2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Moment Latent Reinforcement Learning for Pattern Control in Swarm Robotic Systems

Wei Zhang[1], Haoyu Quan[1], and Jr-Shin Li[2]

Abstract— **Targeted coordination of swarm robotic systems is an emerging robot control task arising from numerous applications across diverse domains, ranging from medicine and agriculture to cyber-physical systems. However, state-of-the-art control techniques for robot swarms often require comprehensive measurement data for each robot and are not scalable with the growth of the swarm size. To address these issues, in this work, we develop a latent space control architecture for robust manipulation of patterns in arbitrarily large, potentially infinite, robot swarms using only partial measurements. In particular, we model such a swarm as a parameterized control system and formulate its patterns in terms of probability distributions. We then develop a moment kernel transform, which generates a reduced latent space representation for the pattern dynamics of the robot swarm over a reproducing kernel Hilbert space. The moment representation of the robot swarm can be learned using partial measurements of the swarm. Building on this, we propose a reinforcement learning (RL)-based pattern control framework operating on the moment latent space. In this framework, the data is organized to flow between the workspace and moment latent space episodically to achieve both robust control performance and high training efficiency. The proposed moment latent RL framework is validated by various pattern control tasks involving wheeled robot swarms, using both numerical simulations and TurtleBot3 swarms in the Gazebo simulator.**

I. Introduction and Related Works

The control of collective behaviors in swarm robotic systems is a recurrent research topic in robotics, with applications spanning diverse disciplines. Notable examples include precision farming, such as autonomous seeding, harvesting, and weed control, in agriculture [1], [2], [3], [4], [5], targeted micromanipulation and therapeutic agent delivery in clinical medicine [6], [7], [8], and unmanned reconnaissance and surveillance in military [9], [10]. To accomplish such complex missions that surpass the capabilities of single robotic systems, swarm robotic systems require more sophisticated stimuli to coordinate individual robots, enabling the entire swarm to exhibit desired collective behaviors [11], [12], [13], [14].

a) Related works: The mainstream of research into control strategies for robot swarms lies in promoting cooperation between individual robots to tackle population-level tasks. This goal aligns with the scope of many control and optimization techniques developed for multi-agent systems, such as consensus, formation and cooperative control, and distributed optimization, for robot swarms [15], [16], [17], [18], [19]. Applications of these techniques to swarm robotic systems generally require comprehensive information about the configuration of each robot and the interactions between every pair of robots. Consequently, scalability issues arise as the swarm size increases. To address this, mean-field models have been introduced to approximate large-scale robot swarms, particularly those consisting of identical robots, thus enabling the design of mean-field controls for robotic applications using an averaged approach [20], [21], [22].

With the rapid development of sensing technology and computing power, leveraging machine learning models and methods to facilitate data-driven control of robot swarms has recently attracted considerable attention. This trend particularly catalyzes the development of evolutionary swarm robotics. In this research thread, control inputs of robot swarms are parameterized using neural networks, whose trainable parameters are optimized by evolutionary algorithms [23], [24], [25]. On the algorithmic side, reinforcement learning (RL), particularly multi-agent RL, has undeniably wrests the dominance of learning-based approaches to robot swarm control [26], [27]. The core idea is to use an RL agent to model the behavior of each robot in a swarm for learning a joint policy, which guides the entire swarm to accomplish a population-level task optimally [28], [29], [30], [31], [32], [33]. However, it is widely known that machine learning methods suffer from the notorious curse of dimensionality when applied to high-dimensional data, leading to high computational complexity, low sample efficiency, and unsatisfactory learning performance [34], [35], [36]. In the context of robotics, when a robot swarm has a large population size, the applied RL algorithm necessarily operates on a high-dimensional computational space, resulting in these adverse effects. Meanwhile, sensing capabilities are also constrained by limited sensor communication bandwidth so that only partial robots in the swarm can be tracked [37], [38], [39].

b) Our contributions: In this work, we propose a novel learning architecture enabled by the introduction of a *moment latent space*. This latent space provides a systematic platform for the effective operation of RL to learn control policies for targeted manipulation of swarm dynamics in a reduced space of the swarm workspace. It features scalability for arbitrarily large, potentially infinite, robot swarms, which may contain structurally similar robots with different dy-

*This work was supported in part by the Air Force Office of Scientific Research under award FA955021-1-0335 and the National Science Foundation under award 2508439.

[1]Wei Zhang and Haoyu Quan are with the Department of Electrical & Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130, USA. {wei.zhang, quanhaoyu}@wustl.edu

[2]Jr-Shin Li is with the Department of Electrical & Systems Engineering, Division of Computational & Data Sciences, and Division of Biology & Biomedical Sciences, Washington University in St. Louis, St. Louis, MO 63130, USA. jsli@wustl.edu

namic characteristics, rather than identical robots as considered in many existing works. Our development begins by formulating such a swarm with heterogeneous robots as a parameterized control system, referred to as an ensemble system, and its patterns as probability measures. Specifically, we introduce the moment kernel transform, which maps an ensemble system to a moment system defined on a reproducing kernel Hilbert space (RKHS). This moment system gives rise to a reduced latent space representation of the system governing the temporal evolution of swarm patterns, which can be learned using partial measurements of the swarm. This dynamic reduction enables the efficient execution of an RL algorithm over the moment latent space to learn pattern control policies. To strike a balance between training efficiency and learning performance, we perform episodic data exchanges between the moment latent space and the robot workspace. Specifically, robot workspace locations are sent to the moment latent space at the beginning of an episode and the learned control policy is fed back to the workspace at the end of the episode. This process also ensures unbiased learning of the moment system. Fig. 1 illustrates the proposed moment latent RL architecture for pattern control of robot swarms. Furthermore, we validate the proposed framework through both numerical and 3D simulations using diverse pattern control tasks for wheeled robot swarms. The major contributions of this work are summarized as follows.

- Formulation of parameterized ensemble systems for arbitrarily large robot swarms and measure-theoretic characterization of their patterns.
- Development of the moment kernel transform, which generates a reduced latent space representation of robot swarms within an RKHS.
- Design of the moment latent RL architecture through episodic interactions between the latent space and the robot workspace.
- Demonstration of the performance and efficiency of the proposed framework using both numerical and 3D simulations.

c) Paper organization: In Section II, we will introduce the parameterized ensemble system modeling of robot swarms and the measure-theoretic characterization of their patterns, which in turn enables a systematic formation of pattern control for general large-scale systems. In Section III, we will introduce the moment kernel transform and then define an RKHS structure on the moment latent space. Section IV will be devoted to the development of the RL architecture based on episodic interactions between the latent space and the workspace for pattern control of robot swarms. In Section V, we will demonstrate the performance and efficiency of the proposed framework through both numerical simulations and TurtleBot3 swarms in Gazebo simulator.

II. Pattern Control of Swarm Robotic Systems

In this section, we present a general formulation for the control of dynamic patterns in large-scale ensembles of dynamical systems, irrespective of their ensemble sizes. Our formulation consists of two components: representing these dynamic ensembles as parameterized control systems and characterizing their patterns in terms of probability measures. This proposed pattern control formulation is specifically applied to swarm robotic systems, which is the primary focus of this paper.

A. Parameterized system representation of robot swarms

In this work, we consider swarms of structurally identical wheeled robots, described by the Dubin's car model (unicycle model), with distinct dynamic characteristics. Such a robot swarm can be represented as a parameterized control system, also referred to as an *ensemble system*, given by

$$\frac{d}{dt}\begin{bmatrix} x(t,\beta) \\ y(t,\beta) \\ \theta(t,\beta) \end{bmatrix} = \beta v(t)\begin{bmatrix} \cos\big(\theta(t,\beta)\big) \\ \sin\big(\theta(t,\beta)\big) \\ 0 \end{bmatrix} + \beta u(t)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{1}$$

where $u(t) \in \mathbb{R}$ and $v(t) \in \mathbb{R}$ are the control inputs manipulating the linear and angular velocities of the robot swarm, respectively, and $X(t,\beta) = \big(x(t,\beta), y(t,\beta)\big) \in \mathbb{R}^2$ and $\theta(t,\beta) \in \mathbb{S}^1$ are the workspace position and heading of the robot characterized by β, respectively. The system parameter β represents the scaling factor affecting the control effects, such as the wheel radius, which varies on the interval $\Omega = [r-\delta, r+\delta]$ around the nominal value r according to a probability distribution λ, due to imperfect manufacturing processes. One significant benefit of this parameterized formulation is that it allows us to consider robot swarms of any population size. In the limit, the ensemble system in (1) represents an infinite swarm of mobile robots with scaling factors spanning the entire tolerated range Ω. In this case, $X_t(\cdot) \doteq X(t,\cdot)$ defines an $\mathbb{R}^2$-valued function over Ω.

B. Measure-theoretic characterization of swarm patterns

At each time, the robots in the ensemble system in (1) form a pattern on the workspace $\mathbb{R}^2$, such as gathering around one point or separating into clusters. The pattern describes how the robots distribute over the workspace. Therefore, it can be mathematically characterized by a probability measure (distribution) μ_t, which for any (measurable) set $B \subseteq \mathbb{R}^2$ reports the proportion $\mu_t(B)$ of the robots in the region B. Because the scaling factors of the robots in B are exactly the inverse image of B under the function X_t, denoted by $X_t^{-1}(B) = \{\beta \in \Omega : X_t(\beta) \in B\}$, the proportion $\mu_t(B)$ coincides with the probability of drawing a sample from $X_t^{-1}(B)$, i.e., $\mu_t(B) = \lambda(X_t^{-1}(B))$. Formally, the measure μ_t defined in this way is the pushforward of λ by the function X_t, denoted by

$$\mu_t = (X_t)_{\#}\lambda. \tag{2}$$

Equivalently, μ_t satisfies $\int_{\mathbb{R}^2} f d\mu_t = \int_\Omega f \circ X_t d\lambda$ for any measurable function $f : \Omega \to \mathbb{R}$. It is worth noting that μ_t also has a total measure of 1, as $\mu_t(\mathbb{R}^2) = \int_{\mathbb{R}^2} d\mu_t = \int_\Omega d\lambda = 1$, and is hence a probability measure.

Pattern control of the robot swarm in (1) then pertains to regulating the "probability distribution" μ_t generated by the

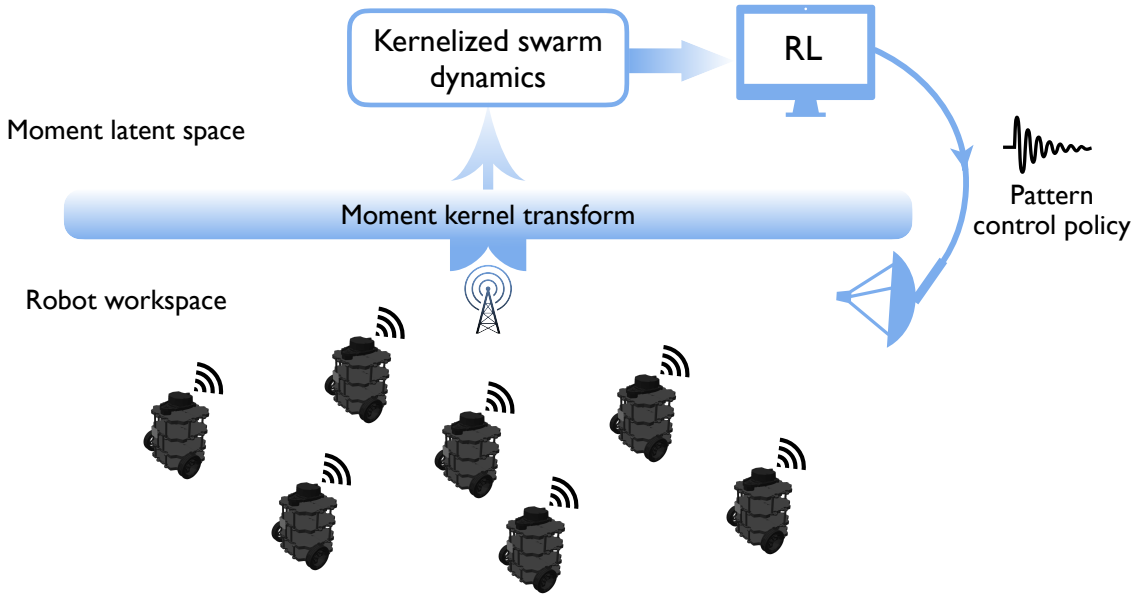


Fig. 1. Illustration of the moment latent reinforcement learning architecture for pattern control of robot swarms.

swarm using the control inputs $u(t)$ and $v(t)$. To facilitate this pattern control task, it is necessary to construct an appropriate representation of μ_t, through which the system governing the pattern dynamics can also be derived. The next section will be devoted to this investigation, where we develop a latent space representation of μ_t in terms of moment sequences.

III. Kernel Representation of Robot Swarms over the Moment Latent Space

In this section, we introduce a moment kernelization technique to facilitate pattern control tasks for the robot swarm in (1). Through the developed moment kernel transform, the swarm pattern is mapped to the moment latent space, over which we construct a kernel representation of the system governing the pattern dynamics. This kernelized system can be approximated using partial measurement data of the robot swarm in an online manner, which then enables the use of online RL approaches for pattern control over the moment latent space.

A. Moment kernel transform

To facilitate distributional control described through the presented measure-theoretic interpretation, it is essential to represent μ_t in an appropriate coordinate system. To this end, we introduce the notion of *ensemble-moments* for the robot swarm in (1), defined by

$$m_k(t) = \int_{\mathbb{R}^2} x^k d\mu_t(x), \tag{3}$$

where $k = (k_1, k_2) \in \mathbb{N}^2$ is a double-index and $x^k = x_1^{k_1} x_2^{k_2}$ is the monomial of total order $|k| = k_1 + k_2$ in $x = (x_1, x_2)$. Using the pushforward definition of the measure μ_t in (2), we obtain an equivalent formula for the k^{th} ensemble-moment as $m_k(t) = \int_{\mathbb{R}^2} x^k d(X_t)_{\#}\lambda(x) = \int_\Omega X_t^k(\beta) d\lambda(\beta)$. Note that this coincides with the k^{th} raw moment of the "random variable" X_t drawn from the probability distribution μ_t.

In practice, it is reasonable to assume that the robots in the swarm spread out in a compact region so that $\sup_{\beta\in\Omega} |X_t(\beta)| \doteq M_t < \infty$, where $|\cdot|$ denotes a norm on $\mathbb{R}^2$. In this case, each ensemble-moment is well-defined as $m_k(t) \le \int_\Omega |X_t|^{|k|} d\lambda \le \int_\Omega M_t^{|k|} d\lambda = M_t^{|k|}$, which also leads to the "exponential convergence" of the moment sequence $m(t) = \big(m_k(t)\big)_{k\in\mathbb{N}^2}$ as $\sum_{k\in\mathbb{N}^2} \frac{1}{k!} m_k(t) \le \sum_{k\in\mathbb{N}^2} \frac{1}{k!} M_t^{|k|} = e^{2M_t} < \infty$, where $k! = k_1! k_2!$. This implies that moment sequences $m(t)$ and robot swarm patterns μ_t are in one-to-one correspondence [40]. Formally speaking, the *moment kernel transform* $\mathcal{K} : \mathcal{P} \to \mathcal{M}$, given by $\mu_t \mapsto m(t)$, is bijective, where $\mathcal{P}$ and $\mathcal{M}$ denote the spaces of swarm patterns and moment sequences, respectively.

The exponential convergence of $m(t)$ shown above further introduces a Hilbert space structure to the moment space $\mathcal{M}$, given by the inner product $\langle m(t_1), m(t_2)\rangle_{\mathcal{H}} = \sum_{k\in\mathbb{N}^2} \frac{1}{k!} m_k(t_1) m_l(t_2)$. Consequently, every moment sequence $m(t)$ satisfies $\|m(t)\|_{\mathcal{H}}^2 = \sum_{k\in\mathbb{N}^2} \frac{1}{k!} m_k^2(t) < \infty$ so that $\mathcal{M}$ is contained within the Hilbert space $\mathcal{H}$ of square-summable (double) sequences with respect to the inner product defined above. It is well-known that $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) [41], and hence $m(t)$ gives a kernel representation of μ_t. Formally, the *moment kernel transform* $\mathcal{K} : \mathcal{P} \to \mathcal{H}$, given by $\mu_t \mapsto m(t)$, is an embedding of the space $\mathcal{P}$ of robot swarm patterns into the RKHS $\mathcal{H}$, and hence a feature map.

B. Moment latent representation of robot swarms

Through the moment embedding, we are able to model the dynamics of the swarm pattern μ_t in the moment latent space using measurement data of the ensemble system in (1). However, when the size of the swarm is large, practical limitations on sensing capabilities and computing power

make it infeasible to obtain comprehensive measurements for each robot. Therefore, a systematic way to approximate the moments using incomplete data is necessary. To this end, we exploit the interpretation of the ensemble-moment $m_k(t)$ as the k^{th} raw moment of the "random variable" X_t. This motivates the introduction of the notion of sample ensemble-moments as

$$\hat{m}_k(t) = \frac{1}{|\Omega_t|} \sum_{\beta \in \Omega_t} X_t^k(\beta) \tag{4}$$

for $k \in \mathbb{N}^2$, where Ω_t is a finite subset of Ω consisting of the system parameters of the observed robots, and $|\Omega_t|$ denotes its cardinality, i.e., the number of observed robots.

Let $\hat{\mu}_t = \frac{1}{|\Omega_t|} \sum_{\beta \in \Omega_t} \delta_{X_t(\beta)} \in \mathcal{P}$ be the empirical distribution on $\mathbb{R}^2$ determined by the workspace locations of the observed robots at time t, where δ_x denotes the point mass at $x \in \mathbb{R}^2$. Then, its k^{th} moments is given by $\int_{\mathbb{R}^2} x^k d\hat{\mu}_t(x) = \frac{1}{|\Omega_t|} \sum_{\beta \in \Omega_t} \int_{\mathbb{R}^2} x^k d\delta_{X_t(\beta)}(x) = \frac{1}{|\Omega_t|} \sum_{\beta \in \Omega_t} X_t^k(\beta)$, which coincides with $\hat{m}_k(t)$. This implies that the sample moment sequence $\hat{m}(t) = \big(\hat{m}_k(t)\big)$ lies in the RKHS $\mathcal{H}$. As a result, the temporal evolution of $\hat{m}(t)$ gives the moment latent representation of the pattern dynamics. Without loss of generality, we assume $\Omega_t = \{\beta_1, \ldots, \beta_n\}$ for all $t \geq 0$, in which case the dynamics of $\hat{m}(t)$ can be derived as

$$\begin{aligned}
\frac{d}{dt}\hat{m}_k(t) &= \frac{d}{dt}\frac{1}{n}\sum_{i=1}^{n} X_t^k(\beta_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{d}{dt} X_t^k(\beta_i) \\
&= \begin{cases}
\frac{k_1 v(t)}{n} \sum_{i=1}^{n} \beta_i x_t^{k_1-1}(\beta_i) \cos\theta_t(\beta_i), & k_2 = 0, \\
\frac{k_2 v(t)}{n} \sum_{i=1}^{n} \beta_i y_t^{k_2-1}(\beta_i) \sin\theta_t(\beta_i), & k_1 = 0, \\
\frac{v(t)}{n} \sum_{i=1}^{n} \beta_i x_t^{k_1-1}(\beta_i) y_t^{k_2-1}(\beta_i) & \\
\quad \cdot \big(k_1 y_t(\beta_i) \cos\theta_t(\beta_i) + k_2 x_t(\beta_i) \sin\theta_t(\beta_i)\big), & k \neq 0.
\end{cases} \\
&\doteq F_k\big(X_t(\beta_1), \ldots, X_t(\beta_n), u(t), v(t)\big)
\end{aligned} \tag{5}$$

Because $\hat{m}(t) \in \mathcal{H}$ as illustrated above, the moment system in (5) necessarily evolves within $\mathcal{H}$ as well, providing the moment latent representation of the pattern dynamics for the robot swarm. It is also important to note that the dynamics of each $\hat{m}_k(t)$ depends only on the control inputs and workspace locations of the observed robots, and hence can be directly learned from the measurement data.

IV. Moment latent online reinforcement learning for pattern control

In this section, we will leverage the developed moment latent representation of the robot swarm to facilitate its pattern control task. In particular, the flow of the data between the work and latent spaces will be fully exploited to establish an instantaneous interaction between these two spaces. This gives rise to an online, on-policy RL architecture for designing an optimal pattern control signal. As a prerequisite to this investigation, it is essential to verify that controlling the moment system, derived from partially observed robots, effectively coordinates the entire robot swarm to achieve the desired pattern.

A. Pattern control over the moment latent space

In the case where observation of the workspace locations is only available for a partial population of robots, without loss of generality, we assume that, at each time t, the system parameters β_i, $i = 1, \ldots, |\Omega_t|$, of the tracked robots are independent random samples drawn from the probability distribution λ. Correspondingly, their workspace locations $X_t(\beta_i)$ are drawn from μ_t independently.

In this scenario, the ensemble sample moment $\hat{m}_k(t)$ introduced in (4) coincides with the sample moment of the probability distribution μ_t. Because $m_k < \infty$ as shown in Section III-A, Khinchine's strong law of large numbers implies that $\hat{m}_k(t) \to m_k(t)$ almost surely as $|\Omega_t| \to \infty$ for all $k \in \mathbb{N}^2$ and time $t \geq 0$ [42]. Formally, we have established the component-wise convergence of the sample moment sequence $\hat{m}(t)$ to the ensemble moment sequence $m(t)$. The consequences of this convergence, crucial for pattern control, include

(1) $\hat{\mu}_t \to \mu_t$ weakly, that is, $\int_{\mathbb{R}^2} f d\hat{\mu}_t \to \int_{\mathbb{R}^2} f d\mu_t$ for any bounded continuous function $f : \mathbb{R}^2 \to \mathbb{R}$.
(2) $\|\hat{m}(t) - m(t)\|_{\mathcal{H}} \to 0$ almost surely.

To further elaborate on these two convergence results, (1) implies that controlling swarm patterns can indeed be carried out in the moment latent space by controlling sample moment sequences. In addition, because of the one-to-one correspondence between swarm patterns and moment sequences, (2) enables the use of the RKHS norm as a measure to evaluate pattern control performance. Of course, in practice, we can only compute moment sequences up to a finite order. In the sequel, and with a slight abuse of notation, $m(t)$ and $\hat{m}(t)$ will denote the ensemble moment and sample moment sequences up to a finite order, respectively.

B. Moment latent reinforcement learning

As mentioned in the previous section, the robots in the swarm being tracked are randomly selected. This stochasticity is reflected in the sample moment sequence as uncertainty, meaning different collections of observed robots may result in different sample moment sequences. This also suggests RL as an appropriate tool for learning a pattern control policy in the uncertain moment latent environment.

1) Episodic latent space-workspace interaction: The application of RL to the robot swarm faces the challenge that different collections of robots are measured at different time. This is because the policy learned using the sample moment sequence generated by one collection of robots may not perform well for other robots whose workspace locations are not detected by the sensors. This issue arises particularly when the training collection does not contain enough robots, resulting in a large error between the moment and sample moment sequences, $\|\hat{m}(t) - m(t)\|_{\mathcal{H}}$. To overcome this issue, data in different episodes are collected from different sets of robots. This ensures that an updated policy is always applied to the entire robot swarm, rather than only to the moment system derived from the sample moment sequence in (5), Consequently, the latent state trajectory in the successive

training loop is generated from robots steered by the latest policy, which warrants an unbiased training process. It is also worth pointing out that the episodic interaction between the workspace and latent space, as shown in Fig. 1, distinguishes the proposed RL structure from existing latent space planning algorithms.

2) Moment latent agent design: To enable the proposed episodic interaction between the moment latent space and workspace, for each training episode, say the i^{th} one, we draw $n^{(i)}$ random samples $\Omega^{(i)} = \{\beta_1^{(i)}, \dots, \beta_{n^{(i)}}^{(i)}\}$ from the system parameter distribution λ. Then, we collect the workspace location trajectories $\{X_{t_0}(\beta_j^{(i)}), \dots, X_{t_N}(\beta_j^i)\}$, $j = 1, \dots, n^{(i)}$ of the robots whose system parameters are in $\Omega^{(i)}$. The data are input to the moment latent space in real-time to compute the discretized moment system using (5) as

$$\hat{m}_k^{(i)}(t_{l+1}) = \hat{m}_k^{(i)}(t_l) + \tau F_k(X_{t_l}(\beta_1^{(i)}), \dots, X_{t_l}(\beta_{n^{(i)}}^{(i)}), u(t), v(t)), \quad (6)$$

where $1/\tau$ is the sampling rate and $|k|$ ranges from 0 to a finite order M. Note that $\Omega^{(i)} \neq \Omega^{(j)}$ holds with probability 1, ensuring that moment systems computed in different episodes are also different. This, in turn, demonstrates the need for multiple episodes enabled by the latent space and workspace interaction to guarantee an unbiased training process.

The computed moment system in (6) serves as the RL agent, with the reward $R_{t_l} = \|\hat{m}(t_l) - m_F\|_{\mathcal{H}}^2$, where m_F is the moment sequence of the desired final pattern. Given a control policy $\pi = (u, v)$, the value function is then given by

$$V_\pi(z) = \mathbb{E}\Big[\sum_{k=l}^{\infty} \gamma^k R_{t_k} \;\Big|\; \hat{m}(t_l) = z\Big] = \mathbb{E}\Big[\sum_{k=l}^{\infty} \gamma^k \|\hat{m}(t_k) - m_F\|_{\mathcal{H}}^2 \;\Big|\; \hat{m}(t_l) = z\Big], \quad (7)$$

where $\mathbb{E}$ is the expectation with respect to the law of the stochastic process $\hat{m}(t)$ and $\gamma \in (0, 1]$ is the discount factor. When $\gamma = 1$, $\lim_{k\to\infty} \|\hat{m}(t_k) - m_F\|_{\mathcal{H}} = 0$ is necessary to guarantee the convergence of $\sum_{k=l}^{\infty} \gamma^k \|\hat{m}(t_l) - m_F\|_{\mathcal{H}}^2$, ensuring that the robot swarm will be asymptotically steered to the desired pattern. Therefore, to guarantee the pattern control performance, γ is chosen to be close to 1.

Remark 1: This moment latent swarm pattern control framework can also be refined to mitigate the number of collisions between robots in the swarm. The idea is to increase the variance of the distribution μ_t by penalizing its moments $m_k(t)$ of the total order $|k| = 2$, ensuring that each pair of robots in the swarm has a larger expected separation. In this case, the value function in (7) becomes

$$V_\pi(z) = \mathbb{E}\Big[\sum_{k=l}^{\infty} \gamma^k \big(\|\hat{m}(t_k) - m_F\|_{\mathcal{H}}^2 - p \sum_{|r|=2} |m_r(t)|^2\big) \Big| \; \hat{m}(t_l) = z\Big],$$

where $p > 0$ is a penalization coefficient. Note that p needs to be carefully designed to balance the control performance and the number of collisions.

V. Examples and Simulations

In this section, we demonstrate the performance and efficiency of the proposed moment latent RL architecture using diverse simulated pattern control tasks. The simulations were conducted using both numerical and ROS2-based TurtleBot3 swarms in Gazebo simulators [43], [44], [45].

A. Reinforcement learning environment setup

In both numerical and ROS2-Gazebo simulations, we implement the actor-critic structured Proximal Policy Optimization (PPO) to learn the pattern control policies. Specifically, both the actor and critic networks are three layer perceptrons, the advantage estimation parameter and clipping range in PPO are set to 0.95 and 0.2, respectively, and the discount factor in the objective function is chosen to be $\gamma = 0.995$. The training process is composed of 1000 episodes, each of which is a 200 snapshot time series. In particular, the time series is generated by the discretized moment system in (6) learned from 30 randomly robots in the ensemble in (1), with the total time $T = 10$ and the sample time $\tau = 0.05$.

B. Numerical simulations

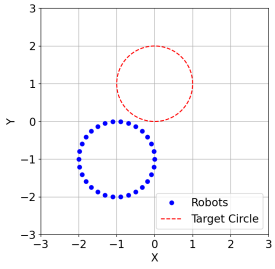
In this section, we report the numerical simulation results of four pattern control tasks. Note that we indeed considered the infinite robot swarm, given by (1), with the system parameter β taking all the values in [0.8,1.2], while the initial and final patterns shown in Figs. 2a and 2b to Figs. 5a and 5b are demonstrated using 30 robots in the swarm.

1) Circle to circle: The initial and final patterns were chosen to be the uniform distributions on the unit circles centered at $(-1, 1)$ and $(0, 1)$, respectively. We then applied the developed moment latent RL to the moment system up to total order 4, and the simulation results are shown in Fig. 2.

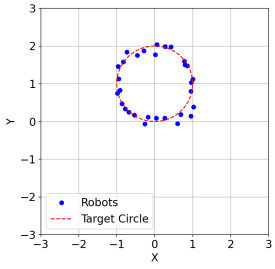
2) Line to one cluster: In this example, the initial pattern was the uniform distribution on the line segment $\{(x, y) : -2 \leq x \leq -1,\ y = 1\}$ and the final one was the uniform distribution on the unit disk centered at $(1, 1)$. We used the moment kernelization up to total order 6, and the simulation result is shown in Fig. 3.

3) One cluster to two clusters: In this case, we picked the initial pattern to be the unit disk-shaped uniform distribution centered at $(0, -2)$, and the final one was a two-cluster pattern, given by the uniform distribution on the disjoint union of the unit disks centered at $(-0.5, 0.5)$ and $(0.5, 1.5)$. The total order of the moment kernelization was 4, and the simulation result is shown in Fig. 4.

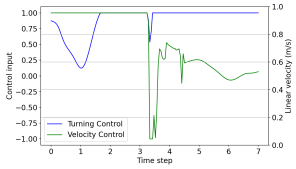
4) Line to rectangle: In this last example, we performed the formation of the rectangular pattern, given by the uniform distribution on the rectangle $[0.8, 0] \times [1.2, 1.5]$, from the line segment $[-1.2, 1.8] \times \{-1\}$, where the moment kernelization order was chosen to be 6. The simulation result is shown in Fig. 5.



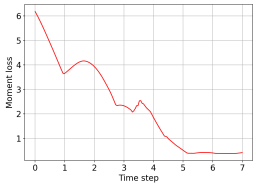
(a) Initial circular pattern



(b) Final circular pattern

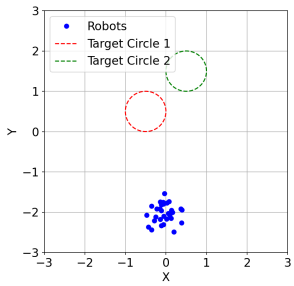


(c) Control input

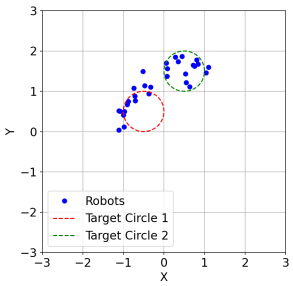


(d) Instantaneous RKHS distance to the final pattern

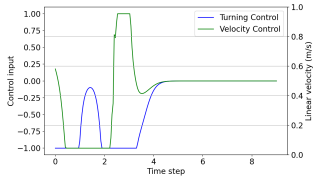
Fig. 2. Circular pattern formation.



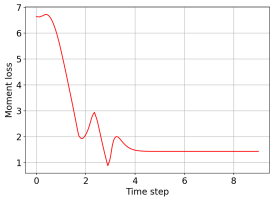
(a) Initial one cluster pattern



(b) Final two cluster pattern

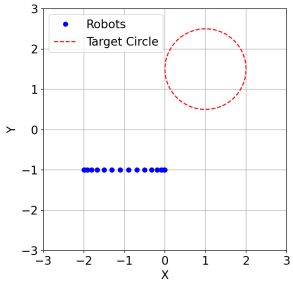


(c) Control inputs

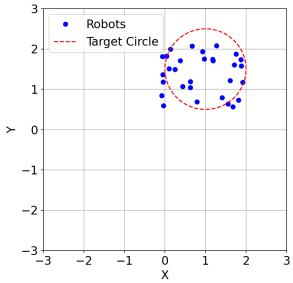


(d) Instantaneous RKHS distance to the final pattern

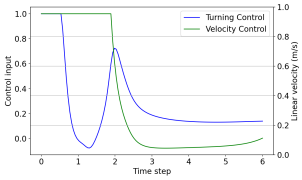
Fig. 4. One cluster to two cluster pattern formation



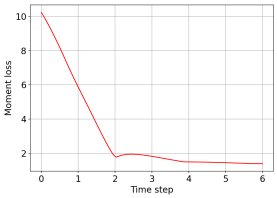
(a) Initial linear pattern



(b) Final disk-shaped pattern

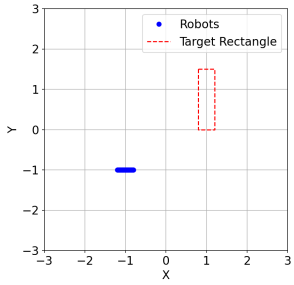


(c) Control inputs

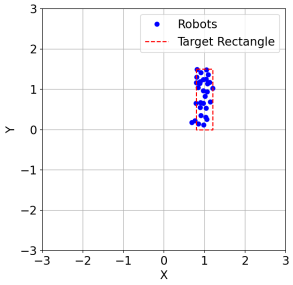


(d) Instantaneous RKHS distance to the final pattern

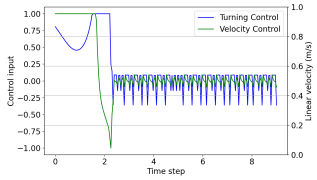
Fig. 3. Line to one cluster formation



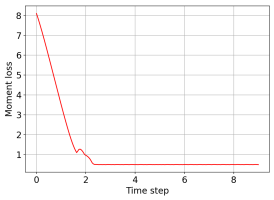
(a) Initial line segment



(b) Final rectangular pattern



(c) Control inputs



(d) Instantaneous RKHS distance to the final pattern

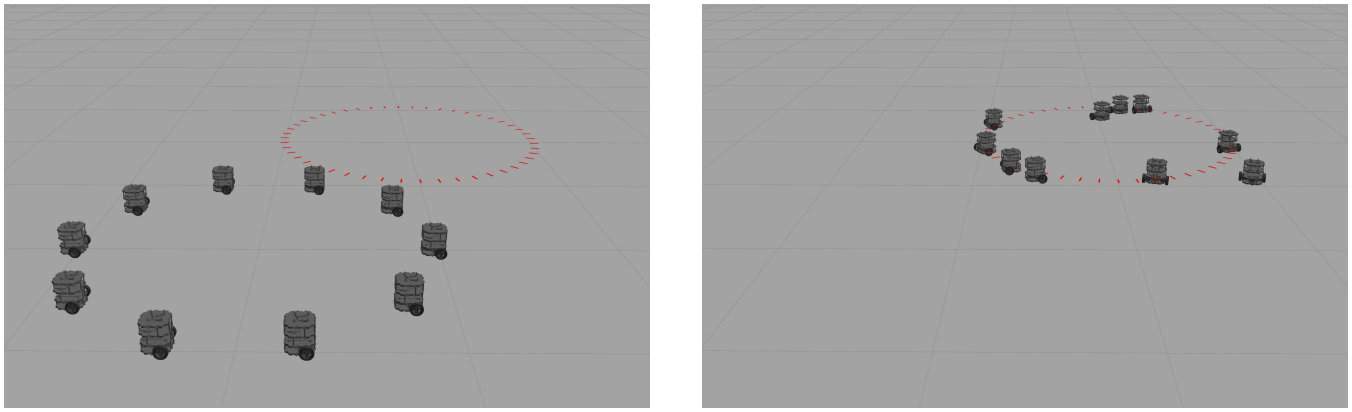
Fig. 5. Linear to rectangular pattern formation

C. TurtleBot Simulation

To demonstrate the real-world applicability of the proposed moment latent RL-enabled pattern control framework, we applied the control inputs learned in the numerical examples in Sections V-B.1 (circle to circle formation) and V-B.2 (line to one cluster formation) to TurtleBot3 swarms in Gazebo simulator. In both cases, we constructed 10 TurtleBot3 whose wheel radii were randomly drawn from $[0.8, 1.2]$. The simulation results are shown in Figs. 6 and 7.

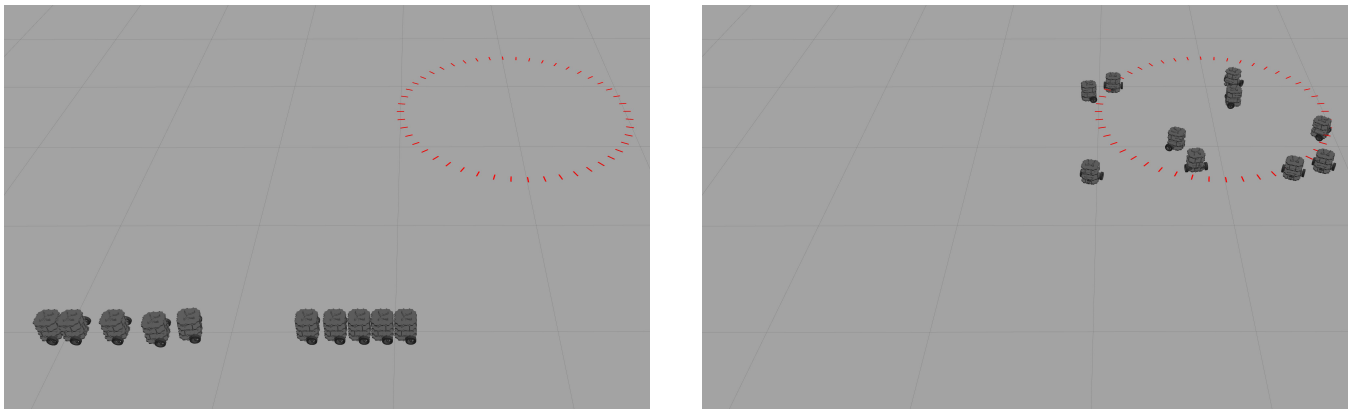
VI. CONCLUSIONS

In this paper, we develop a moment latent reinforcement learning architecture for pattern control of swarm robotic systems. In particular, our method can accommodate for arbitrarily large, in the limit infinite, robot swarms, and require the measurement data for partial robots in the swarms. In particular, we model such a robot swarm as a parameterized control system and characterize its patterns in terms of probability measures. We then introduce the moment kernel



(a) Initial circular pattern (b) Final circular pattern

Fig. 6. Circular pattern formation for TurtleBot3 in ROS2-based Gazebo simulator.



(a) Initial linear pattern (b) Final one-cluster pattern

Fig. 7. Single cluster pattern formation for TurtleBot3 in ROS2-based Gazebo simulator.

transform, generating a reduced latent space representation of the swarm dynamics over a reproducing kernel Hilbert space, which can be learned from partial measurements of the swarm. We then train a PPO agent on the moment latent space to learn the optimal pattern control policies. In the training phase, the data is episodically exchanged between the latent space and robot workspace, which ensures an unbiased learning process with high training efficiency. The proposed robot swarm pattern control framework achieves excellent performance in both numerical simulations and TurtleBot3 swarms in Gazebo simulator.

REFERENCES

[1] D. Slaughter, D. Giles, and D. Downey, "Autonomous robotic weed control systems: A review," *Computers and Electronics in Agriculture*, vol. 61, no. 1, pp. 63–78, 2008, emerging Technologies For Real-time and Integrated Agriculture Decisions.

[2] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, "Monitoring and mapping with robot swarms for agricultural applications," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6.

[3] C. Lytridis, V. G. Kaburlasos, T. Pachidis, M. Manios, E. Vrochidou, T. Kalampokas, and S. Chatzistamatis, "An overview of cooperative robotics in agriculture," *Agronomy*, vol. 11, no. 9, 2021.

[4] W. Zhang, Z. Miao, N. Li, C. He, and T. Sun, "Review of current robotic approaches for precision weed management," *Current Robotics Reports*, vol. 3, no. 3, pp. 139–151, 2022.

[5] E. Karunathilake, A. T. Le, S. Heo, Y. S. Chung, and S. Mansoor, "The path to smart farming: Innovations and opportunities in precision agriculture," *Agriculture*, vol. 13, no. 8, p. 1593, 2023.

[6] M. P. Kummer, J. J. Abbott, B. E. Kratochvil, R. Borer, A. Sengul, and B. J. Nelson, "Octomag: An electromagnetic system for 5-dof wireless micromanipulation," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1006–1017, 2010.

[7] J. Li, X. Li, T. Luo, R. Wang, C. Liu, S. Chen, D. Li, J. Yue, S. han Cheng, and D. Sun, "Development of a magnetic microrobot for carrying and delivering targeted cells," *Science Robotics*, vol. 3, no. 19, p. eaat8829, 2018.

[8] F. Soto, J. Wang, R. Ahmed, and U. Demirci, "Medical micro/nanorobots in precision medicine," *Advanced Science*, vol. 7, no. 21, p. 2002203, 2020.

[9] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang, and Q. Fu, "Reconnaissance mission conducted by uav swarms based on distributed pso path planning algorithms," *IEEE Access*, vol. 7, pp. 105 086–105 099, 2019.

[10] D. Hougen, S. Benjaafar, J. Bonney, J. Budenske, M. Dvorak, M. Gini, H. French, D. Krantz, P. Li, F. Malver, B. Nelson, N. Papanikolopoulos, P. Rybski, S. Stoeter, R. Voyles, and K. Yesin, "A miniature robotic system for reconnaissance and surveillance," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 501–507.

[11] Y. Tan and Z. yang Zheng, "Research advance in swarm robotics," *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013.

[12] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013. [Online]. Available: https://doi.org/10.1007/s11721-012-0075-2

[13] L. Bayındır, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[14] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[15] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 128–133.

[16] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[17] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[18] Z. Lu, T. Zhou, and S. Mou, "Real-time multi-robot mission planning in cluttered environment," *Robotics*, vol. 13, no. 3, 2024.

[19] T. Zhou, Z. Lu, and S. Mou, "Multi-robot formation control with human-on-the-loop," in *2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2024, pp. 1–6.

[20] Z. Liu, B. Wu, and H. Lin, "A mean field game approach to swarming robots control," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 4293–4298.

[21] K. Elamvazhuthi and S. Berman, "Mean-field models in swarm robotics: a survey," *Bioinspiration & Biomimetics*, vol. 15, no. 1, p. 015001, nov 2019. [Online]. Available: https://dx.doi.org/10.1088/1748-3190/ab49a4

[22] T. Zheng, Q. Han, and H. Lin, "Transporting robotic swarms via mean-field feedback control," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4170–4177, 2022.

[23] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-organizing Machines*, ser. A Bradford book. MIT Press, 2000.

[24] V. Trianni, *Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots*, ser. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2008.

[25] G. Francesca and M. Birattari, "Automatic design of robot swarms: Achievements and challenges," *Frontiers in Robotics and AI*, vol. 3, 2016.

[26] E. Yang and D. Gu, "Multiagent reinforcement learning for multi-robot systems: A survey," Tech. Rep., 2004.

[27] J. Orr and A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: A survey," *Sensors*, vol. 23, no. 7, 2023.

[28] R. Sutton and A. Barto, *Reinforcement Learning, second edition: An Introduction*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[29] X. Lan, Y. Liu, and Z. Zhao, "Cooperative control for swarming systems based on reinforcement learning in unknown dynamic environment," *Neurocomputing*, vol. 410, pp. 410–418, 2020.

[30] P. Zhu, W. Dai, W. Yao, J. Ma, Z. Zeng, and H. Lu, "Multi-robot flocking control based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 150 397–150 406, 2020.

[31] C. Sun, M. Shen, and J. P. How, "Scaling up multiagent reinforcement learning for robotic systems: Learn an adaptive sparse communication graph," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 755–11 762.

[32] C. Sun, D.-K. Kim, and J. P. How, "Fisar: Forward invariant safe reinforcement learning with a deep neural network-based optimizer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 617–10 624.

[33] ——, "Romax: Certifiably robust deep multiagent reinforcement learning via convex relaxation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5503–5510.

[34] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2009.

[35] Y. Abu-Mostafa, M. Magdon-Ismail, and H. Lin, *Learning from Data: A Short Course*. AMLBook.com, 2012.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2016.

[37] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter, "Analysis of the scalability of uwb indoor localization solutions for high user densities," *Sensors*, vol. 18, no. 6, 2018.

[38] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer, "Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1460–1467.

[39] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative multi-robot control for target tracking with onboard sensing1," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1660–1677, 2015.

[40] J.-S. Li and W. Zhang, "Distributional control of ensemble systems," 2025.

[41] V. Paulsen and M. Raghupathi, *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, ser. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.

[42] P. Billingsley, *Probability and Measure*, 3rd ed., ser. Wiley Series in Probability and Statistics. Wiley, 1995, vol. 245.

[43] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[44] E. Guizzo and E. Ackerman, "The turtlebot3 teacher," *IEEE Spectrum*, vol. 54, no. 8, pp. 19–20, 2017.

[45] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.