m_p L^2 u}{m_p L^2 (m_c + m_p (1 - c\theta^2))} \\ \frac{d\omega}{dt} &= \frac{m_{pm} m_p v L s\theta - m_p L c\theta (m_p L \omega^2 s\theta - \delta v) + m_p L c\theta u}{m_p L^2 (m_c + m_p (1 - c\theta^2))} \end{aligned} \quad (30)$$

with  $m_{pm} = m_p + m_m$ ,  $c\theta = \cos(\theta)$ , and  $s\theta = \sin(\theta)$ . Here, we consider the following simulation parameters unless stated; otherwise, pendulum mass  $m_p = 1$  kg, cart mass  $m_c = 5$  kg, pendulum length  $L = 0.2$  m, gravitational acceleration  $v = -10$  m/s<sup>2</sup>, and cart damping  $\delta = -2$  Ns/m.

##### A. Data Generation and Training

The training data set of the sensing Koopman AE is generated at the controller based on the calculated control action commands and the received system states from the inverted-cart pole system with a sampling rate of  $\tau_0 = 10$  ms in the time interval  $m \in [0, |\mathcal{M}_{\text{Train}}|]$  and with a random initial condition in the range of  $[-0.5, 0.5]$  for each state dimension. For each initial condition, we solve the system of differential equations in (30) using the fourth-order Runge Kutta numerical method with 0.01 equally spaced step size and trajectory length of  $\mathcal{M}_{\text{Train}} = 250$  s. The training data set consists of  $T_t = 70$ , the validation data set of  $T_v = 20$ , and the test data set of  $T_e = 10$  trajectories. The AE weights are initialized with a normal distribution of the form  $w \sim \mathcal{N}(0, [2/d_{\text{in}}])$  where  $d_{\text{in}}$  is the number of neurons of the input layer. The AE weights

are trained to minimize the total weighted sum loss in (26) for the sensing Koopman AE, via Adam optimizer [54] with batch size of 64 and learning rate of  $10^{-4}$ . The encoder of the sensing Koopman AE consists of three fully connected layers that contain 128, 64, and 32 neurons with rectified linear unit (ReLU) activation, respectively, and the final layer contains  $d$  neurons with linear activation. The decoder begins with two fully connected layers with the width  $(d + q) \in \{2, 3, 4, 5\}$  (i.e., latent state representation dimension plus control action command dimension) and the rest follows the same structure of the encoder. The weighting hyperparameters of the sensing Koopman AE are set as  $c_1 = c_3 = 0.5$ ,  $c_2 = c_4 = 1$ . The wireless communication channels are assumed to follow Rayleigh fading channels. Here, the results are obtained based on the model with the lowest validation error of several training runs. We consider a maximum transmission power value of  $P_t = 20$  dBm, different SNR of  $\text{SNR} \in \{-10, 0, 10, 20\}$  dBm, distance between transmitter–receiver pair of  $D = 100$  m, and the path-loss exponent of  $\eta = 3$ . The data set of the controlling Koopman AE is generated at the actuator based on the system states and the received control action commands from the controller. The rest follows the same structure of the sensing Koopman AE with hyperparameters set as  $c'_1 = c'_3 = 0.5$ ,  $c'_2 = 1$ .

##### B. Evaluation Metrics

In this section, we present the evaluation metrics used to evaluate the proposed sensing Koopman AE and the controlling Koopman AE according to different evaluation metrics in terms of prediction accuracy, wireless communication efficiency, and control stability.

*Prediction Accuracy:* Since the main goal of the proposed sensing Koopman AE and the controlling Koopman AE are to predict future system states and control action commands, respectively, we use the normalized root mean square error (NRMSE) to evaluate the quality of prediction performance. The NRMSE is the root MSE between the predicted and observed signals in a prediction window of length  $M_p$  normalized by the norm of the difference between the minimum and maximum vectors of the observed signals

$$\text{NRMSE}_{M_p} = \frac{\sqrt{\frac{1}{M_p} \sum_{m=M_s+1}^{M_s+M_p} \|\bar{\mathbf{x}}_m - \hat{\mathbf{x}}_m\|_2^2}}{\|\max(\hat{\mathbf{x}}) - \min(\hat{\mathbf{x}})\|_2} \times 100 \quad (31)$$

where  $M_p$  is the prediction time horizon at test that the remote controller makes in the second phase of the remote controlling. Note that if the second phase becomes long, there will be error propagation that will degenerate the prediction performance. As a result, we have two options to enhance the prediction performance: 1) sending a new latent state representation to initialize the prediction and 2) shifting to the first phase of remote controlling to fine-tune the trained Koopman operator.

*Wireless Communication Metric:* To validate the performance of the proposed Koopman AE in terms of communication efficiency, we measure the maximum number of consecutive lost packets  $M_{\text{lost}}$  until the most recent successful transmission. Hence, the number of consecutive lost packets linearly increases with time if the transmitter–receiver



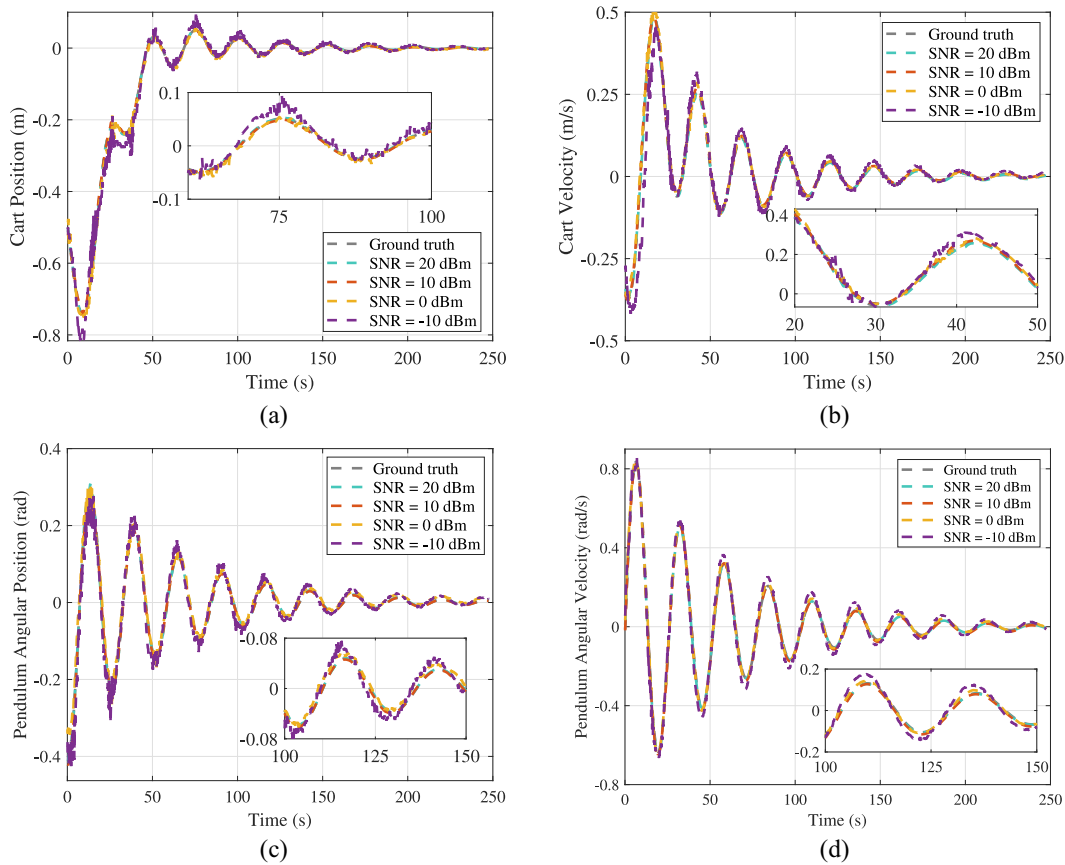


Fig. 2. Time-series of the predicted inverted cart-pole system states utilizing the proposed sensing Koopman AE before observing the ground truth. (a) Cart position. (b) Cart velocity. (c) Pendulum angular position. (d) Pendulum angular velocity.

pair experiences adverse channel conditions or the prediction accuracy of the Koopman AE reaches a predefined threshold.

*Control stability* is a control performance metric in terms of the control system response to any initial condition. The control system is said to be asymptotically stable if, for every  $\epsilon > 0$ ,  $m \geq 0$ , and a given control action command, there exists  $\|\mathbf{x}_m - \mathbf{x}_d\|_2^2 < \epsilon$  as  $m \rightarrow \infty$ , where  $\mathbf{x}_d \in \mathbb{R}^p$  is the desired system state [55]. Then, the control stability along the prediction time horizon at test is the MSCE, i.e., the time-averaged of squared system states deviation from its desired state along the prediction time horizon at the test, given as

$$\text{MSCE}_{M_p} = \frac{1}{M_p} \sum_{m=1}^{M_p} \|\mathbf{x}_m - \mathbf{x}_d\|_2^2. \quad (32)$$

Note that the control stability in the proposed approach depends on the prediction accuracy that are affected by the communication efficiency, highlighting the importance of jointly designing communication, learning, and control operations.

### C. Performance Evaluation of Split Koopman AE

In this section, we evaluate the performance of the proposed two-way split Koopman AE for different communication, prediction, and control parameters.

*SNR Versus State Prediction Accuracy:* Fig. 2 demonstrates the prediction performance of the proposed sensing Koopman

AE compared to the nonpredictive remote monitoring of the nonlinear inverted cart-pole system. The remote controller in the proposed sensing Koopman AE predicts the future time-series system states with different SNR and one-step target prediction depth considered in training compared to the remote controller in the nonpredictive remote monitoring that receives the nonlinear system states at each time slot over an ideal channel. It can be seen that the predicted system states of the proposed sensing Koopman AE with high SNR, i.e.,  $\text{SNR} \in \{10, 20\}$  dBm match closely the observed system states in the nonpredictive remote monitoring compared to the predicted system states with low SNR, i.e.,  $\text{SNR} \in \{-10, 0\}$  dBm.

The reason behind this result is that the stability of training the sensing Koopman AE relies on the communication reliability of the observed system states at the remote controller, highlighting the importance of co-designing the communication and deep learning operations. As a result, the state prediction accuracy is improved at the cost of increasing communication resources in terms of transmission power and allocated channel bandwidth. Moreover, the state prediction results in Fig. 2 emphasize the ability of the sensing Koopman AE in discovering the Koopman invariant subspace with four latent state representations and one-step target prediction depth considered in training. Then, the sensing Koopman AE globally linearizes the nonlinear system dynamics of the inverted cart-pole and predicts the future system states at the remote

TABLE I  
STATE AND ACTION PREDICTION NRMSE FOR DIFFERENT LATENT STATE REPRESENTATION DIMENSIONS AND SNR

Representation Dimensions $d$	SNR [dBm]	State Prediction NRMSE [%]	Action Prediction NRMSE [%]
1	-10	14.38	8.97
	0	14.15	3.87
	10	13.89	0.99
	20	<b>9.20</b>	<b>0.55</b>
2	-10	6.69	9.08
	0	5.79	3.85
	10	5.18	0.99
	20	<b>4.93</b>	<b>0.52</b>
3	-10	3.84	8.91
	0	2.95	3.57
	10	2.18	0.82
	20	<b>2.06</b>	<b>0.51</b>
4	-10	2.82	2.61
	0	1.67	0.87
	10	1.23	0.41
	20	<b>1.03</b>	<b>0.38</b>

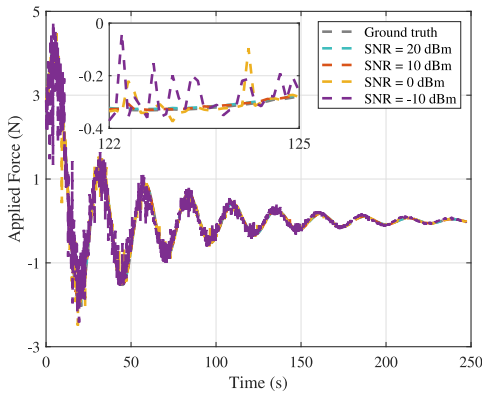


Fig. 3. Time-series of predicted force applied on the cart utilizing the proposed controlling Koopman AE before observing the ground truth.

controller based on the trained Koopman matrix, improving the forward link communication efficiency.

*SNR Versus Action Prediction Accuracy:* For the proposed controlling Koopman AE depicted in Fig. 3, the actuator predicts the future control action commands with different SNR values, and one-step target prediction depth considered in training compared to the actuator in the nonpredictive remote controlling of the nonlinear inverted cart-pole system that observes the control action commands calculated over the Jacobian linearization method at each time slot over an ideal channel. Here, the transmission power in both the forward and reverse communications are linked together since the observed control action command at the actuator is based on the observed system states affecting the control action prediction performance and the control stability. It is clear that the predicted control action commands in the controlling Koopman AE with high SNR, i.e.,  $\text{SNR} \in \{10, 20\}$  dBm approximately coincides with the observed control action commands compared to the predicted control action commands with low SNR,  $\text{SNR} \in \{-10, 0\}$  dBm.

The rationale behind this result is that the learning stability of the controlling Koopman AE depends on the

communication reliability of the observed control action commands at the actuator, highlighting the importance of jointly designing the communication, learning, and control operations. Hence, the action prediction accuracy is improved which yields stabilized control system at the expense of utilizing high communication resources in terms of transmission power and allocated channel bandwidth in the first phase of remote controlling. Moreover, the action prediction results in Fig. 3 depict the ability of the controlling Koopman AE in capturing the nonlinear action dynamics. Hence, the controlling Koopman AE has the ability to predict the future control action commands, improving both the forward and reverse links communication efficiency.

*SNR and Representation Dim Versus Prediction Accuracy:* As shown in Fig. 2, the state prediction performance of the sensing Koopman AE is improved at the cost of increasing the communication resources in terms of 4-D latent state representations and a high SNR value. Table I describes the state prediction performance of the sensing Koopman AE for different SNR values and different latent state representation dimensions. From this table, it is clear that the state prediction accuracy is improved at the cost of increasing the SNR for the same latent state representation dimension, the same sensor transmission power, the same distance from the sensor to the controller, one-step target prediction depth considered in training, and the same training period. The state prediction accuracy is almost the same in the lowest latent state representation dimension, i.e.,  $d = 1$ , for low SNR values, i.e.,  $\text{SNR} \in \{-10, 0, 10\}$  dBm. In contrast, the state prediction accuracy is improved by increasing the SNR values in the other latent state representation dimensions  $d \in \{2, 3, 4\}$ . The rationale behind the state prediction results in Table I is that the lowest latent state representation dimension is not enough to discover the Koopman invariant subspace in the observables space of the nonlinear inverted cart-pole system affecting the state prediction performance. Hence, for obtaining the high state prediction performance, the linearized Koopman subspace reflecting the nonlinear system dynamics is obtained

TABLE II  
STATE AND ACTION PREDICTION NRMSE FOR DIFFERENT LATENT STATE REPRESENTATION DIMENSIONS AND TRAINING PERIODS

Representation Dimension $d$	Trajectory Length [s]	State Prediction NRMSE [%]	Action Prediction NRMSE [%]
1	150	10.71	1.54
	250	9.20	0.55
	350	<b>7.99</b>	<b>0.29</b>
2	150	5.04	1.54
	250	4.93	0.52
	350	<b>3.73</b>	<b>0.35</b>
3	150	3.99	1.51
	250	2.06	0.51
	350	<b>1.22</b>	<b>0.29</b>
4	150	1.28	1.35
	250	1.03	0.38
	350	<b>0.96</b>	<b>0.22</b>

at the expense of increasing the communication resources in terms of selecting a high SNR value and a large latent state representation dimension, leading to a tradeoff between the communication cost and the prediction accuracy.

Table I demonstrates also the control action prediction performance of the controlling Koopman AE compared to the nonpredictive remote controlling of the nonlinear inverted cart-pole system. The controlling Koopman AE located at the actuator locally predicts the future control action commands for different SNR values and different latent state representation dimensions. Note that the control action prediction performance is improved at the cost of increasing the SNR value for the same latent state representation dimension, the same controller transmission power, the same distance from the controller to the actuator, one-step target prediction depth considered in training, and the same training period. Additionally, for the same SNR value, we notice that the control action prediction performance is almost the same for the low latent state representation dimensions of  $d \in \{1, 2, 3\}$  compared to the 4-D latent state representation. This demonstrates the tradeoff between the communication payload size and the control action prediction accuracy, in addition to the fact that the 4-D latent state representations quickly improve the control action prediction accuracy as a result of reflecting the nonlinear action dynamics in the controlling Koopman invariant subspace.

*Trajectory Length and Representation Dim Versus Prediction Accuracy:* As a result of the difficulty in identifying the nonlinear dynamics in an interpretable way, the controller in the nonpredictive remote controlling case receives the system states at each time slot and calculates the control action using the nonlinear control theory. In contrast, our proposed sensing Koopman AE identifies the nonlinear system dynamics in an interpretable linear form that is utilized by the remote controller to calculate the optimal control action using the linear control theory and to predict the future system states even without communication. Table II demonstrates the state prediction accuracy of the sensing Koopman AE for the same sensor transmission power, the same distance from the sensor to controller, different training periods, different latent state representation dimensions, and

one-step target prediction depth considered in training. The state prediction accuracy is improved as the training period increases for the same SNR value and the same latent state representation dimensions, highlighting the tradeoff between the state prediction accuracy and the communication payload size. The state prediction accuracy in Table II is improved as a result of increasing the number of observed system states per trajectory for the same SNR value and the same latent state representation dimension. The rationale behind this result is that increasing the number of observed system states per trajectory ensures the training data set is rich enough to represent the nonlinear system dynamics and guarantees the robustness of discovering the Koopman invariant subspace at the expense of increasing the communication resources.

Table II demonstrates also the control action prediction performance of the controlling Koopman AE for the same controller transmission power, the same distance from the sensor to controller, different training periods, different latent state representation dimensions, and one-step target prediction depth considered in training. It is clear that the control action prediction performance is improved at the cost of increasing the number of received control action commands per trajectory in the training data set for the same SNR value and the same latent state representation dimension, highlighting the tradeoff between the action prediction performance and the communication payload size. Moreover, the action prediction performance is almost the same for low latent state representation dimensions  $d \in \{1, 2, 3\}$ , the same target SNR, and the same trajectory length compared to the action prediction performance for the 4-D latent state representation. The reason behind this result is that the low latent state representation dimensions are insufficient to discover the controlling Koopman invariant subspace of the nonlinear action dynamics. Additionally, for the same latent state representation dimension, increasing the trajectory length increases the action prediction performance since the high sampling rate helps in describing the nonlinear action dynamics and discovering the controlling Koopman invariant subspace.

*SNR and Representation Dimensions Versus Training Loss:* Fig. 4 demonstrates the overall training loss of the sensing Koopman AE for different SNR values and different latent

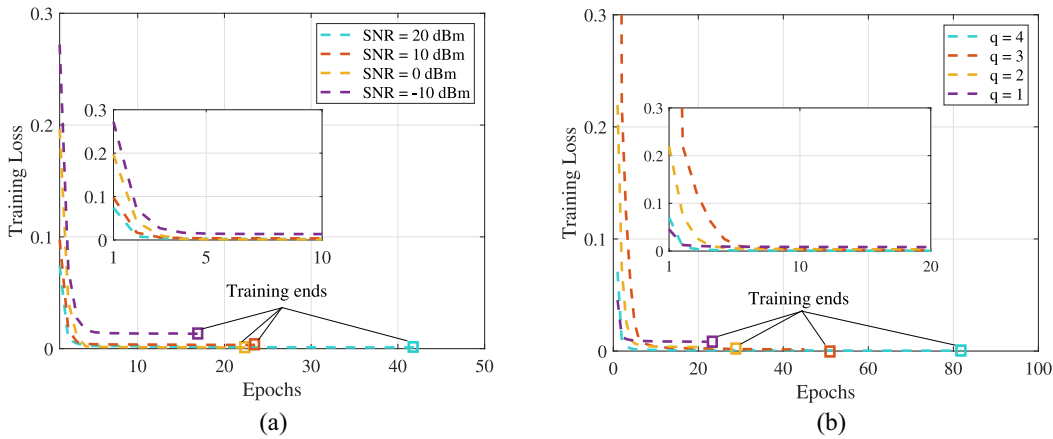


Fig. 4. Training loss of the sensing Koopman AE with (a) different SNR and (b) different latent state representation dimensions.

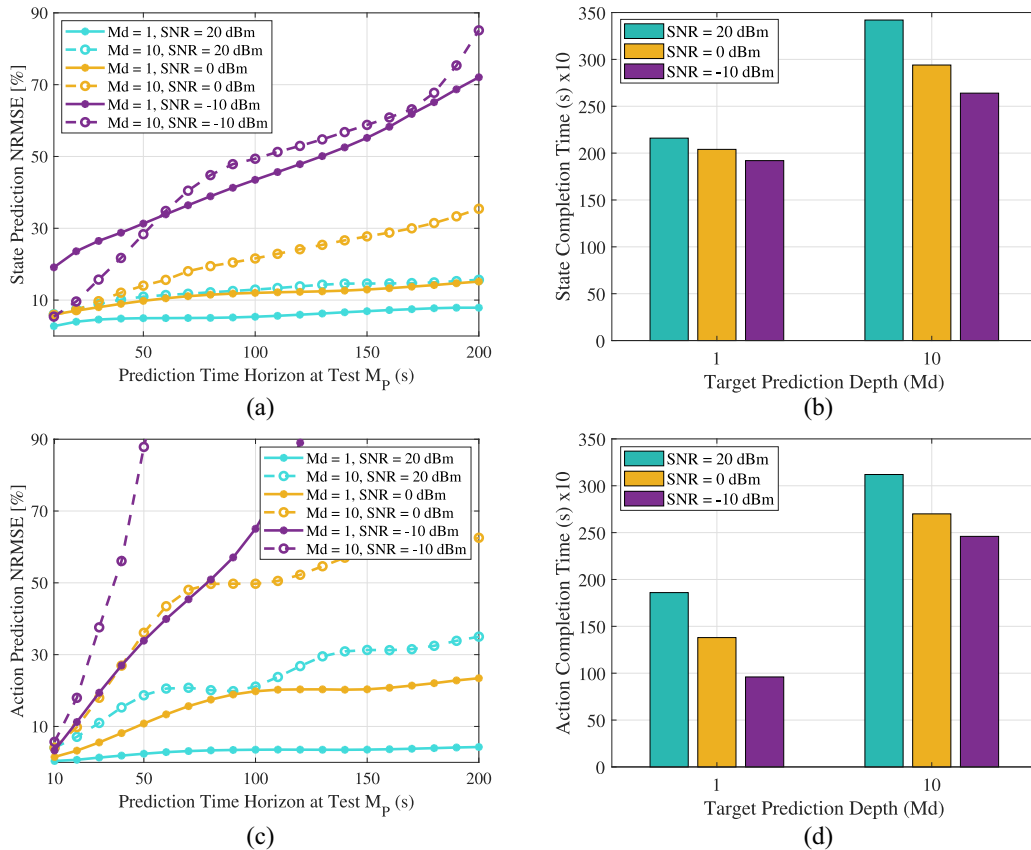


Fig. 5. Prediction performance and training completion time of both the proposed split sensing Koopman learning and the proposed split controlling closed-loop Koopman learning for different target prediction depths and different SNR values. (a) State prediction performance. (b) State training completion time. (c) Action prediction performance. (d) Action training completion time.

state representation dimensions. For the same latent state representation dimension, i.e.,  $d = 3$ , it is clear that the training loss with the lowest SNR value, i.e.,  $\text{SNR} = -10$  dBm converges faster compared to the other SNR values at the cost of achieving low training accuracy. In contrast, the training loss with the high SNR, i.e.,  $\text{SNR} = 20$  dBm converges slowly to the minimum value compared to the other SNR values, i.e.,  $\text{SNR} \in \{-10, 0, 10\}$  dBm. This in turn shows the impact of the SNR value on the training accuracy in which the received data samples to the another part of the sensing Koopman AE affects

the training accuracy, convergence speed, and communication payload size.

For the same SNR value, we can see that the training loss of the sensing Koopman AE slowly converges as the representation dimension increases. More specifically, the communication payload size of the forward and backward propagation signals is proportional to the number of weights parameters in the Koopman hidden layer. As a result, increasing the latent state representation dimensions results in a high latency for transmitting the forward and backward propagation

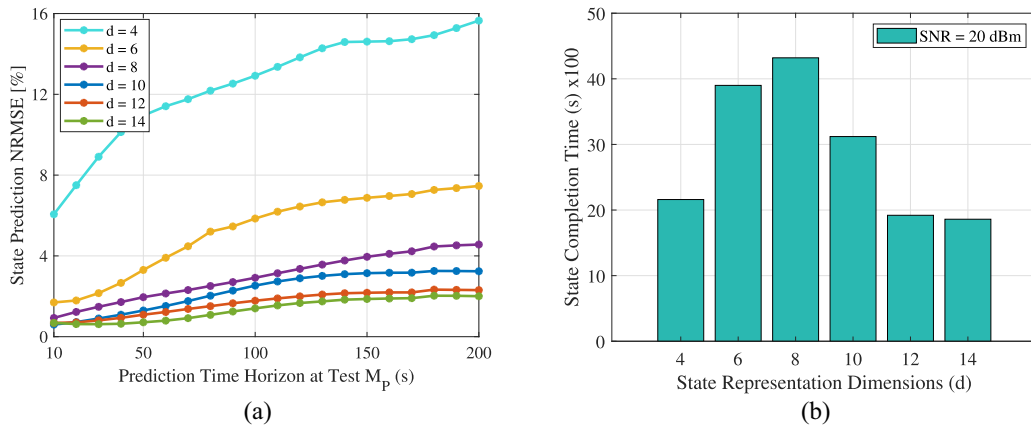


Fig. 6. Prediction performance and training completion time of the proposed sensing Koopman AE with a ten-step target prediction depth, different latent state representation dimensions  $d \in \{4, 6, 8, 10, 12, 14\}$ , and SNR = 20 dBm. (a) State prediction performance. (b) State training completion time.

signals which leads to high prediction accuracy, highlighting the tradeoff between the prediction accuracy, transmission latency, and communication payload size.

**Target Prediction Depth Versus Prediction Performance and Training Completion Time:** To dive deeper into the benefits of both the sensing Koopman AE and the controlling Koopman AE, we increase the range of random initial conditions to generate the training data set as  $[-2.0, 2.0]$  for each state dimension and also increase the trajectory length as  $\mathcal{M}_{\text{Train}} = 500$  s. Here, the results are obtained based on averaging several training runs with different seeds. Fig. 5(a) and (c) show the prediction performance of the sensing Koopman AE and the controlling Koopman AE, respectively, for different target prediction depths considered in training, different SNR values, and 4-D latent state representations. For the same target prediction depth considered in training, we can see that the state prediction performance decreases as the prediction time horizon at the test increases due to the error propagation as the prediction time increases. As a result, based on a predefined prediction threshold, we send a new latent state representation or shift to the first phase of remote control to enhance the state prediction performance.

For the same SNR value, it is clear that the prediction performance of the split Koopman AE with a one-step target prediction depth considered in training is better than that with a ten-step target prediction depth over different prediction time horizons at the test time, i.e., increasing the target prediction depth deteriorates the prediction performance over different prediction time horizons at the test. The rationale behind this result is that the small errors in training the Koopman operator with a high target prediction depth during the first phase of remote control lead to large errors when evaluating the prediction performance for a large prediction time horizon at the test.

Fig. 5(b) and (d) show the training completion time required to train the sensing Koopman AE and the controlling Koopman AE, respectively, for different SNR values and different target prediction depths. For the same SNR value, it is clear that the training completion time associated with a ten-step target prediction depth is larger than that with a one-step target prediction depth due to the high computational complexity associated with a ten-step target prediction depth considered

in training. Hence, increasing the target prediction depth considered in training leads to a high training completion time and a high communication payload size due to the necessity to transmit the system states in the first phase of remote control until the sensing Koopman AE is well-trained.

By analyzing both the prediction performance and the training completion time of the split Koopman AE, we can see that the split Koopman AE with one-step target prediction depth and a low SNR value has better prediction performance compared to that with ten-step target prediction and a high SNR value. In addition, the split Koopman AE with one-step target prediction depth has a low communication cost in terms of transmission power and communication payload size. For instance, the split Koopman AE with one-step target prediction depth and SNR = 0 dBm has better prediction performance compared to that with ten-step target prediction depth and SNR = 20 dBm. Overall, increasing the target prediction depth with the current hyperparameter setting deteriorates the prediction performance compared to the low target prediction depth. Hence, we next study the effect of changing the latent state representation dimensions on the prediction performance with a ten-step target prediction depth for the same SNR value.

**Representation Dim Versus Prediction Performance and Training Completion Time:** Fig. 6 presents the prediction performance and training completion time of the sensing Koopman AE for different latent state representation dimensions, ten-step target prediction depth considered in training, SNR = 20 dBm, and the same previous hyperparameter setting. First, it is clear from Fig. 6(a) that the prediction performance of the sensing Koopman AE with a ten-step target prediction depth is improved at the cost of increasing the latent state representation dimensions, highlighting the tradeoff between the target prediction depth considered in training and the latent state representation dimensions. Furthermore, we can see that the prediction performance with a ten-step target prediction depth starts to saturate after 12-D latent state representations over different prediction time horizons at the test. The reason behind this result is that increasing the latent state representation dimensions in the high target prediction depth setting ensures discovering the Koopman invariant subspace that guarantees the high prediction performance for

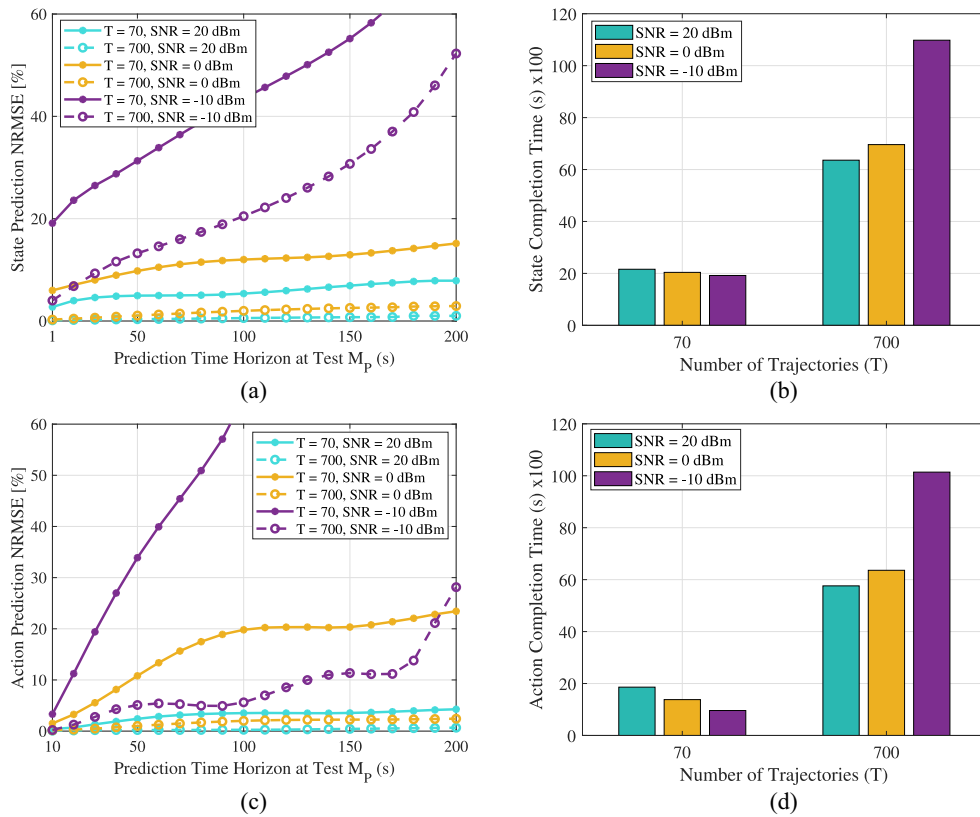


Fig. 7. Prediction performance and training completion time of the proposed sensing Koopman AE with a one-step target prediction depth, different number of trajectories in the training data set  $T_t = \{70, 700\}$ , different SNR values  $\text{SNR} = \{20, 0, -10\}$  dBm. (a) State prediction performance. (b) State training completion time. (c) Action prediction performance. (d) Action training completion time.

a long prediction time horizon at the test. Hence, selecting a large dimension of latent state representations with a large target prediction depth is instrumental in casting nonlinear system dynamics in a linear form at the expense of increasing the communication payload size, leading to a tradeoff between the communication cost and the prediction performance with a high target prediction depth considered in training.

Fig. 6(b) shows the training completion time required to train the sensing Koopman AE with a ten-step target prediction depth considered in training and different latent state representation dimensions. We can see that the training completion time increases as the latent state representation dimensions increase until 8-D latent state representations, after which it starts to decrease until twelve latent state representation dimensions, then it is almost the same. The reason behind this result is that increasing the latent state representation dimensions in the large target prediction depth considered in training hastens the convergence speed of the sensing Koopman AE training to discover the Koopman invariant subspace. As a result, reducing the communication payload size required to observe the system states in the first phase of remote control until the sensing Koopman AE is well trained.

*Trajectories Number Versus Prediction Performance and Training Completion Time:* Fig. 7(a) and (c) demonstrates the prediction performance of the sensing Koopman AE and the controlling Koopman AE, respectively, with a one-step target prediction depth considered in training, 4-D latent state

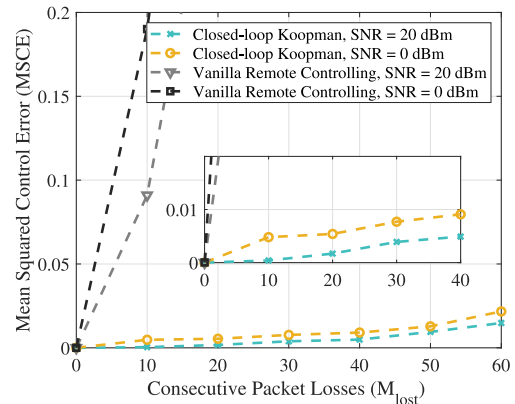


Fig. 8. Control stability for the split Koopman AE and the nonpredictive remote controlling over different values of consecutive packet losses with different SNR values.

representations, the same previous hyperparameter setting, different SNR values, and a different number of trajectories in the training data set. For the same SNR value, it is clear that the prediction performance of the split Koopman AE is improved over different prediction time horizons at the test by increasing the number of trajectories in the training data set as a result of well representing the nonlinear system dynamics by increasing the number of trajectories in the training data set. For the same number of trajectories, we can see the impact of the SNR on the prediction performance, in which the prediction performance with a high SNR value

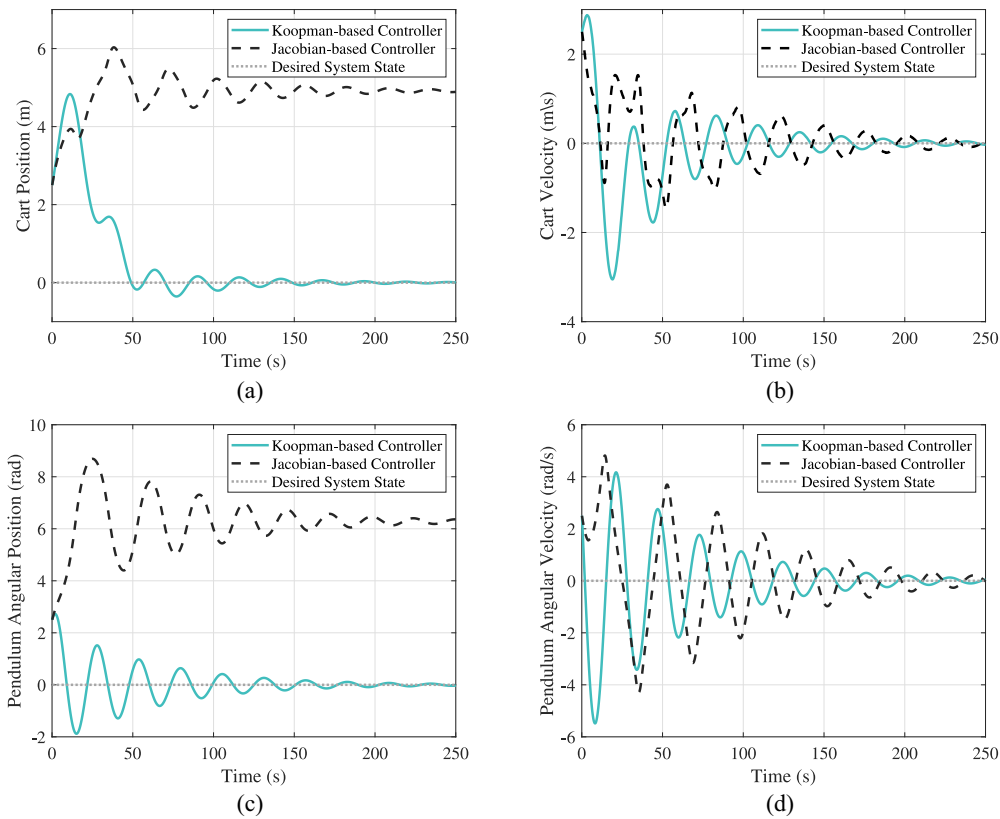


Fig. 9. Comparison of Koopman-based controller versus Jacobian-based controller. (a) Cart position. (b) Cart velocity. (c) Pendulum angular position. (d) Pendulum angular velocity.

is higher than that with a low SNR value. For example, the controlling Koopman AE with  $T_t = 700$  and SNR = 20 dBm has higher prediction performance compared to that with the other SNR values, i.e., SNR  $\in \{0, -10\}$  dBm. In addition, the controlling Koopman AE with  $T_t = 70$  and SNR = 20 dBm has almost the same prediction performance compared to that with  $T_t = 700$  and SNR = 0 dBm while it has better prediction performance compared to the others with different number of trajectories and low SNR values. This presents the relationship between the SNR value and the number of trajectories in the training data set, in which a large number of trajectories in the training data set with a high SNR value has a much larger prediction performance compared to those that have the same number of trajectories with low SNR values. Otherwise, choosing a low number of trajectories in the training data set with a high SNR value has better prediction performance compared to those with a large number of trajectories in the training data set with a low SNR value, highlighting the tradeoff between the communication payload size and the SNR value in improving the prediction performance.

Fig. 7(b) and (d) shows the training completion time required to train the sensing Koopman AE and the controlling Koopman AE, respectively, for a different number of trajectories in the training data set, different SNR values, and the same hyperparameter setting. For the same SNR value, it is clear that the training completion time increases as the number of trajectories in the training data set increases as a result of increasing the payload size of the forward and backward propagation signals to train the split Koopman AE. For

the same number of trajectories in the training data set, it is clear that the training completion time of the split Koopman AE with a large number of trajectories in the training data set decreases as the SNR value increases. The reason behind this result is that observations received at another part of the split Koopman AE are sufficient and reliable to discover the Koopman invariant subspace. In contrast, decreasing the SNR value for the same large number of trajectories in the training data set increases the training completion time, leading to a tradeoff between the training completion time and the communication cost in terms of the SNR value and the number of trajectories in the training data set.

*Control Stability Versus Consecutive Packet Losses:* Fig. 8 illustrates the control stability, i.e., the MSCE, over different values of consecutive packet losses with different SNR values. It is clear that the MSCE for the split Koopman AE is almost the same as the number of consecutive packet losses increases compared to the nonpredictive remote control. The rationale behind this result is that the sensing Koopman AE at the remote controller compensates for the missing system states by locally predicting them after observing a sufficient number of system states along with their latent state representations over favorable channel conditions with different SNR values. Moreover, the actuator locally predicts the missing control action commands utilizing the trained controlling Koopman AE after observing a sufficient number of control action commands, thereby improving the communication efficiency and the control performance. The control performance of the proposed split Koopman AE with SNR = 0 dBm is less than that of one with

SNR = 20 dBm as a result of the communication reliability impact on the prediction accuracy.

*Koopman-Based Versus Jacobian-Based Linear Controller:* In Fig. 9, we show the inverted cart-pole system states for the Koopman-based controller and Jacobian-based controller. Here, the inverted cart-pole system is assumed to be operated from an initial condition far from the equilibrium point, i.e.,  $\mathbf{x}_0 = [2.5 \ 2.5 \ 2.5 \ 2.5]^T$ . Note that the Koopman-based linear controller drives the inverted cart-pole system to the desired states compared to the Jacobian-based controller. The rationale behind failing the Jacobian-based controller in stabilizing the nonlinear system is that the initial condition is far from the equilibrium point,  $\mathbf{x}_d = [0 \ 0 \ 0 \ 0]^T$  compared to the Koopman-based linear controller that is robust against the initial condition as a result of obtaining a linearized Koopman invariant subspace reflecting the nonlinear dynamics with high accuracy.

## V. CONCLUSION

In this article, we proposed a two-way Koopman AE split learning framework for closed-loop real-time remote control with a sensing Koopman AE and a controlling Koopman AE that predict missing packets of state and action information, respectively. Numerical results demonstrate that the proposed approach predicts the future control system state and action with high accuracy for a practical range of target SNR, latent representation dimensions, and training periods. Leveraging the proposed method, developing multiple access schemes, and extending its applicability to multiple closed-loop control systems could be interesting topics for future research. To further improve the communication efficiency, transmitting partial state observations emanating from utilizing multiple distributed sensors exploiting a scheduling scheme could be another interesting topic for future work.

## REFERENCES

- [1] A. M. Girgis, H. Seo, J. Park, M. Bennis, and J. Choi, "Split learning meets Koopman theory for wireless remote monitoring and prediction," in *Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2021, pp. 1191–1196.
- [2] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G wireless communications: Vision and potential techniques," *IEEE Netw.*, vol. 33, no. 4, pp. 70–75, Jul./Aug. 2019.
- [3] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.
- [4] M. Latva-Aho, K. Leppänen, F. Clazzer, and A. Munari, *Key Drivers And Research Challenges for 6G Ubiquitous Wireless Intelligence*, 6G Flagship, Univ. Oulu, Oulu, Finland, 2020.
- [5] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.
- [6] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, "Remote surgery case: Robot-assisted teleneurosurgery," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2004, pp. 819–823.
- [7] C.-F. Liu and M. Bennis, "Taming the tail of maximal information age in wireless industrial networks," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2442–2446, Dec. 2019.
- [8] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7907–7922, Nov. 2019.
- [9] P. Razzaghi, E. Al Khatib, S. Bakhtiari, and Y. Hurmuzlu, "Real time control of tethered satellite systems to de-orbit space debris," *Aerosp. Sci. Technol.*, vol. 109, Feb. 2021, Art. no. 106379.
- [10] M. Iwabuchi, A. Benjebbour, Y. Kishiyama, and Y. Okumura, "Field experiments on 5G ultra-reliable low-latency communication (URLLC)," *NTT Docomo Tech. J.*, vol. 20, no. 1, pp. 14–23, Jul. 2018.
- [11] M. Angelichinoski, K. F. Trillingsgaard, and P. Popovski, "A statistical learning approach to ultra-reliable low latency communication," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5153–5166, Jul. 2019.
- [12] P. Popovski *et al.*, "Wireless access for ultra-reliable low-latency communication (URLLC): Principles and building blocks," *IEEE Netw.*, vol. 32, no. 2, pp. 16–23, Mar./Apr. 2018.
- [13] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
- [14] J. Park and P. Popovski, "Coverage and rate of downlink sequence transmissions with reliability guarantees," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 722–725, Dec. 2017.
- [15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.
- [16] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
- [17] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.
- [18] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space," *Proc. Nat. Acad. Sci. United States America*, vol. 17, no. 5, p. 315, 1931.
- [19] G. D. Birkhoff and B. O. Koopman, "Recent contributions to the ergodic theory," *Proc. Nat. Acad. Sci. United States America*, vol. 18, no. 3, p. 279, 1932.
- [20] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," in *Proc. ICLR Wksp. AI Social Good*, May 2019, pp. 1–7.
- [21] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.
- [22] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," 2017, *arXiv:1703.00395*.
- [23] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nat. Commun.*, vol. 9, no. 1, pp. 1–10, 2018.
- [24] G. Zhao, M. A. Imran, Z. Pang, Z. Chen, and L. Li, "Toward real-time control in future wireless networks: Communication-control co-design," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 138–144, Feb. 2019.
- [25] A. M. Girgis, J. Park, M. Bennis, and M. Debbah, "Predictive control and communication co-design via two-way Gaussian process regression and AoI-aware scheduling," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 7077–7093, Oct. 2021.
- [26] M. Yu, S. Cai, and V. K. Lau, "Event-driven sensor scheduling for mission-critical control applications," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1537–1549, Mar. 2019.
- [27] J. Li, P. Zeng, X. Zong, M. Zheng, and X. Zhang, "Communication and control co-design for wireless sensor networked control systems," in *Proc. 11th World Congr. Intell. Control Autom.*, 2014, pp. 156–161.
- [28] K. Gatsis, M. Pajic, A. Ribeiro, and G. J. Pappas, "Opportunistic control over shared wireless channels," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3140–3155, Dec. 2015.
- [29] B. Chang, L. Zhang, L. Li, G. Zhao, and Z. Chen, "Optimizing resource allocation in URLLC for real-time wireless control systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8916–8927, Sep. 2019.
- [30] M. Eisen, M. M. Rashid, K. Gatsis, D. Cavalcanti, N. Himayat, and A. Ribeiro, "Control aware radio resource allocation in low latency wireless control systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7878–7890, Oct. 2019.
- [31] M. Eisen, M. M. Rashid, K. Gatsis, D. Cavalcanti, N. Himayat, and A. Ribeiro, "Control aware communication design for time sensitive wireless systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 4584–4588.
- [32] W. Wang, R. Postoyan, D. Nešić, and W. M. H. Heemels, "Stabilization of nonlinear systems using state-feedback periodic event-triggered controllers," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, 2016, pp. 6808–6813.
- [33] E. Aranda-Escolástico, M. Abdelrahim, M. Guinaldo, S. Dormido, and W. Heemels, "Design of periodic event-triggered control for polynomial systems: A delay system approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7887–7892, 2017.
- [34] R. Postoyan, A. Anta, W. Heemels, P. Tabuada, and D. Nešić, "Periodic event-triggered control for nonlinear systems," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 7397–7402.



- [35] H. Li and Y. Shi, "Network-based predictive control for constrained nonlinear systems with two-channel packet dropouts," *IEEE Trans. Ind. Electron.*, vol. 61, no. 3, pp. 1574–1582, Mar. 2014.
- [36] D. M. de la Peña and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2076–2089, Oct. 2008.
- [37] M. S. Fadali and A. Visioli, *Digital Control Engineering: Analysis and Design*. London, U.K.: Academic, 2013.
- [38] M. R. Taylor and P. Bhatthawala, "Linearization of nonlinear differential equation by Taylor's series expansion and use of Jacobian linearization process," *Int. J. Theor. Appl. Sci.*, vol. 4, no. 1, pp. 36–38, 2011.
- [39] I. Mezić, "Koopman operator spectrum and data analysis," 2017, *arXiv:1702.07597*.
- [40] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech.*, vol. 656, pp. 5–28, Jul. 2010.
- [41] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM J. Appl. Dyn. Syst.*, vol. 15, no. 1, pp. 142–161, 2016.
- [42] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, 2016, pp. 6492–6499.
- [43] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, Jul. 2018.
- [44] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Generalizing Koopman theory to allow for inputs and control," *SIAM J. Appl. Dyn. Syst.*, vol. 17, no. 1, pp. 909–930, 2018.
- [45] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern Koopman theory for dynamical systems," 2021, *arXiv:2102.12086*.
- [46] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control," 2017, *arXiv:1707.01146*.
- [47] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey, "Local Koopman operators for data-driven control of robotic systems," in *Proc. Robot. Sci. Syst.*, 2019, pp. 1–9.
- [48] M. Bonnert and U. Konigorski, "Estimating Koopman invariant subspaces of excited systems using artificial neural networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1156–1162, 2020.
- [49] S. J. Guastello and L. S. Liebovitch, *Introduction to Nonlinear Dynamics and Complexity*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [50] Y. Koda *et al.*, "Communication-efficient multimodal split learning for mmWave received power prediction," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1284–1288, Jun. 2020.
- [51] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS One*, vol. 11, no. 2, 2016, Art. no. e0150171.
- [52] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [53] Y. Xiao, X. Zhang, X. Xu, X. Liu, and J. Liu, "A deep learning framework based on Koopman operator for data-driven modeling of vehicle dynamics," 2020, *arXiv:2007.02219*.
- [54] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
- [55] K. M. Hangos, J. Bokor, and G. Szederkényi, *Analysis and Control of Nonlinear Process Systems*. vol. 13. Berlin, Germany: Springer, 2004.



**Abanoub M. Girgis** (Member, IEEE) received the B.Sc. degree (Hons.) in electronics and communications engineering from the Thebes Higher Institute of Engineering, Cairo, Egypt, in 2013, and the M.Sc. degree in electronics and communications engineering from Ain Shams University, Cairo, in 2018. He is currently pursuing the Ph.D. degree with the Centre for Wireless Communication, University of Oulu, Oulu, Finland.

He was a Teaching Assistant with the Thebes Higher Institute of Engineering from 2013 to 2018. His main research interests include cyber-physical systems, communication and control co-design, semantic communication, massive MIMO, and signal processing.



**Hywoon Seo** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2012, 2014, and 2020, respectively.

He is an Assistant Professor with the Department of Electronics and Communications Engineering, Kwangju University, Seoul, South Korea. He was a Postdoctoral Research Fellow with Seoul National University, Seoul, and the University of Oulu, Oulu, Finland. His research interests include wireless communication, Internet of Things, physical layer security and privacy, vehicle-to-everything communications, and deep and distributed learning.

Dr. Seo received the Gold Prize in the 24th Samsung Humantech Paper Award in 2018.



**Jihong Park** (Senior Member, IEEE) received the B.S. and Ph.D. degrees from Yonsei University, Seoul, South Korea, in 2009 and 2016, respectively.

He is currently a Lecturer (Assistant Professor) with the School of Information Technology, Deakin University, Geelong, VIC, Australia. His research interests include distributed machine learning, control, and resource management, as well as their applications to 6G semantic, AI-native, and nonterrestrial communications.

Dr. Park received the IEEE GLOBECOM Student Travel Grant and the IEEE Seoul Section Student Paper Contest Bronze Prize in 2014, the 6th IDIS-ETNEWS Paper Award, and the FL-IJCAI Best Student Paper Award in 2022. He served as a Conference/Workshop Program Committee Member for IEEE GLOBECOM, ICC, and INFOCOM, and for NeurIPS, ICML, and IJCAI. He is currently an Associate Editor of *Data Science for Communications* (Frontiers) and *Signal Processing for Communications*. He is a member of ACM.



**Mehdi Bennis** (Fellow, IEEE) received the M.Sc. degree in electrical engineering jointly from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, and the Eurecom Institute, Biot, France, in 2002, and the Ph.D. degree in spectrum sharing for future mobile cellular systems from the University of Oulu, Oulu, Finland, in 2009.

He is a Full (tenured) Professor with the Centre for Wireless Communications, University of Oulu, where he is the Head of the Intelligent Connectivity and Networks/Systems Group (ICON). He has published more than 200 research papers in international conferences, journals, and book chapters. His main research interests are in radio resource management, game theory, and distributed AI in 5G/6G networks.

Dr. Bennis has been the recipient of several prestigious awards, including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the Journal of Wireless Communications and Networks, the all-University of Oulu Award for Research, the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award, and the 2020 Clarivate Highly Cited Researcher by the Web of Science. He is an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS and a Specialty Chief Editor for *Data Science for Communications* (Frontiers) and *Frontiers in Communications and Networks* (Frontiers).



**Jinho Choi** (Senior Member, IEEE) received the B.E. degree (*magna cum laude*) in electronics engineering from Sogang University, Seoul, South Korea, in 1989, and the M.S.E. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1991 and 1994, respectively.

He is a Professor with the School of Information Technology, Deakin University, Geelong, VIC, Australia. Prior to joining Deakin in 2018, he was with Swansea University, Swansea, U.K., as a Professor/Chair of Wireless, and Gwangju Institute of Science and Technology, Gwangju, South Korea, as a Professor. He authored two books published by Cambridge University Press in 2006 and 2010. His research interests include the Internet of Things, wireless communications, and statistical signal processing.

Prof. Choi received a number of best paper awards, including the 1999 Best Paper Award for Signal Processing from EURASIP. He is on the list of Worlds Top 2% Scientists by Stanford University in 2020 and 2021. He is currently an Editor of IEEE WIRELESS COMMUNICATIONS LETTERS and a Division Editor of *Journal of Communications and Networks*. He has also served as an Associate Editor or Editor for other journals, including IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Journal of Communications and Networks*, and *ETRI journal*.