Moment-Based Reinforcement Learning for Ensemble Control

Yao-Chi Yu[1], Vignesh Narayanan[2] and Jr-Shin Li[1]

***Abstract*—Problems involving controlling the collective behavior of a population of structurally similar dynamical systems, the so-called *ensemble control*, arise in diverse emerging applications and pose a grand challenge in systems science and control engineering. Owing to the severely under-actuated nature and the difficulty of placing large-scale sensor networks, ensemble systems are limited to be actuated and monitored at the population-level. Moreover, mathematical models describing the dynamics of ensemble systems are often elusive. Therefore, it is essential to design broadcast controls that excite the entire population in such a way that the heterogeneity in system dynamics are robustly compensated. In this paper, we propose a reinforcement learning-based data-driven control framework incorporating population-level aggregated measurement data to learn a global control signal for steering a dynamic population in the desired manner. In particular, we introduce the notion of ensemble moments induced by aggregated measurements and derive the associated moment system to the original ensemble system. Then, using the moment system, we learn an approximation of optimal value functions and the associated policies in terms of ensemble moments through reinforcement learning. We illustrate the feasibility and scalability of the proposed moment-based approach via numerical experiments using a population of linear, bilinear, and nonlinear dynamic ensemble systems.**

I. Introduction

Large populations of dynamical systems are ubiquitous in many emerging applications across diverse scientific domains, including quantum control systems, neural networks, robotics, and cellular systems [1]–[6]. For instance, electromagnetic pulses are used to excite quantum ensembles in nuclear magnetic resonance spectroscopy and imaging [1], [3], a population of circadian cells are manipulated by the varying light intensities throughout the day to trigger biological clocks [2], and a population of neurons is actuated using neurostimulation to induce desired spiking activity in neuronal networks [5].

The task of precisely steering such populations of dynamical systems or the *ensemble control* problem has received great attention in the past two decades [1], [7]–[11]. In particular, investigations on the fundamental properties of ensemble controllability [12]–[18] and observability [19], [20] have been extensively reported in the literature. Furthermore, using the theoretical tools developed for the analysis of ensemble controllability, computational methods for synthesizing open-loop and optimal control signals to steer an ensemble system have also been investigated [21]–[28]. Primarily, these control synthesis methods rely on the availability of an accurate model describing the ensemble system dynamics, and focus only on a class of linear and bilinear ensembles.

Traditionally, adaptive and learning-based feedback control approaches have been employed to synthesize controls in the presence of model uncertainties by leveraging measurement data [29]–[31]. However, in many practical applications, the feedback information accessible from ensemble systems is spatio-temporally sparse either due to the lack of sophisticated sensing mechanism (e.g., neural systems) or it is infeasible to directly monitor these systems to begin with (e.g., quantum systems). In fact, in many emerging applications such as neuronal networks [32], robot swarms [33], and cellular oscillators [26], the ensemble systems are equipped with the infrastructure to only provide population-level aggregated measurements. In other words, all the subsystems in the population cannot be individually measured, and only an aggregation of measurements at the population level is available at each sampling instant. The existing ensemble control methods fail to exploit this information, although incomplete, to close the feedback-loops in an ensemble control systems to synthesize data-driven closed-loop policies.

To fill this gap, in this paper, we develop a data-driven moment-based control technique that adopts the reinforcement learning (RL) framework for synthesizing feedback control policies. In our approach, we employ the population-level feedback to compute *moment sequences* and utilize them to define the rewards and objectives for the RL algorithm. We demonstrate that the proposed strategy not only facilitates a feasible learning framework for synthesizing controls for ensemble systems but also is scalable (in terms of number of systems in the ensemble), making it data-efficient. Specifically, we demonstrate the efficacy of this approach using three case studies involving an ensemble system with (1) linear; (2) bilinear; and (3) input-affine nonlinear dynamic units.

The major contributions of this work include the (a) systematic RL formulation to study ensemble control problems; (b) design of data-driven moment-based RL algorithm to synthesize ensemble control policies; and (c) comprehensive analysis to verify the efficacy and scalability of the proposed RL strategy. To the best of our knowledge, in this work, we establish the first RL-based ensemble control design algorithm using aggregated measurement data and close the feedback loops of ensemble control systems.

The remainder of this paper is organized as follows. In

*This work was supported in part by National Science Foundation Awards CMMI-1763070, CMMI-1933976, and ECCS-1810202, National Institutes of Health Grant R01GM131403-01, and Air Force Office of Scientific Research Grant FA9550-21-1-0335.

[1]Yao-Chi Yu and Jr-Shin Li are with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA. Email:y.yu@wustl.edu, jsli@wustl.edu.

[2]Vignesh Narayanan is with the AI Institute of University of South Carolina, Columbia, SC, USA. Email:vignar@sc.edu.

Section II, we briefly review some related works in RL, formally introduce ensemble control systems, and present the ensemble control problem using aggregated measurements. In Section III, we define the notion of ensemble moments, moment systems, and the optimal control problem in terms of the moment system. In Section IV, we present our learning strategy to design feedback control using the moment system. Furthermore, we report the results of applying the proposed method to three cases of numerical experiments in Section V. In Section VI, we present detailed discussions on performance considerations, such as scalability and data-sparsity.

II. Background and problem formulation

A. Ensemble systems and control

An ensemble system is a parameterized family of dynamical systems, where the differences in the dynamics of individual units in the ensemble are captured by a dispersion parameter. Formally, the dynamics of an inhomogeneous ensemble is given by

$$\frac{d}{dt}x(t,\beta) = F(\beta, x(t,\beta), u(t)), \quad x(t_0,\beta) = x_0(\beta), \tag{1}$$

where F is the nonlinear dynamics of the ensemble, and $\beta \in K \subset \mathbb{R}^d$ is the dispersion parameter characterizing the variations in the subsystems within the ensemble. Typically, K is assumed to be a compact set [1], and hence the state-space of the ensemble system in (1) is a space of $\mathbb{R}^n$-valued functions defined on K, denoted by $\mathcal{F}(K)$. The state, or profile, of the ensemble system at time t is denoted by $x(t,\beta)$, so $x(t,\cdot) \in \mathcal{F}(K)$, and $u(t) \in \mathbb{R}^l$ is independent of β, which is broadcast to every individual system in the ensemble to steer the entire population to a desired target state.

B. RL and optimal control for a single dynamical system

Here, we provide a brief background on RL and its role in learning an optimal control policy for uncertain nonlinear systems. In particular, a (single) dynamical system can be represented as

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) \in \mathbf{X}, \tag{2}$$

where $\mathbf{X}$ is the set of states associated with the system in (2). We define $\mathbf{U}$ as the sets of feasible controls and $\mathcal{M}$ as the set of all functions $\mu : \mathbf{X} \to \mathbf{U}$ that maps $x(t) \in \mathbf{X}$ to $u(t) \in \mathbf{U}$, where μ denotes the feedback control. We further define the set of all admissible feedback policies as Π, wherein, a feedback policy $\pi \in \Pi$ is admissible if it stabilizes the system and results in a finite cost (3) [34]. The control inputs are selected based on policies that are designed to meet a desired objective. To quantify the performance of the system, we define the performance output associated with a policy μ as

$$J_\mu(x) = \int_{t_0}^{\infty} r(x(t), \mu(x(t))), \tag{3}$$

where $J : \mathbf{X} \to \mathbb{R}$ is the infinite-horizon cost, and r is a utility function (or the instantaneous cost). Thus, starting at an initial condition $x(t_0) = x_0$, a system is governed by (2) through μ for the time $t = [t_0, \infty)$, which results in the cost J_μ based on the instantaneous reward r.

The optimal control problem associated with the infinite horizon cost (3) and the dynamic constraint (2) pertains to finding the admissible policy $\pi^* \in \Pi$ that yields the minimum cost. In other words, the optimal value function V^* is the cost J_π over π, which satisfies

$$V^*(x) = \inf_{\pi \in \Pi} J_\pi(x) = J_{\pi^*}(x). \tag{4}$$

To find V^* and the associated optimal policy π^*, a Hamiltonian is defined that augments the performance measure and the dynamic constraint, and then the stationary condition is utilized to obtain a closed-form expression of the optimal policy, which is a function of the optimal value function [34], [35]. The optimal value function is then built by solving the associated Hamilton-Jacobi-Bellman (HJB) equation, or the integral Bellman equation, which is analogous to the Bellman equation but is written for the continuous-time dynamic constraints (2). In this context, adaptive/approximate dynamic programming techniques have been used to learn a solution to the HJB equation [36]–[40]. However, RL approaches have not been studied in the context of ensemble systems, and the goal of this paper is to develop a formal RL framework for ensemble systems, and analyze its feasibility in controlling dynamic populations.

Next, we illustrate the application of an RL-based control design procedure for a single dynamical system and highlight the challenges associated with its application to an ensemble system.

Example 1: A single Bloch system: To understand how the RL-based control strategy is employed to design controls, we start with the optimal control problem associated with a single Bloch system. This system, which is a bilinear control system evolving on the sphere $\mathbf{S}^2$, is widely studied in quantum control applications [16], and its governing equations are given by

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & -\omega & u_1 \\ \omega & 0 & -u_2 \\ -u_1 & u_2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{5}$$

where $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}'$ denotes the bulk magnetization of a sample of nuclear spins, ω denotes the Larmor frequency of the spins, and u_1 and u_2 are the radio-frequency (rf) fields applied to steer the bulk magnetization to a desired state.

We start with a simple task of steering the Bloch system states x from $x_0 = [0, 0, 1]'$ to $x_F = [0, 0, -1]'$ and consider the system in the rotating frame with respect to ω, i.e., $\omega = 0$ [1]. In this preliminary example, the control set is defined as $\mathbf{U} = (u_1, u_2) \in \{(1,1), (1,0), (0,1), (0,0)\}$. With this control set, the value function profile corresponding to the Bloch system is learned using the temporal-difference (TD) learning scheme [41], and the learned value function is shown in Figure 1 (a). Specifically, the approximation of value function is done through polynomials of states up to the second order. The value function, which is a function of the state variables, can then be used to decide the feedback controls that yield the minimum cost while steering the state of the system between

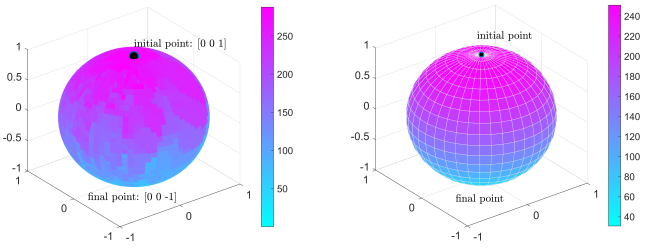

Fig. 1. Value function profile and approximation. (Left) Value corresponds to each state travelled. Unexplored states are colored in light blue. (Right) Approximated value plotted using function of states generated by polynomials up to second order.

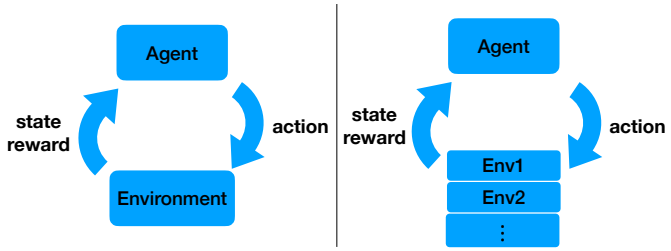

Fig. 2. Illustration of the traditional reinforcement learning task involving a single RL agent navigating in an environment (left) and task considered in this paper, where a single RL agent navigates in a continuum of environments (right).

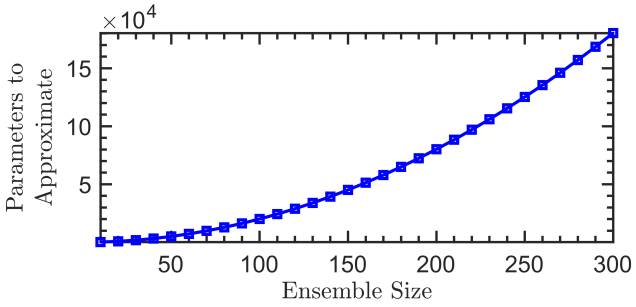

Fig. 3. Ensemble size v.s. parameters required to approximate.

the initial and final points of interest. In this specific case, the optimal control generated from the value function profile is just $(u_1, u_2) = (1, 0)$ for all time instances, which will steer the system from x_0 to x_F.

C. RL problem formulation for an ensemble system

Motivated by the success of controlling the Bloch system, we formally expand the application of RL to ensemble systems. Primarily, the difference between the traditional RL task and the one considered in this paper is illustrated in Fig. 2. In the classical RL, a single agent is sent out to explore and learn a policy in a single environment, while our goal is to use RL in multiple-environment scenarios, which embody an ensemble of systems generating distinctive state trajectories and rewards even under the identical control input.

We study a general class of input-affine ensemble systems defined on a Hilbert space, governed by the dynamics

$$\frac{d}{dt}x(t,\beta) = f(\beta, x(t,\beta)) + \sum_{i=1}^{l} g_i(\beta, x(t,\beta))u_i(t), \qquad (6)$$

where $x(t,\cdot) \in L^\infty(K, \mathbb{R}^n)$, the space of $\mathbb{R}^n$-valued essentially bounded measurable functions over K; the functions f and $g_i, i = 1, \ldots, l$, are smooth nonlinear maps representing the drift and the control vector fields, respectively, and K is a compact set [11].

Contrary to the traditional control problem associated with a single system or a finite collection of systems, in an ensemble, only population-level aggregated measurements are available, which poses the fundamental challenge for controlling ensemble systems. Formally, we define an aggregated measurement as follows.

Definition 1 (Aggregated measurement)**.** *For the system in* (6)*, we denote an aggregated measurement at time* t *as a set* $Y(t)$*, composed of 2-tuples, given by*

$$Y(t) := \{(\beta, x(t,\beta)) \mid \beta \in K_t\}, \qquad (7)$$

where $K_t \subset K$.

Note that the observation set K_t is dependent on time, and it is a proper subset of K for each sampling time. This implies that these measurements are available from only finitely many systems and come from different set of systems at each sampling time. The definition of aggregated measurements, thus, indicates both spatial as well as temporal sparisty in feedback data. Consequently, we cannot adopt traditional feedback strategies to learn ensemble control policies. The primary obstacle in using RL in a multi-state set-up originates from the notorious "curse of dimensionality" issue; that is, computational requirements grow exponentially with the ensemble size [42], as illustrated in Fig 3. In the flowing section, we will introduce a moment-based RL framework to remove this obstacle. In particular, we will formally introduce the *moment system* associated with a given ensemble system. The moment system makes the synthesis of feedback controls tractable and has significant advantages over using the original ensemble system in both model-free and model-based aspects, owing mainly to significantly less parameters to explore. The "curse of dimensionality" issue can therefore be alleviated, which will be addressed in detail in Sections III-B and IV. Specifically, the rapid expansion of the parameter space causes convergence issues and efficiency drops in learning algorithms, which will be addressed in Section IV-C and Figure 5.

III. Moment-based Adaptive Ensemble Control

In this section, we introduce a moment-based learning framework to design rewards, objective functions, and feedback policies for a given ensemble control task.

A. Ensemble moments and their dynamics

We begin by defining the notion of ensemble moments associated with an ensemble state. Specifically, we define a transformation $\mathcal{L}$ that associates each point $x(t,\cdot) \in L^\infty(K, \mathbb{R}^n)$

with a moment sequence $m(t) = (m_0(t), m_1(t), m_2(t), \ldots)'$, where, we define the k-th moment associated with the state of an ensemble system (1) as

$$m_k(t) = \int_K \beta^k x(t,\beta) d\beta \doteq \mathcal{L}(x(t,\cdot)), \tag{8}$$

for $k = 0, 1, \ldots$. Since K is a compact set and $x(t,\cdot) \in L^\infty$, the moments $m(t) \in \mathbf{M}$ are well-defined for $t \geq 0$, where $\mathbf{M}$ is the space of moment vectors associated with $x(t,\cdot) \in L^\infty$.

Remark 1. *Note that though the definition of the ensemble moments are given with respect to $K \subset \mathbb{R}$, this can be easily extended to higher dimensions (i.e., $K \subset \mathbb{R}^d$ for some $d > 1$), and, in this case, k is a multi-index and $\beta^k = \beta_1^{k_1}\beta_2^{k_2}\ldots\beta_d^{k_d}$, where $k = (k_1, \ldots, k_d)' \in \mathbb{N}^d$ with $\sum_{i=1}^d k_i = k$.*

With the introduction of the ensemble moment sequence $m(t)$ associated with the ensemble state $x(t,\beta)$ at time t, an important follow-up question pertains to the time-evolution of the ensemble moments. Using the ensemble system defined in (6) and the moments defined in (8), the dynamics of ensemble moments can be derived as follows

$$\begin{aligned}
\dot{m}_k(t) &= \frac{d}{dt}\int_K \beta^k x(t,\beta) d\beta = \int_K \beta^k \frac{d}{dt} x(t,\beta) d\beta \\
&= \int_K \beta^k \left[f(x(t,\beta),\beta) + \sum_{i=1}^l g_i(x(t,\beta),\beta) u_i(t) \right] d\beta \\
&= \int_K \beta^k f(x(t,\beta),\beta) d\beta + \sum_{i=1}^l u_i(t) \int_K g_i(x(t,\beta),\beta) d\beta
\end{aligned}$$

The second equality is possible when $x(t,\beta)$ and $\dot{x}(t,\beta)$ are both continuous in t and β. In fact, we may further define $\bar{f}(m(t)) = \mathcal{L}\big(f(x(t,\cdot),\cdot)\big) = \mathcal{L}\big(f(\mathcal{L}^{-1}m(t),\cdot)\big)$ and $\bar{g}_i(m(t)) = \mathcal{L}\big(g_i(x(t,\cdot),\cdot)\big) = \mathcal{L}\big(g_i(\mathcal{L}^{-1}m(t),\cdot)\big)$, where $\mathcal{L}$ is defined as in (8), which we refer to as the moment transformation. The moment dynamics can then be written compactly as

$$\frac{d}{dt} m(t) = \bar{f}(m(t)) + \sum_{i=1}^l u_i(t) \bar{g}_i(m(t)), \tag{9}$$

where $\bar{f} = (\bar{f}_0', \bar{f}_1', \ldots)'$ and $\bar{g}_i = (\bar{g}_{i0}', \bar{g}_{i1}', \ldots)'$ for $i = 1, \ldots, l$, which is also in a control-affine form. Here the existence of an inverse map $\mathcal{L}^{-1}$ that takes each point in $\mathbf{M}$ and assigns a unique point in $L^\infty(K, \mathbb{R}^n)$ can be guaranteed by the results of Hausdorff moment problem [43], [44].

Note that the ensemble and its moment system as in (6) and (9), respectively, are driven by the same control input u_i. This is illustrated in the following example.

Example 2: Ensemble Bloch System: Consider an ensemble of Bloch systems [8], indexed by $\beta = (\omega, \epsilon)$,

$$\dot{x}(t,\beta) = \left[\omega\Omega_z + \epsilon u_1(t)\Omega_y + \epsilon u_2(t)\Omega_x\right] x(t,\beta), \tag{10}$$

where $x(t,\beta) \in \mathbb{S}^2 \subset \mathbb{R}^3$, $\epsilon \in [1-\delta, 1+\delta]$ is the dispersion parameter modeling rf-inhomogeneity, $\omega \in [0,1]$ is the Larmor dispersion, $(u_1(t), u_2(t))$ are the external controls, and the matrices

$$\Omega_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \Omega_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \text{ and } \quad \Omega_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Using the definition of the ensemble moments in (8) with a multi-index $\beta = (\omega, \epsilon)$, we can write down the monomials $\beta^1 = (\omega, \epsilon)' \in \mathbb{N}^1$, $\beta^2 = (\omega^2, \omega\epsilon, \epsilon^2)' \in \mathbb{N}^2$, $\beta^3 = (\omega^3, \omega^2\epsilon, \omega\epsilon^2, \epsilon^3)' \in \mathbb{N}^3$, and so on. Then, the ensemble moments of the first- and the second-order can be defined as $m_1 = (m_{1,1}, m_{1,2})'$, and $m_2 = (m_{2,1}, m_{2,2}, m_{2,3})'$ with $m_{1,1} = \int_K \omega x(t,\beta) d\beta$, $m_{1,2} = \int_K \epsilon x(t,\beta) d\beta$ and $m_{2,1} = \int_K \omega^2 x(t,\beta) d\beta$, $m_{2,2} = \int_K \omega\epsilon x(t,\beta) d\beta$, $m_{2,3} = \int_K \epsilon^2 x(t,\beta) d\beta$, respectively. Furthermore, we can derive the dynamics of the ensemble moments using (9). For instance, the first-order moment dynamics of the Bloch ensemble in (10) can be computed as

$$\begin{aligned}
\dot{m}_{1,1}(t) &= \int_K \left[\omega^2\Omega_z + \omega\epsilon(u_1(t)\Omega_y + u_2(t)\Omega_x)\right] x(t,\beta) d\beta \\
&= \Omega_z m_{2,1} + (u_1(t)\Omega_y + u_2(t)\Omega_x) m_{2,2} \\
\dot{m}_{1,2}(t) &= \int_K \left[\omega\epsilon\Omega_z + \epsilon^2(u_1(t)\Omega_y + u_2(t)\Omega_x)\right] x(t,\beta) d\beta \\
&= \Omega_z m_{2,2} + (u_1(t)\Omega_y + u_2(t)\Omega_x) m_{2,3}.
\end{aligned} \tag{11}$$

Now, we formally define the space of moment sequences in order to properly introduce the objective function and performance measure for the proposed RL-based control of infinite-dimensional the moment system.

Lemma 1. *There is a one-to-one correspondence between functions in $L^\infty([a,b], \mathbb{R}^n)$ and functions in $L^\infty([0,1], \mathbb{R}^n)$.*

Proof. This can be easily shown through a linear transformation $\beta = a + \tilde{\beta}(b-a)$, which maps $\tilde{\beta} \in [0,1]$ bijectively onto $\beta \in [a,b]$ and yields

$$\begin{aligned}
m_k(t) &= \int_a^b \beta^k x(t,\beta) d\beta \\
&= \int_0^1 (a + \tilde{\beta}(b-a))^k x(t, a + \tilde{\beta}(b-a)) d\tilde{\beta}.
\end{aligned} \tag{12}$$

The moment sequences mapped onto the interval $[0,1]$ are referred to as the *scaled moment sequence*, denoted by $\tilde{m}(t)$.

Theorem 1. *Let the system defined on $L^\infty([0,1], \mathbb{R}^n)$ and the transformation defined by $\mathcal{L} : L^\infty([0,1], \mathbb{R}^n) \to \mathbf{M}$ as in* (8)*. Then any $\tilde{m}(t) \in \mathbf{M}$ is a square summable sequence for all $t \in \mathbb{R}$, i.e., $\tilde{m}(t) \in \ell^2$.*

Proof. For any time t and $\tilde{m}(t) \in \mathbf{M}$ with $\tilde{m}_k(t)$ defined as in (8) for $k = 0, 1, \ldots$, by using Hölder's inequality, we have

$$\int_0^1 \beta^k x(\beta) d\beta \leq \|\beta^k\|_p \|x\|_q,$$

where $[p,q] \in [1,\infty]$ and $1/p + 1/q = 1$. Take $p = 1$ and $q = \infty$, we derive

$$\tilde{m}_k(t) = \int_0^1 \beta^k x(t,\beta) d\beta \leq \|x\|_{L^\infty} \cdot \int_0^1 |\beta|^k d\beta = \frac{\|x\|_{L^\infty}}{k+1}.$$

Taking squares on both sides of the inequality and summing up from $k = 0$ to ∞ gives

$$\sum_{k=0}^\infty \tilde{m}_k^2(t) \leq (\|x\|_{L^\infty})^2 \cdot \sum_{k=0}^\infty \frac{1}{(k+1)^2} < \infty,$$

where $\|x\|_{L^\infty} < \infty$. As a result, given a compact set K and an ensemble state $x(t,\cdot) \in L^\infty$, the corresponding moment sequence $\tilde{m} \in \mathbf{M} \subseteq \ell^2$.

For the ease of exposition, we use the original notation $m(t)$ to denote the *scaled moment sequence* in the rest of the paper.

B. Performance measure for controlling ensemble moments

The presented moment representation of ensemble systems transforms an uncountably-infinite ensemble system in (6) to a countably-infinite moment system in (9). We will exploit this reduction and utilize the moment system to facilitate the synthesis of RL-based control policies. We will further demonstrate that, unlike traditional discretization techniques presented in [23], this moment-based RL formulation for optimal control of ensemble systems averts the issue of curse-of-dimensionality [34].

To fix ideas, we consider the problem of steering an ensemble system from an initial profile $x_0 \doteq x(t_0,\beta)$ to a final profile x_F. Then, the associated moment sequences to x_0 and x_F are given by the moment transformation, i.e., $m_0 = \mathcal{L}(x_0)$ and $m_F = \mathcal{L}(x_F)$, respectively. To quantify the performance of this task, we define the performance measure,

$$J = \int_0^\infty \langle \hat{m}(t), \hat{m}(t) \rangle_{\ell^2} + \langle u(t), u(t) \rangle_{\mathbb{R}^l} \, dt, \tag{13}$$

involving the trade-off between the terminal error of steering and the control energy, where $\langle \hat{m}(t), \hat{m}(t) \rangle_{\ell^2}$ and $\langle u(t), u(t) \rangle_{\mathbb{R}^l}$ are inner products on ℓ^2 and $\mathbb{R}^l$, respectively, and $\hat{m}(t) = m(t) - m_F$. A general optimal control problem of an ensemble system as in (6) can then be formulated as optimizing, e.g., minimizing, J in (13) subject to the moment dynamics presented in (9). Note that J is well-defined because $m(t) \in \mathbf{M} \subset \ell^2$ according to Theorem 1. Such an optimal control problem remains highly challenging as the moment system is infinite-dimensional.

Remark 2. *Instead of defining the performance measure in terms of the ensemble moments, the direct approach is to define the measure in terms of the error in the ensemble profiles (e.g., $x(t,\beta) - x_F$). Assuming perfect knowledge of the ensemble system dynamics, an open-loop optimal control solution was proposed in [23]. Specifically, this problem was made tractable by discretizing the parameter space K to implement the iterative procedure [23]. Hence, as the number of systems sampled from K increases, the learning problem becomes computationally intractable leading to the resurfacing of the curse-of-dimensionality issue [34].*

As solving an optimal control problem involving a classical nonlinear system as in (2) typically requires finding a solution to the associated Hamilton-Jacobi-Bellman (HJB) equation [34], which in general cannot be solved analytically. In the next section, we will formally introduce a moment-based RL learning algorithm to systematically and effectively approximate and learn optimal control policies for ensemble systems through their truncated moment systems.

IV. Learning strategy for ensemble control

The development of a moment system associated with an ensemble system as introduced in III-A motivates the ensuing investigation of the ensemble control paradigm through moment dynamics. Controlling an ensemble system is difficult in nature because precise aggregated measurements from a large ensemble are often unavailable. Besides, RL generally does not scale well with multiple environments. On the other hand, while computing moments from data is possible, it is computationally infeasible to evaluate infinite orders of moments, which uniquely represent the original ensemble states, at each sampling time. To mitigate this computational intractability, we investigate if a finite-order (approximate) moment system could be used for the design of feedback regulators to fulfill a control task given. Specifically, in the following, we approximate the ensemble states as well as the performance measure in (13) using ensemble moments up to order M and develop a learning algorithm using the truncated moment sequences. We then show that while our algorithm outputs a slight performance drop when the truncated moments are used to design control policies, it circumvents the numerical inefficiency and infeasibility issues that occur when learning a broadcast control policy for a large ensemble of systems, as reported in [23].

A. Approximating objectives and rewards via truncation

The dynamics of the truncated moment system of order M follow

$$\begin{aligned} \dot{\mathbf{m}}(t) &= \bar{f}_M(\mathbf{m}(t)) + \sum_{i=1}^{l} u_i(t) \bar{g}_{M,i}(\mathbf{m}(t)) \\ &= \bar{f}_M(\mathbf{m}(t)) + \bar{g}_M(\mathbf{m}(t)) u(t), \end{aligned} \tag{14}$$

where $\mathbf{m} = (m_0', m_1', \ldots, m_M')'$ denotes the *truncated ensemble moments* of order M, and $\bar{f}_M$ and $\bar{g}_M$ are the corresponding truncated drift and control vector fields. The performance measure $\mathcal{J}$ with respect to the truncated moment sequence and the utility function r are given by

$$\mathcal{J} = \int_0^\infty r(\mathbf{m}(t), u(t)) dt, \tag{15}$$

$$r(\mathbf{m}, u) = (\mathbf{m} - \mathbf{m}_F)' Q_M (\mathbf{m} - \mathbf{m}_F) + u' R u, \tag{16}$$

where $Q_M \succ 0$ is now a finite-dimensional positive definite matrix, and $\mathbf{m}_F$ denotes the truncated moment sequence of the target state. We then have the Hamiltonion for the truncated moment system following [34], [35] as

$$\begin{aligned} H_M(\mathbf{m}, u, -V_{\mathbf{m}}) = &-\dot{\mathbf{m}} \cdot V_{\mathbf{m}}' \\ &- \big((\mathbf{m} - \mathbf{m}_F)' Q_M (\mathbf{m} - \mathbf{m}_F) + u' R u \big), \end{aligned} \tag{17}$$

where $V_{\mathbf{m}}$ is the partial derivative of the value function with respect to $\mathbf{m}$ of appropriate dimension. The HJB equation with respect to the optimal value function V^* yields an optimal control sequence u^*, which satisfies the relation

$$\begin{aligned} 0 &= \min_{u \in \mathbf{U}} \{ r(\mathbf{m}^*, u) + \dot{\mathbf{m}} \cdot V_{\mathbf{m}}' \} \\ &= \dot{\mathbf{m}} \cdot V_{\mathbf{m}}^{*\prime} + \big((\mathbf{m}^* - \mathbf{m}_F)' Q_M (\mathbf{m}^* - \mathbf{m}_F) + u^{*\prime} R u^* \big). \end{aligned} \tag{18}$$

Now, using the first-order necessary condition [45], we obtain the desired relationship between the optimal control law and the optimal value function as

$$u^*(t) = -\frac{1}{2}R^{-1}\bar{g}_M(m(t))'V^*_{\mathbf{m}}(\mathbf{m}(t)). \tag{19}$$

Remark 3. *Note that while the moment system* (9) *is infinite-dimensional, we consider the performance measure* (15) *and define the optimal value function in terms of the truncated moment sequence, which is of finite dimension. Therefore, the optimal value function learned as a solution of the equation* (18) *will not yield an optimal policy for the moment system. However, it will be demonstrated with numerical examples in Section V that this learning strategy yields an approximately optimal control solution for given ensemble control tasks.*

Remark 4. *The advantage of using the moment system instead of the original ensemble system comes from the fact that (i) sample moments can be efficiently computed using the aggregated measurement data and (ii) $m \in \ell^2$ guarantees a quantifiable approximation of a truncated moment system to the infinite moment system. The analysis on the truncation of the moment sequences is presented in Section V-E. On the other hand, although direct discretization of the ensemble system by sampling the parameter in K to obtain a finite collection of systems may be a natural approach, the optimal strategy for controlling the sampled subsystems has no guarantee to be optimal or even a feasible control policy for the entire ensemble.*

Next, we present the algorithm and implementation details to learn the optimal policy in (19) by using the moment-based RL approach.

B. Implementation: discretization and value approximation

We exploit the function approximation property of a parameterized network with linear readout, taking the form $N(z) = \theta'\psi(z)$, where $N(z)$ is the network output, z is the input data, θ is a vector of weights (parameters) and $\psi(z)$ is a vector of activation functions of appropriate dimensions. Our goal is to approximate the value function $V^*_{\mathbf{m}}(\mathbf{m}(t))$ that forms a solution to the HJB equation (18) with the parametrized model such that $V^*_{\mathbf{m}}(\mathbf{m}(t)) \approx \psi(m(t))'\theta^*$ for some ideal weight vector θ^*.

To this end, we first introduce temporal discretization, which defines an equidistant partition of the simulation time t_F by

$$t_{k+1} - t_k = t_F/(N+1), \quad 0 \le k \le N.$$

Here the time interval $[0, t_F]$ is divided into $N+1$ subintervals $[t_k, t_{k+1}], 0 \le k \le N, 0 = t_0 < t_1 < t_2 < \cdots < t_{N+1} = t_F$. During the process of simulation, a set of state-cost training pairs is collected along the trajectory, and $\theta^{(i)} \in \mathbb{R}^s$ is chosen to minimize the error between sample costs $r(\mathbf{m}(t_k), u(\mathbf{m}(t_k)))$ and approximated cost, i.e., $V^{(i)}(\mathbf{m}(t_{k+1})) - V^{(i)}(\mathbf{m}(t_k)) \approx (\psi(\mathbf{m}(t_{k+1}))' - \psi(\mathbf{m}(t_k))')\theta^{(i)}, k = 0, \ldots, N$, in a least square sense. Although the value function to be approximated is generally non-convex, an advantage of such a linear parametric representation is to offer a closed-form solution. By simulating the system from $t_0 = 0$ to $t_{N+1} = t_F$ and collecting the instantaneous costs along the trajectory, we solve the minimization problem defined as

$$\theta^{(i)} = \arg\min_{\theta} \|\Psi\theta - \mathbf{r}\|, \tag{20}$$

for each iteration $i = 0, 1, \ldots$, where

$$\mathbf{r} = \begin{bmatrix} r(\mathbf{m}(t_1), u(\mathbf{m}(t_1))) \\ r(\mathbf{m}(t_2), u(\mathbf{m}(t_2))) \\ \vdots \\ r(\mathbf{m}(t_{N+1}), u(\mathbf{m}(t_{N+1}))) \end{bmatrix} \in \mathbb{R}^{N+1},$$

$$\Psi = \begin{bmatrix} \psi(\mathbf{m}(t_1))' - \psi(\mathbf{m}(t_0))' \\ \psi(\mathbf{m}(t_2))' - \psi(\mathbf{m}(t_1))' \\ \vdots \\ \psi(\mathbf{m}(t_{N+1}))' - \psi(\mathbf{m}(t_N))' \end{bmatrix} \in \mathbb{R}^{(N+1)\times s}.$$

In what follows, we detail our iterative learning approach using a approximate dynamic programming algorithm for the moment systems. The initial moment profile $\mathbf{m}_0$ remains unchanged in each iteration, and the control trajectory generated in the i^{th} iteration is defined by $u^{(i)}(t)$. The proposed learning algorithm is summarized as Algorithm 1 below.

Algorithm 1 Learning Algorithm for Moment Systems

Input: $x(t_0, \beta)$, $x(t_F, \beta)$, E, T, M, K, $\theta^{(0)}$, ψ, ε
Output: θ

Initialization : Calculate $\mathbf{m}(t_0)$, $\mathbf{m}(t_F)$, and $V^{(0)}(\mathbf{m})$.
for $i = 0$ to E **do**
  for $t \le T$ **do**
    $u^{(i)}(t) = -\frac{1}{2}R^{-1}\bar{g}_M(m(t))'V^{(i)}_{\mathbf{m}}$,
    Simulate the system with control law $u^{(i)}(\cdot)$
    Generate the corresponding moment trajectory $\mathbf{m}$ and the instantaneous running cost $r(\mathbf{m}, u) = (\mathbf{m} - \mathbf{m}_F)'Q_M(\mathbf{m} - \mathbf{m}_F) + u'Ru$
    if $\|\mathbf{m} - \mathbf{m_F}\| < \varepsilon$ **then**
      break
    end if
  end for
  Update $\theta^{(i+1)}$ through

$$\theta^{(i+1)} = \arg\min_{\theta} \|\Psi\theta - \mathbf{r}\|,$$

end for
$\theta = \theta^{(i+1)}$
return θ

Remark 5. *The update rule in the algorithm describes the relationship that $V^{(i+1)}$ is generated through the trajectory of $\mathbf{m}$ using $u^{(i)}$, and $u^{(i+1)}$ is the control sequence that is applied in the next iteration, which is based on the updated value function $V^{(i+1)}$.*

We will update θ at the end of every episode through the minimization scheme using least-squares regression as in (20). A block diagram of the proposed learning algorithm for

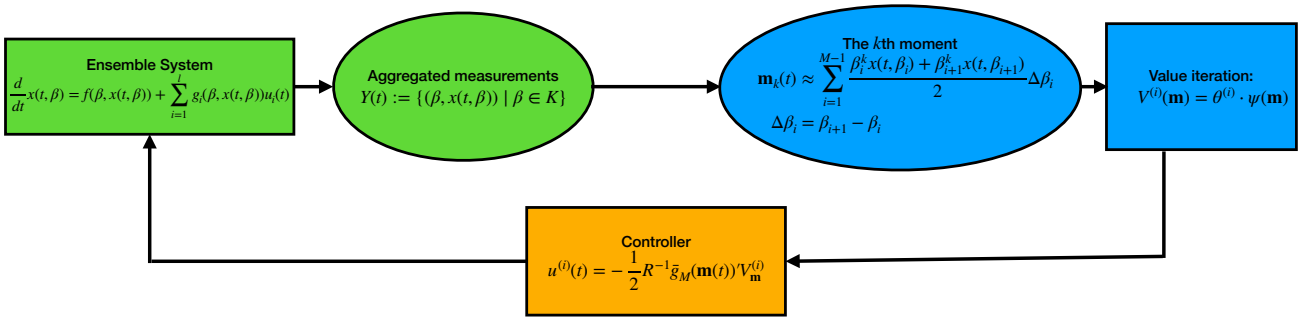

Fig. 4. Block diagram of the proposed algorithm. We evolve the ensemble system to generate aggregated measurement $Y(t)$. From the aggregated measurements, we approximate the first M order moments, where each of the kth order moment in equation (8) is approximated using the trapezoid rule [46]. The rewards (approximated through $\psi(\mathbf{m})$) are collected along the trajectories to update $\theta^{(i)}$, which in turns improves the estimate of the optimal control policy.

moment dynamics is depicted in Fig. 4. We pick the moment-based value function in the ith iteration $V^{(i)}(\mathbf{m})$ to be a quadratic function given by

$$V^{(i)}(\mathbf{m}) = \frac{1}{2}(\mathbf{m} - \mathbf{m}_F)' S^{(i)} (\mathbf{m} - \mathbf{m}_F) = \psi(\mathbf{m})' \theta^{(i)},$$

where $S^{(i)}$ is a constant-valued positive-definite matrix, which has only $s = M(M+1)/2$ independent parameters. The second equality above comes from stacking the upper triangular part of the columns of the matrix S^i into a vector $\theta^{(i)}$ [47]. During the learning process, the truncated moment sequences $\mathbf{m}$, the performance measure $\mathbf{r}$, the matrix formed by the activation functions Ψ and the approximation of the value function V are all computed from data, making this learning framework totally data-driven.

Remark 6. *It is clear that system dynamic $\bar{f}_M$ does not show up in the update rule above, which is a result of differentiating the HJB equation with respect to control $u(t)$ in systems of control-affine form. Therefore, the method is valid without the knowledge of the drift dynamics for the moment system. For completeness, interested readers can refer to the convergence analysis in the work of [37] and [47].*

C. Rationale for learning policies using moment systems

Using moment systems to synthesize feedback controls has significant advantages over using the original ensemble system in both model-free and model-based aspects as addressed in Remarks 2 and 4. More importantly, the moment-based RL method for learning optimal policies is scalable, which is a missing element in classical RL algorithms applied to ensemble systems. This can be demonstrated in terms of the number of learning parameters and convergence performance as shown in Fig. 5(a). As we increase the ensemble (sample) size from 10 to 300 systems, the number of parameters that requires tuning remains constant in the proposed strategy using the moment system for a given truncation order; however, the number of parameters significantly grows when the original ensemble systems are used after discretization of the parameter space K. This gigantic increase in the number of parameters to be trained not only jeopardizes the efficiency of the learning algorithm, but also make the the entire learning process unsuccessful. To solve for equation (20), we require the regressor matrix Ψ to have a dimension of at least $N+1 > s$ to generate a meaningful solution θ provided that Ψ is persistently exciting [48]. For example, in Fig. 5(b), we observed that the RL algorithm for approximating the value function V as a function of ensemble system states did not converge when the number of systems in the ensemble increased. In this case, where the RL algorithm was used to learn a value function for the ensemble system without any moment-transformation, the total number of parameters required to approximate the value function increased exponentially, aggravating the curse-of-dimensionality issues of the RL algorithm. On the contrary, when the learning problem is formulated using the moment systems, the learning algorithm was stable and had better convergence properties.

Note that the selection of an appropriate order of truncation that sufficiently approximates the ensemble system dynamics is an important design step in the moment-based RL approach. In the next section, we present various examples, involving different ensemble control tasks for linear, bilinear, and nonlinear ensemble systems, where we illustrate that low orders of truncation are sufficient to yield excellent control performance. To the best of our knowledge, this work is the first RL-based control framework offering tractable strategies for synthesizing optimal feedback policies for ensemble systems.

V. Simulation Analysis

In this section, we illustrate the capability of our moment-based RL framework in synthesizing controls for diverse classes of ensemble systems.

A. Example 3: Ensemble of harmonic oscillators

Consider an ensemble of linear dynamical systems given by

$$\frac{d}{dt}x(t,\beta) = A(\beta)x(t,\beta) + Bu(t), \tag{21}$$

where $A(\beta) = \begin{pmatrix} 0 & -\beta \\ \beta & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and $\beta \in [-1, 1]$. It was shown that this ensemble system is ensemble controllable [8]. We considered the task of steering this ensemble from the initial profile $x_0 = (1,1)'$ to the target profile $x_F = (0,0)'$ for

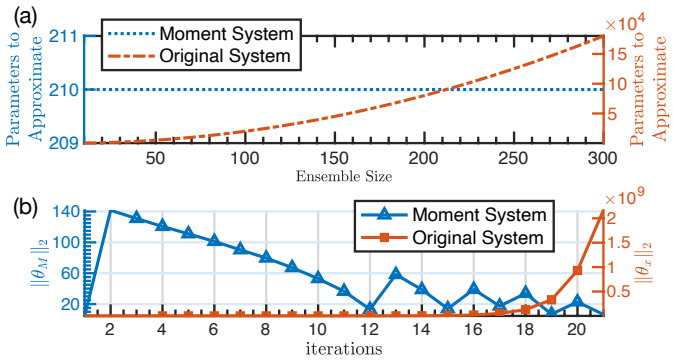

Fig. 5. (a) Parameters needed to approximate the value function using an RL algorithm when truncated moment systems are used (blue) and when discretized harmonic oscillator ensembles are used (red). (b) Convergence analysis of learning algorithms that are based on moment system (blue) and the original system (red). The x-axis represents the number of iterations and the blue and red y axes are the Euclidean norm of the parameters obtained via RL algorithms with the moment system and the ensemble system, respectively.

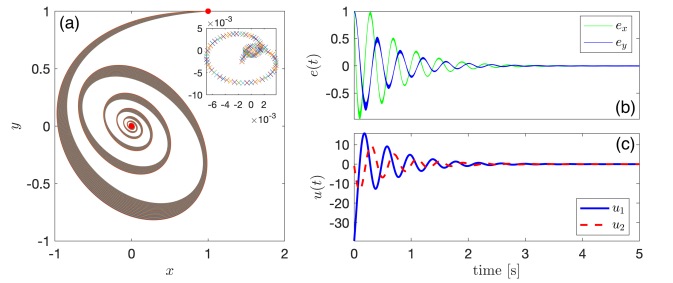

Fig. 6. (a) Trajectories of the controlled ensemble harmonic oscillators ($N = 100$). The final states recorded in the figure demonstrates the target state (marked as a red dot) and simulated endpoint of the ensemble (marked as colored $\times$). (b) Errors with respect to target state along the trajectories. (c) The feedback control trajectories learned using 10 moments.

all β by minimizing the performance measure defined in (13) with order of moment system defined by $M = 20$. For simulations, we sampled 100 Harmonic oscillators from the ensemble with their frequencies uniformly distributed in $[-1, 1]$. The associated moment system follows, for $0 \leq k \leq M-1$,

$$\dot{\mathbf{m}}_k(t) = \frac{d}{dt}\int_\Omega \beta^k x(t,\beta)d\beta = \int_\Omega \beta^k (A(\beta)x + Bu)d\beta,$$

for which the control policy was derived using (19) as

$$u(t) = -R^{-1}\left(B\sum_{k=0}^{M}\int_\Omega \beta^k d\beta\right)' V_{\mathbf{m}}. \tag{22}$$

In Fig. 6, we show the state trajectories as well as the control policies learned via Algorithm 1. Following our RL framework, the ensemble was successfully steered from the initial profile to the desired target profile.

B. Example 4: Ensemble of Bloch systems

Consider an ensemble of Bloch systems, modeling the time-evolution of a sample of nuclear spins immersed in an external magnetic field, given by

$$\dot{x}(t,\beta) = (\omega\Omega_z + \beta\,[u_1(t)\Omega_y + u_2(t)\Omega_x])\,x(t,\beta), \tag{23}$$

where $x(t,\beta) \in \mathbb{R}^3$, $\omega = 0.6$, $\beta \in [0.9, 1.1]$ models the inhomogeneity in the applied control fields u_1 and u_2 [1], and $\Omega_x, \Omega_y, \Omega_z$ are defined as in Example 2.

We considered the design of an inversion pulse $(u_1(t), u_2(t))'$ that drives the ensemble from the initial state $x_0 = (0, 0, 1)'$ to the target state $x_F = (0, 0, -1)'$. We used the same performance measure as in Example 3 with the order of moments chosen by $M = 10$, $Q_M \in \mathbb{R}^{30\times 30}$ with diagonal entries being 10^{-10}, and R is the 2×2 identity matrix. In Fig. 7 (a) and (b), state trajectories of the ensemble are shown. The control inputs learned using the proposed approach is shown in Fig. 7(c), which are applied to the moment system resulting in the states shown in Fig. 7(d).

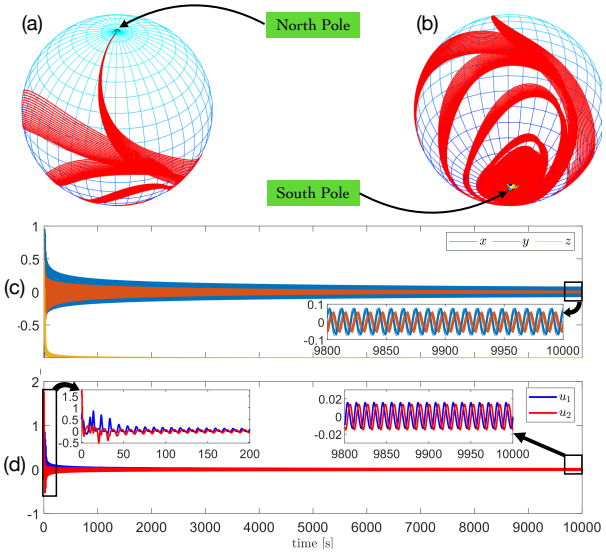

Fig. 7. The control task of an ensemble of 20 Bloch systems. (a) and (b) are the simulated trajectories from the initial state (north pole) to the target state (south pole). (c) The state trajectories of x, y, and z. The zoomed-in figure shows the trajectories of the ensemble in the x and y direction in the last 200 seconds of the simulation. (d) The control input learned through the reinforcement learning framework. The left and right zoomed-in figures show the initial and final 200 seconds of the entire simulation period, respectively.

C. Example 5: Ensemble of nonlinear systems

In this example, we considered steering an ensemble of nonlinear systems of the form, $\dot{z}(t,\beta) = f(z,\beta) + g(\beta)u(t)$ with $\beta \in [0.5, 1]$, where

$$z = (x, y)', \quad f = \beta\begin{pmatrix} y \\ -y - \sin x \end{pmatrix}, \quad g = (0, \beta)', \tag{24}$$

from the initial $x_0 = (2, 1)'$ to the final state $x_F = (1, 0)'$ for all β. We developed the control policy using the associated moment system of order $M = 20$. The same performance measure as in Examples 3 and 4 was used with the ij^{th} entry of the matrix $Q_M \in \mathbb{R}^{40\times 40}$ was selected as $Q_{M(i,j)} = 0$ for all $i \neq j$ and $Q_{M(i,j)} = 1$ for all $i = j$, and R was set as 2×2 identity matrix. As shown in Fig. 8(a), the learned

control sequence steered the entire ensemble approximately to the neighborhood of the target state. The x, y trajectories and controls are shown in Fig. 8(b) and Fig. 8 (c), respectively.

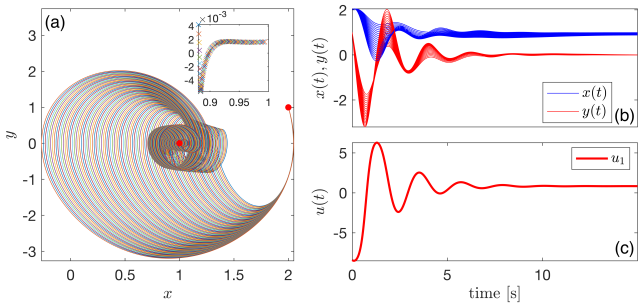

Fig. 8. The trajectory snapshots of the ensemble systems for Example 3. (a) State trajectories of an ensemble of 300 nonlinear systems described in equation (24). The zoomed-in figure on the top-right corner of (a) shows the neighborhood of the desired final states x_F of the ensemble. The simulated end states of the trajectories are marked with colored $\times$. (b) The state trajectories in the x and y direction respectively. (c) The control input learned through the reinforcement learning framework. Note that the control vector field g is zero in the first entry, so we only the control u_1 to fulfill the steering task.

D. Example 6: Pattern formation

In this example, we present a more complex task of pattern formation in an ensemble of harmonic oscillators modeled in (21). For our simulation experiment, we used 20 harmonic oscillators with β uniformly distributed in $[0.6, 0.8]$. The target pattern x_F is a circle of radius 1 centered at the origin; namely, $x(t_F, \beta_j) = (\cos((j-1)/2\pi), \sin((j-1)/2\pi)), 1 \leq j \leq 20$, and the entire ensemble was initialized at $(5, 5)$ at $t = 0$. The results in Fig. 9(a) demonstrates that the proposed algorithm learned the controls to achieve this challenging pattern shaping task via feedback controls in Fig. 9(b).

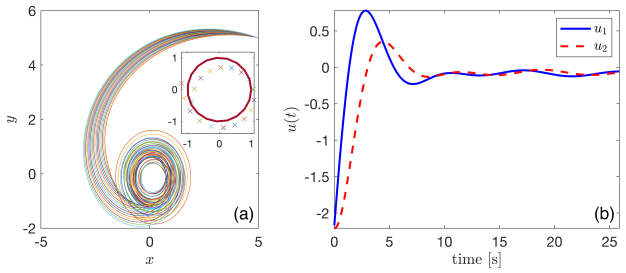

Fig. 9. An ensemble of 20 harmonic oscillators with dispersion in $\omega \in [0.6, 0.8]$. (a) Trajectories of the ensemble of 20 harmonic oscillators. The inset demonstrates the final state reached (marked in $\times$) and the desired final state (solid curve). (b) The control input learned through the RL framework using $M = 10$.

E. Decreasing error with increasing truncation terms

To analyze the influence of the truncation of the moment terms on the quality of the learned control input, we studied the control tasks in Examples 4-6 again with the same range of β and a consistent ensemble size of 50. The learning algorithm was employed for various order of moments M ranging from 2 to 50. As recorded in Fig 10, we observed that the least absolute errors (LAE) given by $\Sigma|\tilde{m}_F - \mathbf{m}_F|$, where $\tilde{m}_F$ denotes the approximated final moment profile computed via trapezoid rule, decreased as the order of moment M increases. This result was expected because the higher the order of moments considered, the truncated moment sequence $\mathbf{m}(t)$ approximates the ensemble profile better. This improvement in approximation is reflected in the learned control sequence, which is capable of steering the entire ensemble closer to the target state. This is due to the property of $m \in \ell^2$, whereby given the truncated moment sequence $\mathbf{m}$ with order up to M, it is guaranteed that as $M \to \infty$, $\mathbf{m} \to m$.

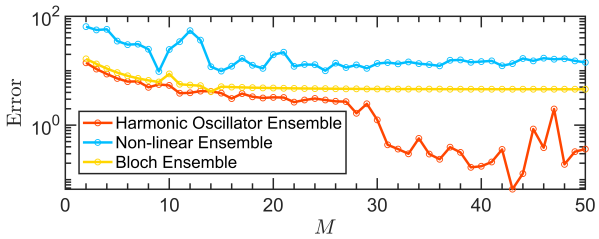

Fig. 10. Least absolute error is plotted with an increasing number of truncation order M. From the figure, the error becomes stable after $M = 20$ for the nonlinear and bilinear ensemble. For linear ensemble, smaller errors can be observed even when we take more truncation orders.

VI. Performance Analysis

It is worth mentioning that the performance of our algorithm is robust to the size of the aggregated measurement. Here, we provide an illustrative example using a population of harmonic oscillators described in Section V-A. Once the range of the dispersion parameter β is known, this method can be extended to a greater number of systems. In the following analyses, we look at how the controls learned and the final regulation error varies under two cases: (i) when the size of the population is varied, and (ii) when varying amount of aggregated measurements were available at each sampling instant.

For the first case, we applied our method to a total number of 10 harmonic oscillators with moments calculated up to the $10th$ order. The controls learned is shown in Fig 11(a). The same control input was then applied to ensembles with varying number of systems, i.e, the control input was applied to ensembles ranging from 10 to 100 systems, whose dispersion parameters were selected from the interval $K = [0.9, 1.1]$. The average error in the final state is recorded in Fig. 11(b), demonstrating the successful control of the ensemble. This was calculated as $1/n \sum_1^n |x_F - x_f|$, where x_F is the desired final state, x_f is the simulated final state, and n is the number of harmonic oscillators in each ensemble.

In the second case, we checked the control performance of the moment-based control policy when the set of states measured is limited. In equation (7), we define the aggregated measurement at time t as $Y(t)$, in which the observation set K_t varies with time. That is to say, the number of data that

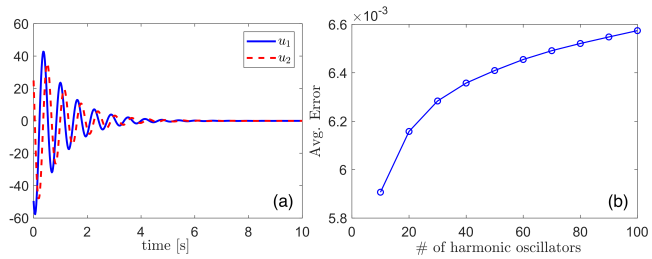

Fig. 11. Performance analysis of the same input applied to increasing size of ensemble. (Left) Control input learned using 10 harmonic oscillators and 10 moments. The task is the same as a single harmonic oscillator described in (21). (Right) Number of harmonic oscillators v.s. average error of the final states. The error along x and y directions are not as noticeable as the one along z direction. (Bottom-right) The control input learned through the RL framework.

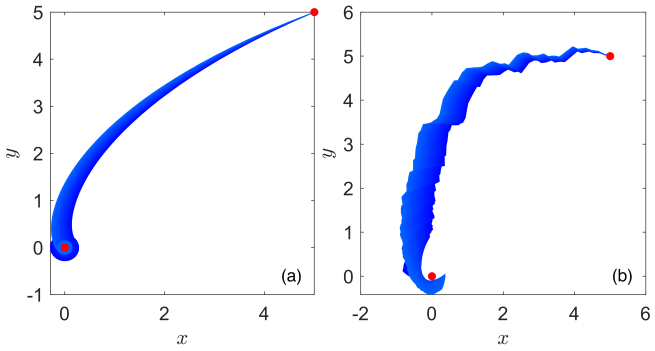

Fig. 13. Controlled trajectories of ensemble Harmonic oscillators when data corresponding to (a) 10 oscillators are missing (b) 95 oscillators are missing.

we observe at the sampling time t_i may not be the same at t_j for $i \neq j$. To analyze the influence of such partial and incomplete measurements on our learning framework, we simulated the time-evolution of an 100-oscillator ensemble with only a limited number of tractable (observable) oscillators at each time instance.

As shown in Fig. 12, we were able to fulfill the task of steering the population to the neighborhood of the desired states despite missing over 80 percent of the observations. The reason is that although the states available are decreasing, we found that the computed sample moments up to arbitrarily high order (in this examples, we set $M = 20$) were good approximations of the actual moments. Throughout the simulation, we randomly dropped the observations for (a) 10 oscillators and (b) 95 oscillators at each time instance. The comparison of trajectories learned are shown in Fig. 13. Note that the success of driving the ensemble to the neighborhood of the desired states owes to the small range of the dispersion parameters; in this case $\beta \in [0.9,\ 1.1]$. When the range of β was increased, an increasing number of moments were required to steer the population, and a lower tolerance for missing measurements was observed.

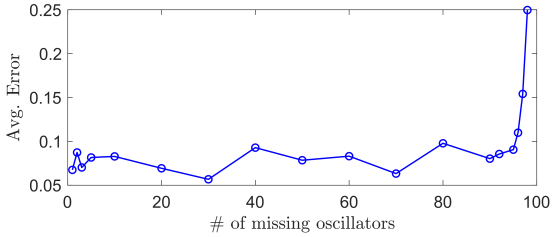

Fig. 12. Performance analysis with limited measurements. The total number of ensemble is 100, whereas only limited number of oscillators are observable. The average error is calculated the same way as our previous analysis. In this figure, we can see that the error is acceptable even when 80% of the measurements from the oscillators are missing. The error grows drastically when 90% of the oscillators are not observable.

VII. Conclusions

In this paper, we propose a moment-based RL scheme to learn a control sequence for steering a population of dynamical systems. By introducing the notion of moments, along with tools from RL, we were able to design a control sequence for steering the moments of the ensemble, which are representative of the entire population, from an initial state to a desired final state. The proposed algorithm bypasses the derivation of a closed form solution to the control sequence, but learns the control directly from data. Moreover, the three examples with linear, bilinear, and nonlinear ensemble systems demonstrate the feasibility of the proposed method. Furthermore, we also demonstrated that controlling the moments of the ensemble is similarly effective in controlling the original systems, which is helpful when parts of the snapshots of the ensemble are missing or only partial feedback information of the population is available.

Using our framework, we can also form a desired pattern for the ensemble via feedback using our approach. To achieve a unique pattern transfer, the patterns (characterized by the distribution of the final states in the parameter space) must be uniquely represented by the moment sequence so that steering the moment sequence results in the transformation to the desired patterns. In addition, the performance of RL method using a moment-based framework is robust to missing data, which addresses a common problem in many practical applications. The method also scales well with increasing ensemble size, and this allows for the possibility of utilizing feedback for nonlinear ensembles in diverse fields. The numerical analysis demonstrated that the truncation error converges with increasing order of moments. However, a formal quantification of the computational errors due to truncating the moments and approximations in computing the moments are not presented here and will be part of our future research.

References

[1] J.-S. Li and N. Khaneja, "Control of inhomogeneous quantum ensembles," *Phys. Rev. A*, vol. 73, p. 030302, Mar 2006. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.73.030302

[2] D. M. Keenan, J. Licinio, and J. D. Veldhuis, "A feedback-controlled ensemble model of the stress-responsive hypothalamo-pituitary-adrenal axis," *Proceedings of the National Academy of Sciences*, vol. 98, no. 7, pp. 4028–4033, 2001. [Online]. Available: https://www.pnas.org/content/98/7/4028

[3] N. Augier, U. Boscain, and M. Sigalotti, "Adiabatic ensemble control of a continuum of quantum systems," *SIAM Journal on Control and Optimization*, vol. 56, no. 6, pp. 4045–4068, 2018. [Online]. Available: https://doi.org/10.1137/17M1140327

[4] K. Kuritz, S. Zeng, and F. Allgöwer, "Ensemble controllability of cellular oscillators," *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 296–301, 2019.

[5] A. R. Mardinly, I. A. Oldenburg, N. C. Pégard, S. Sridharan, E. H. Lyall, K. Chesnov, S. G. Brohawn, L. Waller, and H. Adesnik, "Precise multimodal optical control of neural ensemble activity," *Nature Neuroscience*, vol. 21, no. 6, pp. 881–893, 2018.

[6] A. Becker and T. Bretl, "Approximate steering of a unicycle under bounded model perturbation using ensemble control," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 580–591, 2012.

[7] J.-S. Li, "Control of inhomogeneous ensembles," *Ph.D. dissertation*, May 2006.

[8] J.-S. Li and N. Khaneja, "Ensemble control of bloch equations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 528–536, March 2009.

[9] J.-S. Li, "Ensemble control of finite-dimensional time-varying linear systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 345–357, Feb 2011.

[10] A. Becker and T. Bretl, "Approximate steering of a unicycle under bounded model perturbation using ensemble control," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 580–591, June 2012.

[11] J.-S. Li, I. Dasanayake, and J. Ruths, "Control and synchronization of neuron ensembles," *IEEE Transactions on automatic control*, vol. 58, no. 8, pp. 1919–1930, 2013.

[12] J.-S. Li, "Ensemble control of finite-dimensional time-varying linear systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 345–357, 2010.

[13] R. Brockett and N. Khaneja, "On the stochastic control of quantum ensembles," in *System theory*. Springer, 2000, pp. 75–96.

[14] W. Zhang and J.-S. Li, "On controllability of time-varying linear population systems with parameters in unbounded sets," *Systems & Control Letters*, vol. 118, pp. 94–100, 2018.

[15] S. Zeng and F. Allgoewer, "A moment-based approach to ensemble controllability of linear systems," *Systems & Control Letters*, vol. 98, pp. 49–56, 2016.

[16] J.-S. Li and N. Khaneja, "Ensemble control of bloch equations," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 528–536, 2009.

[17] W. Zhang and J.-S. Li, "Analyzing controllability of bilinear systems on symmetric groups: Mapping lie brackets to permutations," *arXiv preprint arXiv:1708.02332*, 2017.

[18] X. Chen, "Controllability of continuum ensemble of formation systems over directed graphs," *Automatica*, vol. 108, p. 108497, 2019.

[19] S. Zeng, S. Waldherr, C. Ebenbauer, and F. Allgöwer, "Ensemble observability of linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1452–1465, 2015.

[20] S. Zeng, H. Ishii, and F. Allgöwer, "Sampled observability and state estimation of linear discrete ensembles," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2406–2418, 2016.

[21] J.-S. Li, J. Ruths, T.-Y. Yu, H. Arthanari, and G. Wagner, "Optimal pulse design in quantum control: A unified computational method," *Proceedings of the National Academy of Sciences*, vol. 108, no. 5, pp. 1879–1884, 2011.

[22] A. Zlotnik and J.-S. Li, "Synthesis of optimal ensemble controls for linear systems using the singular value decomposition," in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 5849–5854.

[23] S. Wang and J.-S. Li, "Free-endpoint optimal control of inhomogeneous bilinear ensemble systems," *Automatica*, vol. 95, pp. 306–315, 2018.

[24] W. Zhang and J.-S. Li, "Uniform and selective excitations of spin ensembles with rf inhomogeneity," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5766–5771.

[25] A. Zlotnik, R. Nagao, I. Z. Kiss, and J.-S. Li, "Phase-selective entrainment of nonlinear oscillator ensembles," *Nature communications*, vol. 7, p. 10788, 2016.

[26] K. Kuritz, S. Zeng, and F. Allgöwer, "Ensemble controllability of cellular oscillators," *IEEE Control Systems Letters*, vol. 3, no. 2, pp. 296–301, 2018.

[27] S. Zeng, W. Zhang, and J.-S. Li, "On the computation of control inputs for linear ensembles," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6101–6107.

[28] Y. Chen and J. Karlsson, "State tracking of linear ensembles via optimal mass transport," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 260–265, 2018.

[29] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[30] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*. Courier Corporation, 2012.

[31] S. Sastry and M. Bodson, *Adaptive control: stability, convergence and robustness*. Courier Corporation, 2011.

[32] G. Hong and C. M. Lieber, "Novel electrode technologies for neural recordings," *Nature Reviews Neuroscience*, vol. 20, no. 6, pp. 330–345, 2019.

[33] M. S. Couceiro, C. M. Figueiredo, J. M. A. Luz, N. M. Ferreira, and R. P. Rocha, "A low-cost educational platform for swarm robotics." *International Journal of Robots, Education & Art*, vol. 2, no. 1, 2012.

[34] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, 2009.

[35] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995, vol. 1.

[36] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems*, vol. 12, no. 2, pp. 19–22, 1992.

[37] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477 – 484, 2009.

[38] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear hjb solution using approximate dynamic programming: Convergence proof," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 943–949, Aug 2008.

[39] A. Heydari, "Revisiting approximate dynamic programming and its convergence," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2733–2743, Dec 2014.

[40] D. P. Bertsekas, "Value and policy iterations in optimal control and adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 500–509, March 2017.

[41] G. Tesauro, "Temporal difference learning and td-gammon," *Commun. ACM*, vol. 38, no. 3, p. 58–68, Mar. 1995. [Online]. Available: https://doi.org/10.1145/203330.203343

[42] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.

[43] F. Hausdorff, "Momentprobleme für ein endliches intervall." *Mathematische Zeitschrift*, vol. 16, no. 1, pp. 220–248, 1923.

[44] I. P. Natanson, *Constructive function theory*. Ungar, 1965.

[45] D. Kirk, *Optimal control theory: an introduction*, ser. Prentice-Hall networks series. Prentice-Hall, 1970.

[46] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*. Springer-Verlag New York, 2002.

[47] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 2, pp. 140–153, May 2002.

[48] S. Boyd and S. Sastry, "On parameter convergence in adaptive control," *Systems & Control Letters*, vol. 3, no. 6, pp. 311–319, 1983.