Reinforcement Learning-Based Optimal Sensor Placement for Spatiotemporal Modeling

Zhi Wang, Han-Xiong Li, *Fellow, IEEE*, and Chunlin Chen, *Member, IEEE*

***Abstract*—A reinforcement learning-based method is proposed for optimal sensor placement in the spatial domain for modeling distributed parameter systems (DPSs). First, a low-dimensional subspace, derived by Karhunen–Loève decomposition, is identified to capture the dominant dynamic features of the DPS. Second, a spatial objective function is proposed for the sensor placement. This function is defined in the obtained low-dimensional subspace by exploiting the time-space separation property of distributed processes, and in turn aims at minimizing the modeling error over the entire time and space domain. Third, the sensor placement configuration is mathematically formulated as a Markov decision process (MDP) with specified elements. Finally, the sensor locations are optimized through learning the optimal policies of the MDP according to the spatial objective function. The experimental results of a simulated catalytic rod and a real snap curing oven system are provided to demonstrate the feasibility and efficiency of the proposed method in solving the combinatorial optimization problems, such as optimal sensor placement.**

***Index Terms*—Distributed parameter systems (DPSs), Karhunen–Loève decomposition (KLD), optimal sensor placement, reinforcement learning (RL), spatiotemporal modeling.**

I. Introduction

OPTIMAL sensor placement techniques play a significant role in spatiotemporal modeling [1] of the distributed parameter systems (DPSs). The sensor failures of not responding precisely can cause harmful influences on the entire life of the relevant process equipment, and increase operational difficulties to fulfill specific environmental needs. The modeling and control of DPSs are usually confined by the type and corresponding cost of available sensors. To ensure good reliability for the modeling and control of DPSs, the available information measured by the limited number of sensors should be efficiently utilized to provide an overall estimation for the entire area of the physical system, such as the unobserved states and unknown parameters. It may be possible to install a large number of sensors to measure numerous aspects of the physical field. However, this can be infeasible in practice due to expensive initial and maintenance costs. The state and parameter estimation techniques can provide an alternative solution for this problem, where other variables and parameters can be reconstructed accurately by strategically measuring the essential variables with a limited number of sensors. In order to obtain the most benefit from this technique, the available sensors should be placed at *optimal* locations.

Optimizing the sensor locations within a distributed process is challenging since most distributed processes are intrinsically nonlinear with infinite dimensions. Based on the dynamic linear systems theory, early methods for state estimation in DPSs exploited the optimal state-space observers derived from approximation models of the partial differential equation (PDE), such as finite difference method [2] or finite element method [3]. Many measures had been exploited to sensor placement on a distributed system, such as error covariance matrix of Kalman filters [4], the trace and the determinant of error matrix [5], estimation error caused by the unobservable subspace [6], and variable measurement structures [7]. The other kind of measures for sensor placement was based on the observability Gramian or the observability matrix [8], such as the smallest eigenvalue/determinant/trace of the inverse of the Gramian [9], the condition number of the observability matrix [10], and the trace and the spectral norm of the observability Gramian [11]. Other measures included measurement independence [12], the determinant of the Fisher information matrix [13], reliability using the probability of sensor failure [14], cost constrained by data reconciliation [15], the best compromise between the measurement cost, the process information [16], etc. These early alternatives had been applied only to linear systems and/or to a small number of sensors without any general systematic method [17]. Most of the above approaches relied on complex control assumptions and schemes, or an exhaustive search over a large set of candidate placements that were defined beforehand. Therefore, they were infeasible for complex nonlinear systems which required a high-dimensional representation.

Assume that we have only m sensors out of n candidate placements ($m < n$), where n depends on the resolution of the

Manuscript received August 6, 2018; accepted February 22, 2019. Date of publication March 18, 2019; date of current version May 7, 2020. This work was supported in part by the GRF Project from the RGC of Hong Kong under Grant CityU: 11205615, and in part by the National Key Research and Development Program of China under Grant 2016YFD0702100. This paper was recommended by Associate Editor R. Selmic. *(Corresponding author: Han-Xiong Li.)*

Z. Wang is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China, and also with the State Key Laboratory of High Performance Complex Manufacturing, Central South University, Changsha 410083, China (e-mail: njuwangzhi@gmail.com).

H.-X. Li is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong (e-mail: mehxli@cityu.edu.hk).

C. Chen is with the Department of Control and Systems Engineering, Nanjing University, Nanjing 210093, China (e-mail: clchen@nju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2901897

discretized physical system. The task is to figure out the available m sampling locations so that the predefined measure can be optimized. That is, we need to choose the most informative m locations from the assumed n ones. The complexity of the optimal sensor placement is considered to be NP-hard [18]. It can be theoretically solved by a brute-force method that inspects every optional placement for the m sensor locations out of the $\binom{n}{m}$ combinations. Under the circumstance, the computational complexity is exponential owing to the combinatorial property, which makes these approaches unpractical even if the values of n and m are modest. There are mainly five types of approaches for the NP-hard optimal sensor placement problem: 1) proper orthogonal decomposition (POD)-based methods; 2) convex optimization methods; 3) greedy methods; 4) heuristics; and 5) machine learning techniques.

POD-based methods [19]–[22] focused on placing sensors using the spatial structure of the underlying phenomena, that is, the decomposed low-dimensional modes. The sensor locations could be heuristically arranged at the extrema of the POD modes in fluid community [19], [23], [24], or to be optimized by a guided search on the spatial subspace [17], [20]. The major drawback was that POD was sensitive to experimental settings, such as input signals, initial conditions, and the number of snapshots [1]. Moreover, using a finite number of POD mode amplitudes as state estimators was sometimes inaccurate, as the relevant aerodynamic properties were not always linearly related to the small subset of selected modes [25].

Convex optimization methods [26] were based on relaxing the Boolean constraints $\{0, 1\}^n$ that represented the sensor locations to the convex set $[0, 1]^n$. This relaxation was commonly not tight and heuristic tricks were often added to optimize the sensor locations. As there was no prior guarantee on the distance from the optimal placement, Joshi and Boyd [26] derived an online bound for the quality of the obtained placement based on the gap between the primal and the dual problem. Meantime, this approach required that the optimization objective and the feasible set were convex.

Greedy algorithms, leveraging the submodularity of the objective functions [27], [28], simplified the problem by using a series of optimal local steps instead of global ones. A greedy algorithm was proved to have polynomial complexity and result in a suboptimal solution that performed within a $(1 - [1/e])$ of the optimum [28]–[30]. It had been investigated in sensor placement applications, such as the fluids flow reconstruction [31] and the ocean modeling [19], [23]. This approach required the objective functions to be submodular in general. However, common criteria, such as A-, D-, and E-optimality, could not be addressed using the concept of submodularity [29].

Heuristic methods were valid alternatives to reduce the expensive cost of the brute-force search. Challenges regarding the NP-hard problems made heuristics the feasible alternative for various complex optimization problems in the real-world applications. Many kinds of heuristics had been applied to the sensor placement configuration, including subspace-based guided search [17], [20]; tabu search [32]; entropy-based heuristic [33]; clustering-based heuristic [34]; simulated annealing [35]; genetic algorithms (GAs) [36]; etc. Moreover, in the convex relaxation [26] approach, heuristics were also needed to help choose the sensor locations.

Machine learning techniques, underlying the field of artificial intelligence and the computer simulation of thinking, provide a brand new way to address sensor placement problems that are high-dimensional, complex, and full of uncertainties. Krause *et al.* [29] modeled the spatial phenomena as the Gaussian processes and developed a lazy learning scheme of greedy algorithms for choosing sensor locations. Kasper *et al.* [37] learned the constrained placement with a suitable linear estimator based on randomly generating the sensor locations. Semaan [25] employed a random forests algorithm to select the most important input variables as the optimal sensor locations, which circumvented the necessity for POD and for optimization. It relied on many sensor inputs and the selection of response functions during the training phase, curtailing its direct experimental implementation.

Reinforcement learning (RL) [38], [39] is a machine learning methodology that has been widely investigated in the areas of computational intelligence [40]–[45]. Recent breakthroughs of deep RL algorithms [46], [47] make RL state-of-the-art technology in the artificial intelligence community. Defined in terms of optimization of Markov decision processes (MDPs), RL theory addresses the problem that how an autonomous active agent learns the optimal policies while interacting with an initially unknown environment. The self-learning property from unknown environments makes RL a promising candidate for the optimization or control of real systems, including evolutionary computation [48], fuzzy control [49], quantum computation [44], computer architecture [50], etc. Nevertheless, to the best of our knowledge, the RL-based optimization for sensor placement problem has not been addressed yet.

In this paper, we develop an integrated RL-based optimal sensor placement method for spatiotemporal modeling of DPSs. First, a spatial objective function is proposed to evaluate the spatiotemporal modeling performance over the entire time and space domain. Second, the sensor placement configuration is formulated as an MDP with specified elements $\langle s, a, s', r\rangle$: sensor locations as the state s, change of only one sensor location as the action a, new sensor locations as the next state s', and the objective function of the new locations as the reward r. Finally, the optimization process is executed by the iteration of state value functions until converging to an optimal policy, that is, leading to optimal sensor locations.

The main advantages of applying RL for solving sensor placement problems lie in five aspects.

1) Defined in a mathematical MDP framework, RL algorithms can theoretically guarantee the convergence toward the global optimum and, hence, provide a more stable learning process.
2) RL algorithms learn optimal policies directly from interactions with the unknown DPS environment without a model. Compared to the analytical methods, the RL-based method can work with all kinds of objective functions, circumventing requirements of convexity or submodularity.

3) RL algorithms execute an intensified search by exploitation and a diversified search by exploration, which makes it an efficient method for various NP-hard problems like optimal sensor placement in this paper.
4) RL algorithms are naturally implemented in a fully incremental way based on immediate rewards obtained during task execution [38], [48], enabling online learning for broad practical applications.
5) When utilizing advanced techniques, such as value function approximation [51] and deep learning [46], RL algorithms can overcome the "curse of dimensionality," and has an increasing potential for solving extremely high-dimensional problems.

The experimental results implemented on a simulated catalytic rod benchmark and a real snap curing oven system are given to show the feasibility and efficiency of the proposed RL-based optimal sensor placement method.

The rest of this paper sequentially presents the formulation of the sensor placement problem for a given class of DPSs in Section II, the integrated RL-based optimal sensor placement method in Section III, the experiments in Section IV, and the conclusions in Section V.

II. Problem Description

In this paper, a general class of DPSs is considered, which can be represented by the following nonlinear PDE:

$$\frac{\partial y(x,t)}{\partial t} = \mathcal{L}\left(y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \ldots, \frac{\partial^{n_0} y}{\partial x^{n_0}}\right) + \bar{\boldsymbol{B}}(x)\boldsymbol{u}(t) \tag{1}$$

subject to the mixed-type boundary conditions

$$q\left(y, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \ldots, \frac{\partial^{n_0-1} y}{\partial x^{n_0-1}}\right)\Bigg|_{x=x_a \text{ or } x=x_b} = 0 \tag{2}$$

and the initial condition

$$y(x,0) = y_0(x) \tag{3}$$

where $t \in [0, \infty)$ is the temporal variable, $x \in [x_a, x_b] \subset \mathbb{R}$ is the spatial coordinate, $y(x,t) = [y(x_1,t), \ldots, y(x_n,t)]^T \in \mathbb{R}^n$ is the spatiotemporal output, and $\boldsymbol{u}(t) \in \mathbb{R}^p$ is the temporal input. $\mathcal{L} \in \mathbb{R}^n$ is a complex vector function which contains a nonlinear spatial differential operator of order n_0, $\bar{\boldsymbol{B}}(x)$ is a matrix function of appropriate dimensions which describes how the temporal inputs are distributed in spatial domains, q is a nonlinear vector function, and $y_0(x)$ is a smooth vector function referring to the initial output. A common approach to modeling the unknown nonlinear DPSs leads to the time-space separation framework based on Karhunen–Loève decomposition (KLD) [1], where the spatiotemporal output can be decoupled into a set of orthogonal spatial basis functions (BFs) with corresponding temporal coefficients that captures the most relevant dynamics of the system, as described in Appendix A.

Calculating the dominant BFs refers to an eigenvalue problem related to spatial integral operators. The quality of computed BFs is largely dependent on the precise information that is measured on the sampling spatial domain. Unfortunately, that information is only attainable from a limited number of possibly expensive sensors. This makes the optimal sensor placement essential for spatiotemporal modeling of complex DPSs in research and engineering practice.

Nevertheless, the sensor placement for unknown nonlinear DPSs is challenging due to the following factors: 1) strongly nonlinear spatiotemporal dynamics of the physical system; 2) the NP-hard complexity of the placement problem; 3) the lack of the accurate model of the system; and 4) no guarantee on convexity or submodularity of the objective function. In summary, a model-free and computationally efficient approach in consideration of the unknown nonlinear spatiotemporal dynamics is needed to optimize the sensor locations for modeling this type of DPS.

III. Reinforcement Learning-Based Optimal Sensor Placement

In this section, the integrated RL-based optimal sensor placement method is presented for spatiotemporal modeling. First, a spatial objective function is proposed for evaluating the spatiotemporal modeling performance based on the reduced-order subspace identified from time-space separation (Appendix A). Second, the sensor placement problem is mathematically formulated as an MDP with specified elements. Finally, based on the defined MDP, the sensor locations is optimized according to the spatial objective function after the convergence of a temporal difference (TD) learning algorithm.

A. Spatial Objective Function

A spatial objective function regarding sensor locations is derived for evaluating spatiotemporal modeling performance. By taking advantage of the time-space separation property, it is defined on the reduced-order subspace obtained by KLD, and aims at minimizing the reconstruction error over the entire time and space domain.

1) Reduced-Order Subspace: Assume that $\boldsymbol{Y} = \{\boldsymbol{y}_j\}_{j=1}^{l}$ is the spatiotemporal output data set of the given PDE system (1), where $\boldsymbol{y}_j = \boldsymbol{y}(\boldsymbol{x}, t_j)$ is a vector of the possible n spatial measurements at time j. By KLD, the subspace $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1(\boldsymbol{x}), \ldots, \boldsymbol{\varphi}_k(\boldsymbol{x})]$ is obtained as a set of k orthonormal and n-dimensional BFs, representative of the original system space with a minimum number k of degrees of freedom. The measurements $\boldsymbol{y}$ can be expressed in terms of $\boldsymbol{\Phi}$ as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{c} + \boldsymbol{\epsilon} \tag{4}$$

where $\boldsymbol{c} \in \mathbb{R}^k$ is the temporal modes in the low-dimensional space and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is zero-mean independent identically distributed (i.i.d.) Gaussian noise with variance σ^2 in each dimension.[1]

2) Measurements Projection: The operator $\boldsymbol{P}_m \in \mathbb{R}^{m \times n}$ is defined to project a vector of measurements $\boldsymbol{y} \in \mathbb{R}^n$ on its m subcoordinates in order to get the candidate measurements $\boldsymbol{y}_m = \boldsymbol{P}_m\boldsymbol{y}$ with reduced dimension. It projects the

[1] $\boldsymbol{\epsilon}$ consists of two parts $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2$. $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_1)$ is the i.i.d. Gaussian distributed truncation error on Karhunen-Loève reconstruction [52], [53], where $\boldsymbol{\Sigma}_1$ is a diagonal covariance matrix with each diagonal element being σ_1^2. $\boldsymbol{\epsilon}_2 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_2)$ is the i.i.d. Gaussian noise that perturbs the measurements [26], [53], with each diagonal element of $\boldsymbol{\Sigma}_1$ being σ_2^2. According to the property of Gaussian independent random variables, we have $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$, where each diagonal element of the covariance matrix is $\sigma^2 = \sigma_1^2 + \sigma_2^2$.

reduced-order plane in (4) onto its m coordinates, denoted as measurement subspace M_m, so that

$$\hat{\boldsymbol{y}}_m = \boldsymbol{P}_m \boldsymbol{\Phi} \hat{\boldsymbol{c}} + \boldsymbol{P}_m \boldsymbol{\epsilon} \tag{5}$$

where $\boldsymbol{P}_m \boldsymbol{\epsilon}$ is the projection error.

3) Least-Squares Problem: The modeling on space M_m leads to find the temporal coefficient vector $\hat{\boldsymbol{c}}$ that minimizes the distance between the real data $\boldsymbol{y}_m$ and its estimate $\hat{\boldsymbol{y}}_m$. It is equivalent to the minimum least-squares problem as follows:

$$\min_{\hat{\boldsymbol{c}}} \left(\boldsymbol{y}_m - \boldsymbol{P}_m \boldsymbol{\Phi} \hat{\boldsymbol{c}}\right)^T \left(\boldsymbol{y}_m - \boldsymbol{P}_m \boldsymbol{\Phi} \hat{\boldsymbol{c}}\right). \tag{6}$$

The maximum likelihood solution [53] of $\boldsymbol{c}$ is

$$\hat{\boldsymbol{c}} = \left(\boldsymbol{Q}_m^T \boldsymbol{Q}_m\right)^{-1} \boldsymbol{Q}_m^T \boldsymbol{y}_m \tag{7}$$

where $\boldsymbol{Q}_m = \boldsymbol{P}_m \boldsymbol{\Phi}$. The operator $\boldsymbol{P}_m$, which characterizes the available measurements, is assumed to span $\mathbb{R}^k$ [26]. The least-squares error is conditioned on the spectrum of $\boldsymbol{T}_m = \boldsymbol{Q}_m^T \boldsymbol{Q}_m \in \mathbb{R}^{k \times k}$, a Hermitian-symmetric matrix that determines the reconstruction performance.

4) Error Analysis: As shown in (4), the available measurements $\boldsymbol{y}_m$ are assumed to be perturbed by an i.i.d. Gaussian noise with zero mean and variance σ^2 in each dimension. The estimation error $\boldsymbol{c} - \hat{\boldsymbol{c}}$ has zero mean and covariance

$$\boldsymbol{\Sigma}_m = \sigma^2 \boldsymbol{T}_m^{-1}. \tag{8}$$

The η-confidence ellipsoid of the estimation error, that is, the minimum volume ellipsoid that contains $\boldsymbol{c} - \hat{\boldsymbol{c}}$ with probability η, is given by

$$\omega_\alpha = \left\{ z | z^T \boldsymbol{\Sigma}_m^{-1} z \leq \alpha \right\} \tag{9}$$

where $\alpha = F_{\chi_k^2}^{-1}(\eta)$ and $F_{\chi_k^2}$ is the cumulative distribution function of a χ-squared random variable with k degrees of freedom. The volume of the η-confidence ellipsoid can be a scalar measure on the quality of estimation error as follows:

$$\mathrm{vol}(\omega_\alpha) = \frac{(\alpha \pi)^{k/2}}{\Gamma\left(\frac{k}{2+1}\right)} \det\left(\boldsymbol{\Sigma}_m^{1/2}\right) \tag{10}$$

where Γ is the Gamma function. In practice, the log form is always adopted as

$$\log \mathrm{vol}(\omega_\alpha) = \beta - \left(\frac{1}{2}\right) \log \det(\boldsymbol{T}_m) \tag{11}$$

where β is a constant that depends only on σ, k, and η. The log volume of the confidence ellipsoid identifies a quantitative measure of how informative the collection of m measurements is.

5) Objective Function $\mathcal{F}$: A subset of m out of n candidate sensors should be selected to minimize the log volume of confidence ellipsoid. This can be transformed to the following optimization problem:

$$\underset{\boldsymbol{P}_m}{\text{maximize}}\, \mathcal{F}(\boldsymbol{P}_m) = \log \det(\boldsymbol{T}_m) = \log \det\left(\boldsymbol{\Phi}^T \boldsymbol{P}_m^T \boldsymbol{P}_m \boldsymbol{\Phi}\right) \tag{12}$$

where the spatial projection matrix $\boldsymbol{P}_m$ is the optimization variable, and we interpret $\log \det(\boldsymbol{T}_m)$ as $-\infty$ if $\boldsymbol{T}_m$ is singular.

TABLE I
SENSOR PLACEMENT CONFIGURATION

Formulation of MDP	
state s	$\boldsymbol{P}_m$
action a	$\boldsymbol{\Psi}$
next state s'	$\boldsymbol{P}'_m = \boldsymbol{P}_m \boldsymbol{\Psi}$
reward $r(s,a)$	$\mathcal{F}(\boldsymbol{P}'_m)$
state value function	$v_\pi(s)$

B. Formulation of Markov Decision Process

The basic concepts of MDP and RL are introduced in Appendix B. In the sensor placement configuration of the PDE system (1), the projection matrix $\boldsymbol{P}_m$ indicates that m sensors are to be selected among the candidate n measurements. Mathematically, $\boldsymbol{P}_m$ can be considered as the first m rows of an n-dimensional permutation matrix. The positions $i_1, \ldots, i_{j-1}, i_j, i_{j+1}, \ldots, i_m$ of element 1 in the rows of $\boldsymbol{P}_m$ encodes the m sensors to be used. Hence, $\boldsymbol{P}_m$ can be modeled as the state s of the RL configuration directly. An action a is defined as the change of only one sensor to be activated, which refers to a transformation matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$ such that $\boldsymbol{P}'_m = \boldsymbol{P}_m \boldsymbol{\Psi}$, where $\boldsymbol{P}'_m$ is the successor state s' with element 1 in positions $i_1, \ldots, i_{j-1}, i'_j, i_{j+1}, \ldots, i_m (i_j \neq i'_j)$. After executing action a in state s, the system transits to state s' with projection matrix $\boldsymbol{P}'_m$, and the one-step reward $r(s,a)$ is set as the corresponding objective function $\mathcal{F}(\boldsymbol{P}'_m)$. The value function, $v_\pi(s)$, is the expected cumulative return when starting in s under policy π thereafter, defined as

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{\tau=0}^{\infty} \gamma^\tau r_{t+\tau+1} | s_t = s \right] \tag{13}$$

where $\mathbb{E}_\pi[\,\cdot\,]$ denotes the expected value under the circumstance that the agent follows the given policy π. The value function indicates how good it is for the agent to be in a given state. With these preliminaries, the sensor placement configuration can be formulated as an MDP shown in Table I.

C. RL-Based Sensor Placement Method

Based on the proposed spatial objective function and the formulated MDP, an integrated RL algorithm for optimal sensor placement is shown in Algorithm 1.

The learning parameters are initialized in lines 1–5. The learning process consists of several episodes, and each episode consists of no more than $t_{\max}$ time steps in lines 6–22. At the beginning of each episode, the state of sensor locations is initialized arbitrarily in order to explore the entire state space. The episode is terminated early if the state s_t at time t results in an objective function $\mathcal{F}(s_t)$ that is greater than the recorded maximum $\mathcal{F}_{\max}$. This process aims at intensively searching the potential optimum within the local region of this learning episode.

For a state s, there are $m(n-m)$ available actions $a \in \mathcal{A}(s)$ and corresponding successor states $s' \in \mathcal{S}'$ with value functions $v(s')$. In order to explore the set of possible actions and exploit experiences from the reinforcement returns, the actions are selected using ε-greedy strategy for a tradeoff between exploration and exploitation. The greedy part of the action selection strategy will guide the sensors to be located in states

Algorithm 1: RL-Based Optimization for Sensor Locations

1 Initialize state value functions $v(s)$ ($\forall s \in \mathcal{S}$), and the policy $\pi{:}p^{\pi}(s_t, a_t)$ arbitrarily
2 Initialize the recorded maximum $\mathcal{F}$: $\mathcal{F}_{max} \leftarrow 0$
3 Initialize the number of recorded episodes: $N \leftarrow 0$
4 $N_{max} \leftarrow$ maximum episodes for indicating convergence
5 $t_{max} \leftarrow$ maximum time steps in each episode
6 **while** $N \leq N_{max}$ **do**
7 **while** $t \leq t_{max}$ *and* $\mathcal{F}(s_t) \leq \mathcal{F}_{max}$ **do**
8 Initialize $t = 1, s_t$
9 Select action a_t using the $\varepsilon-$greedy strategy
10 Take a_t, observe next state s_{t+1} and reward r_{t+1}
11 Temporal difference error $\delta_{t+1} = r_{t+1} + \gamma v(s_{t+1}) - v(s_t)$
12 Value iteration $v(s_t) \leftarrow v(s_t) + \alpha_t \delta_{t+1}$
13 $s_t \leftarrow s_{t+1}$
14 $\mathcal{F}(s_t) \leftarrow$ spatial objective function in (12)
15 **if** $\mathcal{F}(s_t) > \mathcal{F}_{max}$ **then**
16 $\mathcal{F}_{max} \leftarrow \mathcal{F}(s_t)$, $s^* \leftarrow s_t$
17 $N \leftarrow 0$
18 **else**
19 $N \leftarrow N + 1$
20 **end**
21 **end**
22 **end**
23 Obtain optimal sensor locations: $P_m^* \leftarrow s^*$

with greater objective functions, that is, exploitation. The other part of the random selection is applied for diversely searching over the whole space and avoiding the local optimum, i.e., exploration. Finally, if the recorded $\mathcal{F}_{\max}$ has not been promoted for a preset number of consecutive episodes, $N_{\max}$, then the state converges to the optimum s^* with maximal objective function $\mathcal{F}^*$ according to the convergence analysis in Remark 1, and the learning process stops. The optimal sensor placement $\boldsymbol{P}_m^*$ is obtained for providing the best performance of spatiotemporal modeling using the limited set of sensors in line 23.

Remark 1 (Convergence of Algorithm 1): Consider an RL-based sensor placement agent in a nondeterministic MDP, for every state s, the value $v_t(s)$ will converge to the optimal state value function $v^*(s)$ if the following constraints are satisfied [38], [39].

1) The rewards in the whole learning process satisfy $(\forall s, a)|r_s^a| \leq R$, where R is a finite constant value.
2) A discount factor $\gamma \in [0, 1)$ is adopted.
3) During the learning process, the non-negative learning rate α_t satisfies

$$\lim_{T\to\infty} \sum_{t=1}^{T} \alpha_t = \infty, \qquad \lim_{T\to\infty} \sum_{t=1}^{T} \alpha_t^2 < \infty. \tag{14}$$

IV. Simulation Experiments

To test the proposed RL-based method, a simulated catalytic rod and a practical snap curing oven system are studied. Let $c(z, t), z = 1, \ldots, k$ denote the real temporal modes of the reduced-order space in (4), and $\hat{c}(z, t)$ be the modes estimated with the limited number of sensors in (7). Let $y(x, t)$ denote the measured output of the system, and $\hat{y}(x, t)$ be the predicted output with the selected sensors. The performance indices for evaluating the quality of sensor locations are defined as the rooted mean square errors (RMSEs)

$$\begin{aligned} \mathrm{RMSE}(c, \hat{c}) &= \left(\int \Sigma \big(c(z, t) - \hat{c}(z, t)\big)^2 dz / \int dz \Sigma \Delta t \right)^{\frac{1}{2}} \\ \mathrm{RMSE}(y, \hat{y}) &= \left(\int \Sigma \big(y(x, t) - \hat{y}(x, t)\big)^2 dx / \int dx \Sigma \Delta t \right)^{\frac{1}{2}}. \end{aligned} \tag{15}$$

In our RL setting, the state s is the locations of the available m sensors out of the n candidate measurements. The action a is defined as the change of the location of only one sensor. After executing a in s, the system transits to new locations s', and the objective function $\mathcal{F}$ in (12) regarding s' is defined as the one step reward $r(s, a)$. In order to demonstrate the performance of sensor placement, the RL-based optimization method is compared to three classic methods: 1) convex optimization; 2) greedy method; and 3) GA. The parameter settings for the one-step value iteration are as follows: learning rate $\alpha = 0.1$ and discount factor $\gamma = 0.95$. For each group of experiments, ε-greedy exploration strategy is applied to investigate the performance of the proposed algorithm. The exploration rate ε is set as 1.0 at the very beginning and then gradually decreases to 0. All the algorithms are implemented with Python 3.6 running on Mac OS X with Intel Core i5-7360 2.30 GHz and 8-GB RAM, and all the experimental results presented in this paper are averaged over 100 runs.

A. Benchmark of Catalytic Rod

In this benchmark, a classical transport-reaction process in chemical industry [54] is considered, where the temperature distribution on a catalytic rod is modeled from a limited number of sensors. The mathematical model of the following parabolic PDE can be used to describe the rod temperature evolution over time and space as:

$$\begin{aligned} \frac{\partial y(x, t)}{\partial t} = \frac{\partial^2 y(x, t)}{\partial x^2} &+ \beta_T \left(e^{-\frac{\gamma}{1+y}} - e^{-\gamma} \right) \\ &+ \beta_u \big(\boldsymbol{b}^T(x) \boldsymbol{u}(t) - y(x, t) \big) \end{aligned} \tag{16}$$

subject to the Dirichlet boundary and initial conditions

$$y(0, t) = 0, \qquad y(\pi, t) = 0, \qquad y(x, 0) = y_0(x)$$

where $y(x, t)$ is the rod temperature, $\boldsymbol{u}(t)$ is the temporal input function, $\boldsymbol{b}(x)$ is the spatial distribution of input actuators, β_T is the heat of reaction, β_u is the heat transfer coefficient, and γ denotes the activation energy. The process parameters are usually set as

$$\beta_T = 50, \qquad \beta_u = 2, \qquad \gamma = 4.$$

There are four input actuators $\boldsymbol{u}(t) = [u_1(t), \ldots, u_4(t)]^T$, $u_i(t) = 1.1 + 5\sin(t/10 + i/10)$, $(i = 1, \ldots, 4)$, with the spatial distribution function $\boldsymbol{b}(x) = [b_1(x), \ldots, b_4(x)]^T$, $b_i(x) =$

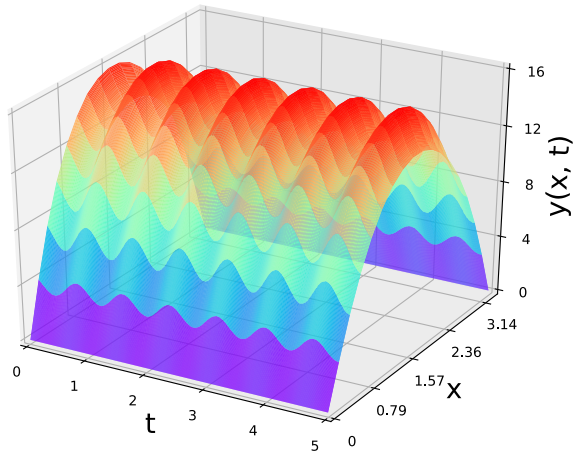


Fig. 1. Snapshots produced by direct numerical simulation of the nonlinear transport-reaction process.

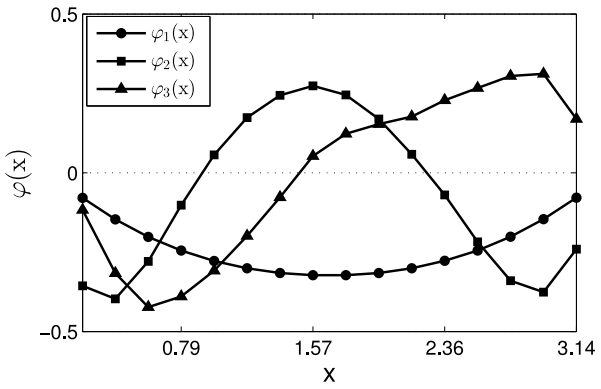


Fig. 2. First three BFs chosen in accordance with the energy criterion.

$H(x-(i-1)\pi/4)-H(x-i\pi/4)$, $(i=1,\ldots,4)$, and $H(\cdot)$ is the standard Heaviside function. The noise-free streaming data generated from (16) is sampled at time interval $\Delta t = 0.01$. The initial condition $y_0(x)$ is set to be the steady state with the input $u_i(t) = 1.1, i = 1,\ldots,4$.

The simulated snapshots used are depicted in Fig. 1. Each snapshot contains the temperature values from 16 equispaced positions sampled in the spatial domain at a given time. The first three dominant BFs which are capable of capturing more than 99% of the dominant dynamics of the system are depicted in Fig. 2. Based on the reduced-order subspace of the system, the sensor locations with a limited number m is optimized according to the derived spatial objective function using the RL-based method. The optimal sensor locations obtained by the proposed method are depicted in Fig. 3 for different sets of available measurements. Fig. 4 presents the maximum reward and the average reward acquired in sequential learning episodes for different sets of available sensors. It can be observed that the received rewards grow rapidly at the first learning episodes, indicating an intensified and efficient learning process of the RL-based sensor placement method.

In the next stage, three techniques are considered for comparison to the proposed method: 1) sensor selection via convex optimization [26]; 2) greedy method [30]; and 3) GA [36]. As an example, Fig. 5 shows the sensor arrangements for

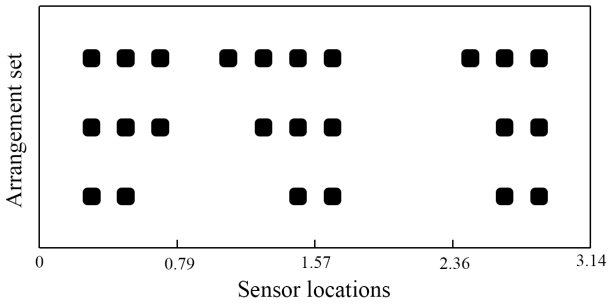


Fig. 3. Optimal sensor locations obtained by the RL-based method with a different number ($m = 6, 8,$ and10) of available sensors.

TABLE II
RMSEs WITH LIMITED SENSORS SELECTED BY DIFFERENT METHODS. BEST PERFORMANCES ARE MARKED IN BOLD

	Sensors	Convex	Greedy	GA	RL
RMSE$(c,\hat{c})$	m=6	3.51E-05	8.22E-05	5.96E-05	**3.24E-05**
	m=8	3.16E-05	6.68E-05	4.37E-05	**3.15E-05**
	m=10	2.95E-05	4.88E-05	3.30E-05	**2.52E-05**
RMSE$(y,\hat{y})$	m=6	4.40E-04	4.84E-04	4.86E-04	**4.12E-04**
	m=8	4.28E-04	4.68E-04	4.74E-04	**4.18E-04**
	m=10	4.15E-04	4.26E-04	4.12E-04	**4.09E-04**

eight available sensors attained by convex optimization, greedy method, GA, and RL, respectively. Table II presents the RMSEs in (15) with reduced sensor locations using the four methods. It can be observed that, compared to the other three methods, the optimal sensor locations obtained by the RL-based method provide a closer approximation to the original system.

In order to evaluate the estimation properties of the tested methods at a dynamic level, observation experiments are carried out with $m = 8$ available sensors selected by the four methods. The output of the system, in these experiments, is perturbed by the Gaussian white noise with zero mean and standard deviation $\sigma(x_i) = A_d(x_i)n_d$, where $A_d(x_i) = (\max(y(x_i,t))-\min(y(x_i,t)))/3$, $(i=1,\ldots,n)$ and $n_d = 0.05$. The evolution of the dominant three true c-states and the estimated ones is presented in Fig. 6. The corresponding RMSEs with reduced sensor locations using the four methods are shown in Table III. The capability of observing the dominant modes of the distributed process at a dynamic level is highly dependent on the type of sensor placement. The RL-based sensor placement method presents a superior observer performance compared to the alternative sensor selection methods like convex optimization, greedy method, or GA.

Further, we investigate a series of experiments with a different number of candidate n and available m sensors. The configuration is shown in Table IV. The size of the state-action space varies by an order of magnitude as the number of sensors increases. The performances of the GA- and RL-based placement methods are observed. For a fair comparison, the size of the population in one generation of GA is set to the same as the number of maximum steps in one episode of RL. Fig. 7 shows the obtained (normalized) recorded maximum objection function $\mathcal{F}_{\max}$ regarding learning episodes/generations under

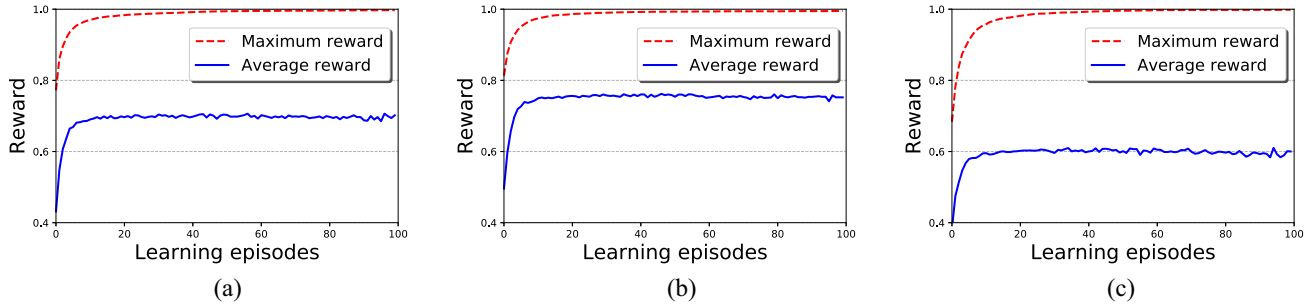


Fig. 4. RL performances for optimizing sensor locations with different sets of available sensors. (a) $m = 6$. (b) $m = 8$. (c) $m = 10$.

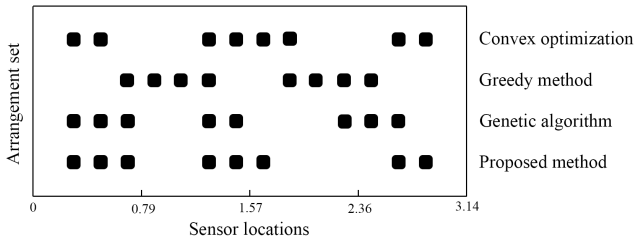


Fig. 5. Optimal sensor arrangements for $m = 8$ available sensors attained by convex optimization, greedy method, GA, and RL, respectively.

TABLE III
RMSEs WITH LIMITED SENSORS SELECTED BY DIFFERENT METHODS AT A DYNAMIC LEVEL. BEST PERFORMANCES ARE MARKED IN BOLD

	Sensors	Convex	Greedy	GA	RL
RMSE$(c, \hat{c})$	m=6	4.21E-03	6.12E-03	4.97E-03	**3.97E-03**
	m=8	3.43E-03	4.04E-03	4.50E-03	**3.01E-03**
	m=10	2.32E-03	3.56E-03	2.73E-03	**2.18E-03**
RMSE$(y, \hat{y})$	m=6	4.23E-02	4.40E-02	4.27E-02	**4.21E-02**
	m=8	4.01E-02	4.02E-02	4.08E-02	**3.96E-02**
	m=10	3.83E-02	3.93E-02	3.92E-02	**3.83E-02**

TABLE IV
CONFIGURATION OF THE SENSOR PLACEMENT PROBLEM UNDER DIFFERENT SCALES AND THE CORRESPONDING RUNNING TIME OF THE GA- AND RL-BASED METHODS

Sensors	# of states	# of actions	Running time (s)	
			GA	RL
n=12, m=6	924	36	0.36	0.13
n=16, m=8	12870	64	0.96	0.50
n=20, m=10	184756	100	4.70	5.85
n=24, m=12	2704156	1448	26.96	36.40

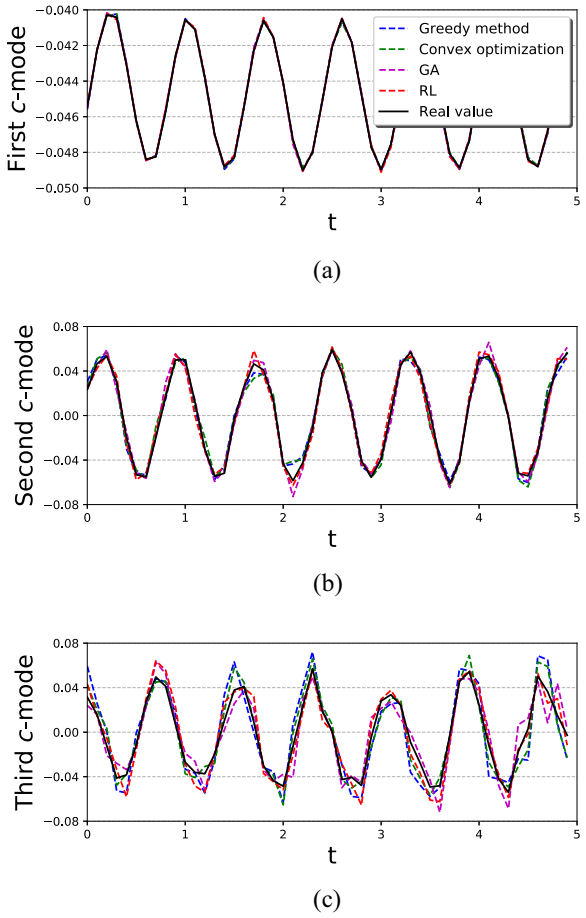


Fig. 6. Evolution of the real and the estimated temporal modes under temperature output noise with eight sensors selected by different methods. (a) First c-mode. (b) Second c-mode. (c) Third c-mode.

different scales of the state-action space. It can be observed that the RL-based method executes a faster and more efficient learning process toward the optimal sensor locations compared to the GA-based method. The corresponding running time is given in Table IV, showing that the RL-based method has a comparable computational cost with the GA-based method.

B. Application to Snap Curing Oven System

In this section, the proposed RL-based sensor placement algorithm is validated using a real-world implementation—a curing thermal process in the snap oven [55]. As shown in Fig. 8, the heaters embedded in the heating block are used to heat the integrated circuit on the lead frame. Nitrogen is filled inside the oven to avoid oxidation. The temperature in the oven changes over time and space due to the complex thermal process inside including radiation, convection, and conduction. As shown in Fig. 9, the snap oven has four heaters ($h1$–$h4$) for heating, and 16 candidate thermocouple sensors ($s1$–$s16$) on the lead frame used to measure the temperature. For better modeling and control, the limited number of sensors should be carefully selected to reflect the complex spatiotemporal dynamics of the system.

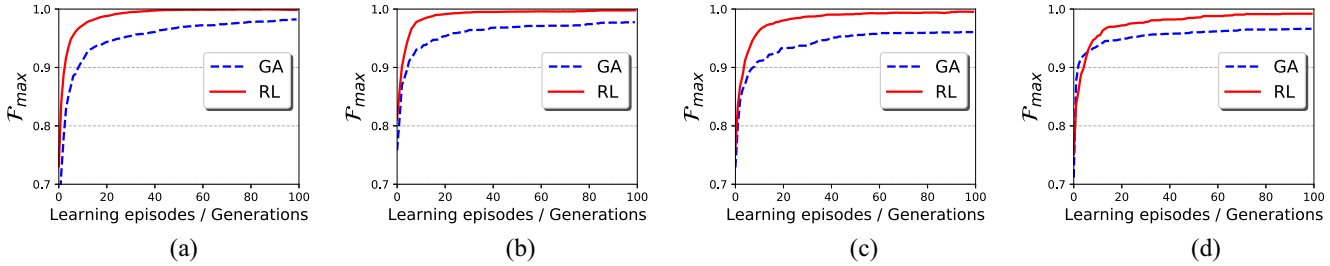


Fig. 7. Performances of the obtained maximum objective function of the GA- and RL-based sensor placement methods under different scales. (a) n =12, m =6. (b) n =16, m =8. (c) n =20, m =10. (d) n =24, m =12.

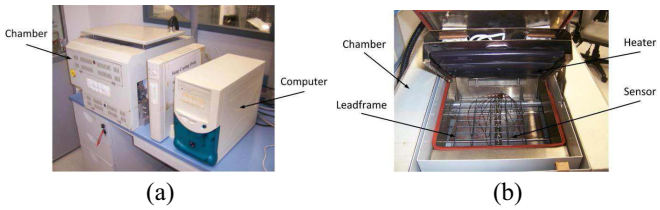


Fig. 8. Snap curing oven system. Curing (a) thermal system and (b) oven.

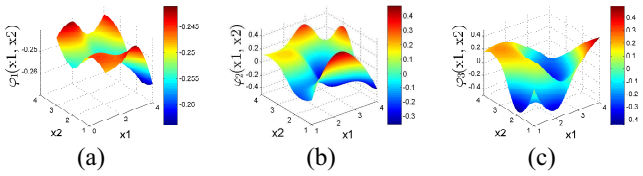


Fig. 11. First three 2-D BFs of the snap curing oven system. (a) $\varphi_1(x1, x2)$. (b) $\varphi_2(x1, x2)$. (c) $\varphi_3(x1, x2)$.

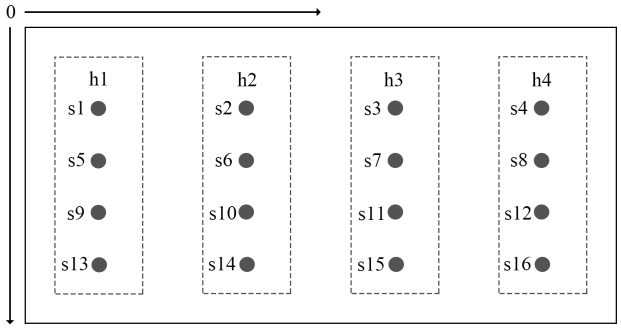


Fig. 9. Available sensor locations for modeling.

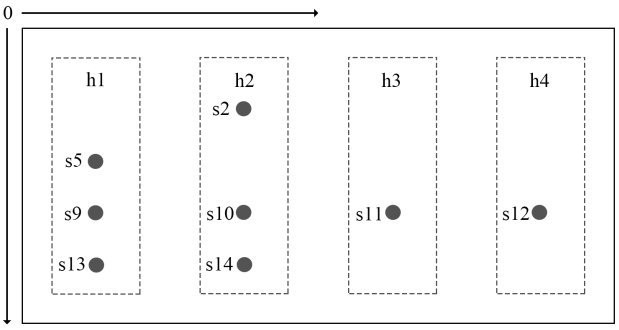


Fig. 12. Optimal placement for eight available sensors obtained by the RL-based method.

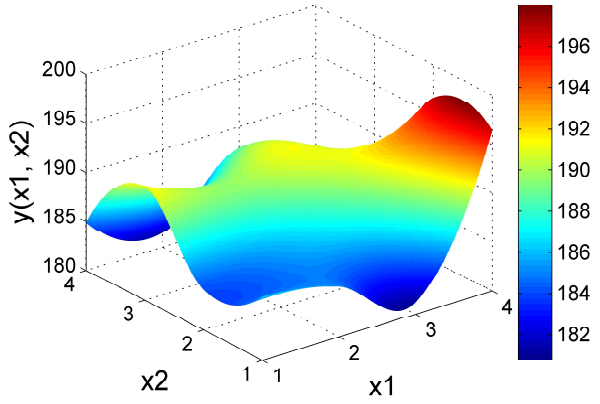


Fig. 10. Temperature distribution at time 1000 in the curing oven.

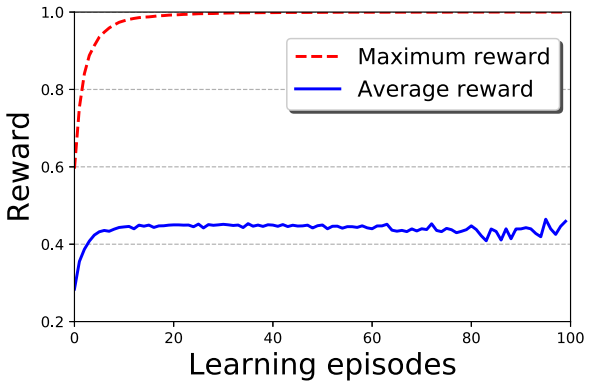


Fig. 13. Learning performance for placing eight available sensors by the RL-based method.

In this experiment, 3000 temperature data are collected with a sampling interval of Δt = 10 s. As an example, the real snapshots at time 1000 are depicted in Fig. 10. Based on the dominant 2-D spatial BFs as shown in Fig. 11, the acquired optimal placement for eight available sensors using the proposed RL-based method and the corresponding learning performance are shown in Figs. 12 and 13, respectively. It can be observed that the RL-based method can obtain a near-optimal maximal reward with only a few learning episodes.

The RL-based sensor placement method is further compared to convex optimization, greedy method, and GA with a different number of available sensors (m = 4, 6, 8, 10, and 12) at both static and dynamic levels. At the dynamic level,

TABLE V
RMSE($c, \hat{c}$) AT BOTH STATIC AND DYNAMIC LEVELS WITH LIMITED SENSORS SELECTED BY DIFFERENT METHODS IN THE SNAP CURING OVEN. BEST PERFORMANCES ARE MARKED IN BOLD

Static level / Dynamic level (E-03)				
Sensors	Convex	Greedy	GA	RL
m = 4	**3.3 / 13.2**	11.3 / 64.5	3.5 / 47.5	**3.3 / 13.2**
m = 6	**3.0 / 8.6**	10.1 / 38.5	5.0 / 40.2	**3.0 / 8.6**
m = 8	2.7 / 7.4	8.7 / 27.4	2.4 / 38.5	**2.2 / 6.4**
m = 10	2.3 / 5.8	3.0 / 20.0	2.8 / 33.0	**2.2 / 4.3**
m = 12	**0.8 / 2.8**	2.3 / 19.0	1.3 / 31.9	**0.8 / 2.8**

the output data of the snap curing oven is perturbed by the Gaussian white noise with mean zero and standard deviation as described in the above simulated experiment. The RMSEs between the real c-modes and the estimated ones using these four methods are shown in Table V. It is confirmed that, in the real oven case, the RL-based method also provides a better sensor placement scheme for reconstructing and observing the system's dynamics.

V. CONCLUSION

In this paper, we presented an integrated RL-based optimal sensor placement method for spatiotemporal modeling of DPSs. The sensor placement problem for unknown DPSs is mathematically formulated as an MDP with the proposed spatial objective function. The promising properties of RL enable the sensor placement algorithm to be implemented in an online, model-free, and fully incremental way. The experimental results on a simulated catalytic rod and a practical snap curing oven system have demonstrated that the proposed method provides better sensor locations for reconstructing and observing the system's dynamics. Under the same circumstances, the RL-based method executes a more efficient learning process and has a comparable computational cost compared to the GA-based method. Our future work will focus on more efficient (deep) RL algorithms for more complex sensor placement problems.

APPENDIX A
TIME-SPACE SEPARATION

For time-space separation of the PDE system (1) [54], [56]–[58], KLD [52] is widely utilized for calculating the empirical eigenfunctions and deriving accurate reduced-order approximations. In practice, assume the system output variable $\{y(x_i, t)\}_{i=1,t=1}^{n,l}$, denoted as *snapshot*, is uniformly sampled in both the time and space coordinates, where $x \in \Omega$ is the spatial variable, Ω is the spatial domain, and t is the time variable. Define the norm, inner product, and ensemble average as $||f(x)|| = (f(x), f(x))^{1/2}$, $(f(x), g(x)) = \int_\Omega f(x)g(x)dx$, and $\langle f(x,t)\rangle = (1/l)\sum_{t=1}^{l} f(x,t)$. Inspired by the Fourier theory, the spatiotemporal variable $y(x,t)$ can be truncated into a dominant number k of orthonormal spatial BFs $\{\varphi_i(x)\}_{i=1}^{k}$ with corresponding temporal coefficients $\{c_i(t)\}_{i=1}^{k}$

$$y_k(x,t) = \sum_{i=1}^{k} \varphi_i(x)c_i(t) \tag{17}$$

where $y_k(x,t)$ denotes the kth-order approximation. The temporal coefficients can be computed from

$$c_i(t) = (\varphi_i(x), y(x,t)), i = 1, \ldots, k. \tag{18}$$

Time-space separation aims to compute the most dominant spatial BFs $\{\varphi_i(x)\}_{i=1}^{k}$ among the spatiotemporal output $\{y(x_i,t)\}_{i=1,t=1}^{n,l}$ using KLD. Finding the typical $\{\varphi_i(x)\}_{i=1}^{k}$ can be achieved by minimizing the corresponding Lagrangian function

$$J = \langle ||y(x,t) - y_k(x,t)||^2\rangle + \sum_{i=1}^{k} \lambda_i((\varphi_i, \varphi_i) - 1) \tag{19}$$

corresponding to constraints $(\varphi_i, \varphi_i) = 1, \varphi_i \in L^2(\Omega), i = 1, \ldots, k$, the necessary condition of the solution can be expressed as

$$\int_\Omega R(x,\zeta)\varphi_i(\zeta)d\zeta = \lambda_i\varphi_i(x), (\varphi_i, \varphi_i) = 1, i = 1, \ldots, n \tag{20}$$

where $R(x,\zeta) = \langle y(x,t)y(\zeta,t)\rangle$ is the spatial two-point correlation function.

The solution of (20) can be achieved by a computationally efficient method of snapshots [52]. The eigenfunction (spatial BFs) $\varphi_i(x)$ can be transformed into a linear combination of the snapshots as

$$\varphi_i(x) = \sum_{t=1}^{l} \gamma_{it} y(x,t). \tag{21}$$

After substituting (21) into (20), the necessary condition is computed as

$$\int_\Omega \frac{1}{l}\sum_{t=1}^{l} y(x,t)y(\zeta,t)\sum_{k=1}^{l}\gamma_{ik}y(\zeta,k)d\zeta = \lambda_i\sum_{t=1}^{l}\gamma_{it}y(x,t). \tag{22}$$

Then, this eigenvalue problem is transformed to a simplified form of an $l \times l$ matrix eigen-decomposition problem as

$$\boldsymbol{C}\boldsymbol{\gamma}_i = \lambda_i\boldsymbol{\gamma}_i \tag{23}$$

where $\boldsymbol{\gamma}_i = [\gamma_{i1}, \ldots, \gamma_{il}]^T$ is the ith eigenvector, and

$$C_{tk} = \frac{1}{l}\int_\Omega y(\zeta,t)y(\zeta,k)d\zeta \tag{24}$$

is defined as the temporal two-point correlation function. The solution of problem (23) yields the eigenvectors $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_l$, which in turn can be used for constructing the eigenfunctions $\varphi_1(x), \ldots, \varphi_l(x)$ in (21). Because the matrix C is symmetric and positive semidefinite, the derived eigenfunctions are orthogonal. The dominant k spatial BFs $\{\varphi_i(x)\}_{i=1}^{k}$ are selected in the descending order of the magnitude of the corresponding eigenvalues, which can capture more than 99% of the system *energy* according to

$$E_k = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{j=1}^{l}\lambda_j}. \tag{25}$$

APPENDIX B
REINFORCEMENT LEARNING

Standard RL theories are based on the concept of MDP. An MDP is denoted as a tuple $\langle S, A, R, P\rangle$, where S is the state space, A is the action space, R is the reward function, and P is the state transition probability. The policy of MDP is defined as a function $\pi : S \to \Pr(A)$, that is, a probability distribution in the state-action space. The objective is to estimate the optimal policy π^* that satisfies

$$J_{\pi^*} = \max_{\pi} J_{\pi} = \max_{\pi} E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{26}$$

where $\gamma \in [0, 1)$ is the discount factor, r_t is the reward at time-step t, $E_{\pi}[\,\cdot\,]$ stands for the expectation under policy π, and J_{π} is the expected cumulative reward.

When the agent has no prior knowledge of the initially unknown environment, TD learning (a widely used RL algorithm) can achieve optimal policies from delayed rewards. At a certain time step t, the agent observes the state s_t and, then, chooses an action a_t according to ε-greedy strategy

$$p(s, a) = \begin{cases} 1 - \varepsilon + \varepsilon/N_a, & \text{if } a = \arg\max_{a_i \in A(s), s_i' \in S'} v(s_i') \\ \varepsilon/N_a, & \text{otherwise} \end{cases} \tag{27}$$

where $N_a = |A(s)|$ denotes the cardinality of the action set. After executing action a_t, the agent gets into the next state s_{t+1} and receives a reward r_{t+1} (a reflection of how good that action is in a short-term sense). Then, the agent will choose the next action a_{t+1} according to the best known knowledge.

The TD error is defined by

$$\delta_{t+1} = r_{t+1} + \gamma v(s_{t+1}) - v(s_t) \tag{28}$$

where r_{t+1} is the reinforcement signal received at time $t + 1$ and γ is the discount factor used to determine the present value of future rewards. The iteration rule of value functions is given by

$$v(s_t) \leftarrow v(s_t) + \alpha_t \delta_{t+1} \tag{29}$$

where α_t is the learning rate. After the algorithm converges, the optimal value functions under an optimal policy satisfy the Bellman equation

$$v_*(s) = \max_{a \in \mathcal{A}(s)} \sum_{s'} p(s'|s, a)\big[r(s, a, s') + \gamma v_*(s')\big]. \tag{30}$$

More details about TD learning can be found in [38].

REFERENCES

[1] H.-X. Li and C. Qi, "Modeling of distributed parameter systems for applications—A synthesized review from time-space separation," *J. Process Control*, vol. 20, no. 8, pp. 891–901, 2010.

[2] S. Kumar and J. H. Seinfeld, "Optimal location of measurements for distributed parameter estimation," *IEEE Trans. Autom. Control*, vol. AC-23, no. 4, pp. 690–698, Aug. 1978.

[3] L. C. Windes, A. Cinar, and W. H. Ray, "Dynamic estimation of temperature and concentration profiles in packed bed reactors," *Chem. Eng. Sci.*, vol. 44, no. 10, pp. 2087–2106, 1989.

[4] S. Omatu, S. Koide, and T. Soeda, "Optimal sensor location for a linear distributed parameter system," *IEEE Trans. Autom. Control*, vol. AC-23, no. 4, pp. 665–673, Aug. 1978.

[5] S. B. Jorgensen, L. Goldschmidt, and K. Clement, "A sensor location procedure for chemical processes," *Comput. Chem. Eng.*, vol. 8, nos. 3–4, pp. 195–204, 1984.

[6] M. Morari and M. J. O'Dowd, "Optimal sensor location in the presence of nonstationary noise," *Automatica*, vol. 16, no. 5, pp. 463–480, 1980.

[7] J. Romagnoli, J. Alvarez, and G. Stephanopolus, "Variable measurement structures for process control," *Int. J. Control*, vol. 33, no. 2, pp. 269–289, 1981.

[8] A. K. Singh and J. Hahn, "Determining optimal sensor locations for state and parameter estimation for stable nonlinear systems," *Ind. Eng. Chem. Res.*, vol. 44, no. 15, pp. 5645–5659, 2005.

[9] P. C. Müller and H. I. Weber, "Analysis and optimization of certain quantities of controllability and observability for linear dynamic system," *Automatica*, vol. 8, no. 3, pp. 237–246, 1972.

[10] D. Dochain, N. Tali-Maamar, and J. P. Babary, "On modeling, monitoring and control of fixed bed bioreactors," *Comput. Chem. Eng.*, vol. 21, no. 11, pp. 1255–1266, 1997.

[11] F. W. J. van den Berg, H. C. J. Hoefsloot, H. F. M. Boelens, and A. K. Smilde, "Selection of optimal sensor position in a tubular reactor using robust degree of observability criteria," *Chem. Eng. Sci.*, vol. 55, no. 4, pp. 827–837, 2000.

[12] A. V. Wouwer, N. Point, S. Porteman, and M. Remy, "An approach to the selection of optimal sensor locations in distributed parameter systems," *J. Process Control*, vol. 10, no. 4, pp. 291–300, 2000.

[13] Z. H. Qureshi, T. S. Ng, and G. C. Goodwin, "Optimum experimental design for identification of distributed parameter systems," *Int. J. Control*, vol. 31, no. 1, pp. 21–29, 1980.

[14] Y. Ali and S. Narasimhan, "Sensor network design for maximizing reliability of linear processes," *AIChE J.*, vol. 39, no. 5, pp. 820–828, 1993.

[15] D. J. Chmielewski, T. Palmer, and V. Manousiouthakis, "On the theory of optimal sensor placement," *AIChE J.*, vol. 48, no. 5, pp. 1001–1012, 2002.

[16] K. R. Muske and C. Georgakis, "Optimal measurement system design for chemical processes," *AIChE J.*, vol. 49, no. 6, pp. 1488–1494, 2003.

[17] A. A. Alonso, C. E. Frouzakis, and I. G. Kevrekidis, "Optimal sensor placement for state reconstruction of distributed process systems," *AIChE J.*, vol. 50, no. 7, pp. 1438–1452, 2004.

[18] F. Bian, D. Kempe, and R. Govindan, "Utility based sensor selection," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, Nashville, TN, USA, Apr. 2006, pp. 11–18.

[19] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis, "Efficient sensor placement for ocean measurements using low-dimensional concepts," *Ocean Model.*, vol. 27, nos. 3–4, pp. 160–173, 2009.

[20] A. A. Alonso, I. G. Kevrekidis, J. R. Banga, and C. E. Frouzakis, "Optimal sensor location and reduced order observer design for distributed process systems," *Comput. Chem. Eng.*, vol. 28, nos. 1–2, pp. 27–35, 2004.

[21] P. Wolf, S. Moura, and M. Krstic, "On optimizing sensor placement for spatio-temporal temperature estimation in large battery packs," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 973–978. doi: 10.1109/CDC.2012.6426191.

[22] P. Kumar, Y. M. E. Sayed, and R. Semaan, "Optimized sensor placement using stochastic estimation for a flow over a 2D airfoil with Coanda blowing," in *Proc. 7th AIAA Flow Control Conf.*, Atlanta, GA, USA, 2014, pp. 1–11.

[23] Z. Zhang, X. Yang, and G. Lin, "POD-based constrained sensor placement and field reconstruction from noisy wind measurements: A perturbation study," *Mathematics*, vol. 4, no. 2, p. 26, 2016.

[24] P. Mokhasi and D. Rempfer, "Optimized sensor placement for urban flow measurement," *Phys. Fluids*, vol. 16, no. 5, pp. 1758–1764, 2004.

[25] R. Semaan, "Optimal sensor placement using machine learning," *Comput. Fluids*, vol. 159, pp. 167–176, Dec. 2017.

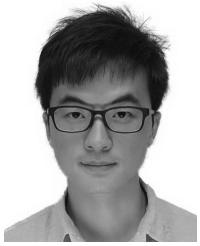
[26] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[27] G. L. Nemhauser and L. A. Wolsey, "Best algorithms for approximating the maximum of a submodular set function," *Math. Oper. Res.*, vol. 3, no. 3, pp. 177–188, 1978.

[28] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, Washington, CA, USA, Jun./Jul. 2011, pp. 1057–1064.

[29] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, Jan. 2008.

[30] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1135–1146, Mar. 2014.
[31] K. Willcox, "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Comput. Fluids*, vol. 35, no. 2, pp. 208–226, 2006.
[32] S. Lau, R. Eichardt, L. D. Rienzo, and J. Haueisen, "Tabu search optimization of magnetic sensor systems for magnetocardiography," *IEEE Trans. Magn.*, vol. 44, no. 8, pp. 1442–1445, Jun. 2008.
[33] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Berkeley, CA, USA, Apr. 2004, pp. 36–45.
[34] R. Mukherjee and S. O. Memik, "Systematic temperature sensor allocation and placement for microprocessors," in *Proc. 43rd Annu. Design Autom. Conf.*, San Francisco, CA, USA, Jul. 2006, pp. 542–547.
[35] F. Y. S. Lin and P. L. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage/discrimination in sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 1, pp. 43–45, Jan. 2005.
[36] W. Liu, W.-C. Gao, Y. Sun, and M.-J. Xu, "Optimal sensor placement for spatial lattice structure based on genetic algorithms," *J. Sound Vib.*, vol. 317, nos. 1–2, pp. 175–189, 2008.
[37] K. Kasper, L. Mathelin, and H. Abou-Kandil, "A machine learning approach for constrained sensor placement," in *Proc. Amer. Control Conf. (ACC)*, Chicago, IL, USA, Jul. 2015, pp. 4479–4484. doi: 10.1109/ACC.2015.7172034.
[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[39] C. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[40] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, "Reinforcement learning for constrained energy trading games with incomplete information," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3404–3416, Oct. 2017.
[41] Z. Zhang, D. Zhao, J. Gao, D. Wang, and Y. Dai, "FMRQ—A multiagent reinforcement learning algorithm for fully cooperative tasks," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1367–1379, Jun. 2017.
[42] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2216–2226, Jun. 2018.
[43] C. Chen, D. Dong, H.-X. Li, J. Chu, and T.-J. Tarn, "Fidelity-based probabilistic *Q*-learning for control of quantum systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 920–933, May 2014.
[44] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1207–1220, Oct. 2008.
[45] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Trans. Mechatronics*, to be published. doi: 10.1109/TMECH.2019.2899365.
[46] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[47] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.
[48] S. Mabu, K. Hirasawa, and J. Hu, "A graph-based evolutionary algorithm: Genetic network programming (GNP) and its extension using reinforcement learning," *Evol. Comput.*, vol. 15, no. 3, pp. 369–398, 2007.
[49] C.-F. Juang, J.-Y. Lin, and C.-T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 2, pp. 290–302, Apr. 2000.
[50] E. Ipek, O. Mutlu, J. F. Martínez, and R. Caruana, "Self-optimizing memory controllers: A reinforcement learning approach," *ACM SIGARCH Comput. Archit. News*, vol. 36, no. 3, pp. 39–50, 2008.
[51] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, no. 6, pp. 1107–1149, 2003.
[52] L. Sirovich, *New Perspectives in Turbulence*, 1st ed. New York, NY, USA: Springer-Verlag, 1991.
[53] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[54] Z. Wang and H.-X. Li, "Incremental spatiotemporal learning for online modeling of distributed parameter systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published. doi: 10.1109/TSMC.2018.2810447.
[55] H. Deng, H.-X. Li, and G. Chen, "Spectral-approximation-based intelligent modeling for distributed thermal processes," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 5, pp. 686–700, Sep. 2005.
[56] Y. Feng and H.-X. Li, "Detection and spatial identification of fault for parabolic distributed parameter systems," *IEEE Trans. Ind. Electron.*, to be published. doi: 10.1109/TIE.2018.2877188.
[57] B.-C. Wang and H.-X. Li, "A sliding window based dynamic spatiotemporal modeling for distributed parameter systems with time-dependent boundary conditions," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2018.2859444.
[58] Y. Feng and H.-X. Li, "Dynamic spatial independent component analysis based abnormality localization for distributed parameter systems," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2019.2900226.



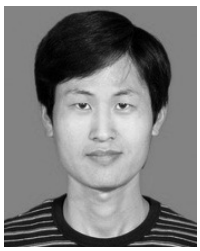
Zhi Wang received the B.E. degree in automation from the Department of Control and Systems Engineering, Nanjing University, Nanjing, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong.

His current research interests include reinforcement learning, system modeling, machine learning, and robotics.

Han-Xiong Li (S'94–M'97–SM'00–F'11) received the B.E. degree in aerospace engineering from the National University of Defense Technology, Changsha, China, in 1982, the M.E. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1991, and the Ph.D. degree in electrical engineering from the University of Auckland, Auckland, New Zealand, in 1997.

He is a Professor with the Department of SEEM, City University of Hong Kong, Hong Kong. He has a broad experience in both academia and industry. He has authored 2 books and about 20 patents, and published over 200 SCI journal papers with an *H*-index of 42 (Web of Science). His current research interests include process modeling and control, system intelligence, distributed parameter systems, and battery management system.

Dr. Li was a recipient of the Distinguished Young Scholar (overseas) by the China National Science Foundation in 2004, a Chang Jiang Professorship by the Ministry of Education, China, in 2006, and a National Professorship in China Thousand Talents Program in 2010. He serves as an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. He was an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS from 2002 to 2016 and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS from 2009 to 2015. He serves as a Distinguished Expert for Hunan Government and the China Federation of Returned Overseas Chinese.



Chunlin Chen (S'05–M'06) received the B.E. degree in automatic control and the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

He is currently a Professor and the Head of the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing, China. He was with the Department of Chemistry, Princeton University, Princeton, NJ, USA, from 2012 to 2013. He had visiting positions with the University of New South Wales, Kensington, NSW, Australia, and the City University of Hong Kong, Hong Kong. His current research interests include machine learning, intelligent control, and quantum control.

Dr. Chen is the Co-Chair of the Technical Committee on Quantum Cybernetics, IEEE Systems, Man and Cybernetics Society.