2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Learning Rigidity-based Flocking Control using Gaussian Processes with Probabilistic Stability Guarantees

Thomas Beckers[1], George J. Pappas[1], Leonardo J. Colombo[2]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992465

***Abstract*— Flocking control of multi-agents system is challenging for agents with partially unknown dynamics. These dynamics typically occur due to unfavorable nonlinearities such as friction or production inaccuracies. This paper proposes an online learning-based controller to stabilize flocking motion of double-integrator agents with additional state-dependent, unknown dynamics by using a Gaussian process (GP). Agents interaction is described by a time-invariant infinitesimally minimally rigid undirected graph. We provide a decentralized control law that exponentially stabilizes the motion of the agents and captures Reynold boids motion for swarms by using GPs as an online learning-based oracle for the prediction of the unknown dynamics. In particular the presented approach guarantees a probabilistic bounded tracking error with high probability.**

I. Introduction

Flocking, swarming, and schooling are common emergent collective motion behaviors exhibited in nature. These natural collective behaviors can be leveraged in multirobot systems to safely transport large cohesive groups of robots within a workspace. To capture these effects, Reynolds introduced three heuristic rules: cohesion; alignment; and separation, to reproduce flocking motions in computer graphics in [1]. Decentralized flocking control algorithms based on Reynolds's rules have gained much attention in the recent years due to the increasing amount of mobile and aerial robotic swarms [2], [3], [4].

Decentralized flocking stabilization techniques can be based on artificial potential fields [2], negative gradient of potential functions [5] and rigidity theory [3], [4]. For the latter approach, most of techniques to stabilize a set of agents to a desired rigid shape are given for single integrator agents [6]. Double integrator models have been extensively studied for flocking control since [7] and [2]. In particular rigidity-based flocking control has been studied in [3] and [4], but non of them consider uncertainties in the model.

The design of safe controllers for robotic swarms is a substantial aspect for an increasing range of application domains. However, parts of the robot's dynamics and external disturbances are often unknown or very time-consuming to model, e.g., wind fields in drone control or velocity dependent friction. To overcome the issue of unknown dynamics, learning-based control laws have been proposed but they are limited to iterative learning [8], leader-follower formations [9], [10], [11] or imitation [12]. We present a decentralized controller for agents with second-order dynamics by using Gaussian Processes, for an online learning of the unknown dynamics, while agents exhibit Reynolds rules of flocking. The presented learning-based controller guarantees a probabilistic bounded error to desired flocking motions with high probability and where the bound can be explicitly given by solving an optimization problem over the data-set. The online learning approach allows to improve the model and, thus, the stability performance during run-time.

Decentralized learning-based control algorithms for multi-agent systems has been recently studied in [13] using graph neural networks and also in [14], [15]. In particular, recent learning-based methods for flocking control of second-order agents can be found in [16], [12], [17], but none of them includes 3D flocking control with stability guarantee in the performance. In detail, there are no available results for the design of a learning-based flocking control law for double integrator agents under partially unknown dynamics, based on online learning data-driven models with exponential stability guarantees.

In data-driven control, data of the unknown system dynamics is collected and used to predict the dynamics in areas without training data. In contrast to parametric models, those models are highly flexible and are able to reproduce a large class of different dynamics [18]. Recently, Gaussian process (GP) models [19] has been increasingly used for modeling dynamical system due to many beneficial properties such as the bias-variance trade-off and the strong connection to Bayesian statistics. In contrast to many other techniques, GP models provide not only a prediction but also a measure for the uncertainty of the model. This powerful property makes them very attractive for many applications in control, e.g., model predictive control [20], feedback linearization [21], and tracking control [22], as the uncertainty measure allows to provide performance and safety guarantees. The purpose of this article is to employ the power of learning-based approaches, in particular, GPs, for decentralized flocking control of second-order agents with partially unknown dynamics guaranteeing the probabilistic boundedness of the error to desired flocking motions, with high probability.

The remaining article is as follows: we introduce rigid and connected graphs on Section II. After the problem setting

[1] T. Beckers and G. Pappas are with Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA {tbeckers, pappasg}@seas.upenn.edu,

[2] L. Colombo is with the Centre for Automation and Robotics (CSIC-UPM), Ctra. M300 Campo Real, Km 0,200, Arganda del Rey - 28500 Madrid, Spain, leonardo.colombo@car.upm-csic.es. This work was supported by a 2020 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation and by the AFOSR under grant FA9550-19-1-0265 (Assured Autonomy in Contested Environments.The BBVA Foundation accepts no responsibility for the opinions, statements and contents included in the project and/or the results thereof, which are entirely the responsibility of the authors.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.

in Section III, the online learning-based controller is given in Section IV. Numerical examples are shown in Section V.

II. Agents in a network

Consider $n \geq 2$ autonomous agents whose positions are denoted by $\boldsymbol{q}_i \in \mathbb{R}^d$, $d = \{2, 3\}$ and denote by $\boldsymbol{q} \in \mathbb{R}^{dn}$ the stacked vector of agents' positions[1]. Neighbor's relationships are described by an undirected and time-invariant graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$ with the ordered edge set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$. The set of neighbors for $i \in \mathcal{N}$, is defined by $\mathcal{N}_i := \{j \in \mathcal{N} : (i,j) \in \mathcal{E}\}$. Agents can sense the relative positions of its nearest neighbors, in particular, agents can measure its Euclidean distance from other agents in the subset $\mathcal{N}_i \subseteq \mathcal{N}$. We denote the incidence matrix $B \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{E}|}$ that establish the neighbors' relationships for $\mathbb{G}$, and by $\mathcal{E}_k^{\text{head}}$ and $\mathcal{E}_k^{\text{tail}}$ the head and tail nodes, respectively, of the edge $\mathcal{E}_k$, i.e., $\mathcal{E}_k = (\mathcal{E}_k^{\text{head}}, \mathcal{E}_k^{\text{tail}})$ for $k = 1, \ldots, |\mathcal{E}|$. The stacked vector of relative positions between neighboring agents, denoted by $\boldsymbol{z} \in \mathbb{R}^{d|\mathcal{N}|}$, is given by $\boldsymbol{z} = \overline{B}^T \boldsymbol{q}$, where $\overline{B} := B \otimes I_d \in \mathbb{R}^{d|\mathcal{N}|\times d|\mathcal{E}|}$ with I_d being the $(d \times d)$ identity matrix, and $\otimes$ the Kronecker product. Note that $\boldsymbol{q}_k \in \mathbb{R}^d$ in $\boldsymbol{q}$ corresponds to $\boldsymbol{q}_i - \boldsymbol{q}_j$ for the edge $\mathcal{E}_k$. We define $\boldsymbol{z}_k := \boldsymbol{q}_i - \boldsymbol{q}_j$ to simplify notation. A *framework* for $\mathbb{G}$ is defined as the pair $(\mathbb{G}, \boldsymbol{q})$. Flocking control can be achieved by means of the Laplacian matrix associated with $\mathbb{G}$. The Laplacian matrix L is the matrix whose entries are given by $l_{ij} = -1$ for $i \neq j$, if there is an edge between agents j and i, else $l_{ij} = 0$. Moreover, $l_{ii} = -\sum_{j \in \mathcal{N}_i} l_{ij}$. In the case of $\mathbb{G}$ being an undirected graph, it follows that $L = BB^T$ (see [23] for instance). For a connected and undirected graph there holds $\text{rank}(L) = |\mathcal{N}| - 1$ and $\text{null}(L) = \text{null}(B) = \{\mathbf{1}_{|\mathcal{N}|}\}$. Therefore, $\text{null}(L) = \text{null}(B) = \{\mathbf{1}_{|\mathcal{N}|} \otimes I_{d\times d}\}$. Thus L is positive definite by restricting it to the vector space of $\text{span}(\mathbf{1}_{|\mathcal{N}|} \otimes I_{d\times d})^{\perp}$ and the number of connected components in the graph equals the algebraic multiplicity of its null eigenvalue. Besides, the second smallest eigenvalue of L, denoted by λ_2, is known as algebraic connectivity of $\mathbb{G}$, as it is related with the interconnection of the nodes.

To guarantee convergence to desired shapes we consider the *rigidity* of the desired formation shapes. The *rigidity matrix* for the framework $(\mathbb{G}, \boldsymbol{q})$ is defined as, see [24], $R(\boldsymbol{z}) = \frac{1}{2}\frac{\partial \ell_{\mathbb{G}}(\boldsymbol{q})}{\partial \boldsymbol{q}} = D(\boldsymbol{z})^{\top}\overline{B} \in \mathbb{R}^{|\mathcal{E}|\times d|\mathcal{N}|}$, with $D(\boldsymbol{z}) = \text{diag}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{|\mathcal{E}|}) \in \mathbb{R}^{d|\mathcal{E}|\times|\mathcal{E}|}$ and distance measure function $\ell_{\mathbb{G}} : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}^{|\mathcal{E}|}$ defined by $\ell_{\mathbb{G}}(\boldsymbol{q}) = \left(\left\|\boldsymbol{q}_i - \boldsymbol{q}_j\right\|^2\right)_{(i,j)\in\mathcal{E}} = D^{\top}(\boldsymbol{z})\boldsymbol{z}$. Denote the desired distance between neighboring agents over the edge $\mathcal{E}_k$ as d_k and define the squared distance error for $\mathcal{E}_k$ as $\boldsymbol{e}_k = \left\|\boldsymbol{q}_i - \boldsymbol{q}_j\right\|^2 - d_k^2 = \|\boldsymbol{z}_k\|^2 - d_k^2$, with stacked squared distance vector error denoted by $\boldsymbol{e} = \left[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{|\mathcal{E}|}\right]^{\top}$.

[1]Matrices are denoted with capital letters. $\mathcal{N}(\mu, \Sigma)$ describes a normal distribution with mean μ and covariance Σ. The probability function is denoted by P. $\mathbb{E}[X]$ denotes the expected value of a random variable X. $\mathbb{R}_{>0}$ denotes the set of positive real numbers. The Euclidean norm is denoted by $||\cdot||$ and by $|\mathcal{X}|$ the cardinal of the set $\mathcal{X}$.

A framework $(\mathbb{G}, \boldsymbol{q})$ is said to be *rigid* if it is not possible to smoothly move one node of the framework without moving the rest while maintaining the inter-agent distance given by $\ell_{\mathbb{G}}(\boldsymbol{q})$, see [24]. An *infinitesimally rigid* framework is a rigid framework which is invariant under and only under infinitesimally transformations under $R(\boldsymbol{z})$, i.e., $\ell_{\mathbb{G}}(\boldsymbol{q} + \delta\boldsymbol{q}) = \ell_{\mathbb{G}}(\boldsymbol{q})$ where $\delta\boldsymbol{q}$ denotes an infinitesimal displacement of $\boldsymbol{q}$. It is well known (see [24] for instance) that a framework $(\mathbb{G}, \boldsymbol{q})$ is infinitesimally rigid in $\mathbb{R}^d$ if $\boldsymbol{q}$ is a regular value of $\ell_{\mathbb{G}}(\boldsymbol{q})$ and $(\mathbb{G}, \boldsymbol{q})$ is rigid in $\mathbb{R}^d$. In particular, $(\mathbb{G}, \boldsymbol{q})$ is *infinitesimally rigid* in $\mathbb{R}^2$ if $\text{rank}\, R(\boldsymbol{z}) = 2n - 3$ (respectively, $\text{rank}\, R(\boldsymbol{z}) = 3n - 6$ in $\mathbb{R}^3$).

The framework $(\mathbb{G}, \boldsymbol{q})$ is said to be *minimally rigid* if it has exactly $2n-3$ edges in $\mathbb{R}^2$ or $3n-6$ edges in $\mathbb{R}^3$. This means that if we remove one edge from a minimally rigid framework $(\mathbb{G}, \boldsymbol{q})$, then it is not rigid anymore. Thus, the only motions over the agents in a minimally rigid framework, while they are already in the desired shape, are the ones defining translations and rotations of the whole shape, see [6]. One important property for the stabilization to desired motions in shape control with flocking behaviour [3] is that the rigidity matrix $R(\boldsymbol{z})$ has full row rank if $(\mathbb{G}, \boldsymbol{q})$ is minimally and infinitesimally rigid.

III. Modeling flocking control of double integrator agents with Gaussian Processes

Consider the set $\mathcal{N}$ consisting of $n \geq 2$ free autonomous agents evolving on $\mathbb{R}^d$, $d = \{2, 3\}$, as in Section II, under a double integrator dynamics, that is $\begin{cases} \dot{\boldsymbol{q}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \boldsymbol{u}. \end{cases}$

According to the Reynolds flocking model [1], the motion of every agent in the flock is defined by the three rules of *alignment, cohesion and separation*. Cohesion and separation might be achieved by using artificial potential fields [2]. We are interested on shape control with flocking motion, that is, agents reach a desired formation shape and they also move along the 2D plane or 3D space by achieving a consensus on their velocities. To exponentially achieve this collective behaviour, and provide their convergence rates, similarly as in [3] and [4], we make the following assumption

Assumption 1 $(\mathbb{G}, \boldsymbol{q})$ *is an infinitesimally and minimally rigid framework with* $\mathbb{G}$ *undirected, static and connected.*

To reach a desired shape, for each edge $\mathcal{E}_k = (i, j)$ in the infinitesimally and minimally rigid framework we introduce the artificial potential functions $V_k : \mathbb{R}^d \to \mathbb{R}$, given by $V_k(\boldsymbol{z}_k) = \frac{1}{4}(||\boldsymbol{z}_k||^2 - d_k^2)^2$, to provide a measure for the interaction between agents and their nearest neighbors (see [6] for a detailed discussion on the choices of elastic potential functions). In these potentials, $\boldsymbol{z}_k$ denotes the relative position between agents for the edge $\mathcal{E}_k$, and d_k denotes the desired length for the edge $\mathcal{E}_k$.

The acceleration of agent i is determined by

$$\boldsymbol{u}_i(t) = \underbrace{-l_{ij}\sum_{j\in\mathcal{N}_i}(\boldsymbol{v}_i - \boldsymbol{v}_j)}_{\text{alignment}} - \underbrace{\sum_{j\in\mathcal{N}_i}\nabla_{\boldsymbol{q}_i}V_k}_{\text{cohesion + separation}},$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.

that is, for $\mathcal{L} = L \otimes I_d$, then $\boldsymbol{u}(t) = -\mathcal{L}\boldsymbol{v} - R^T(\boldsymbol{z})e(\boldsymbol{z})$.

The closed loop system, called *double integrator flocking stabilization system* [3], [4] is given by

$$\begin{cases} \dot{\boldsymbol{q}}_i = \boldsymbol{v}_i \\ \dot{\boldsymbol{v}}_i = -l_{ij} \sum\limits_{j \in \mathcal{N}_i} (\boldsymbol{v}_i - \boldsymbol{v}_j) - \sum\limits_{j \in \mathcal{N}_i} \nabla_{\boldsymbol{q}_i} V_k. \end{cases} \tag{1}$$

Therefore, we can define the artificial potential function $V_0 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$ for the overall networked control system as $V_0(\boldsymbol{z}) = \sum_{k=1}^{|\mathcal{E}|} V_k(||\boldsymbol{z}_k||)$. In order to control the velocity of the agents, we introduce the disagreement vector $\boldsymbol{\delta} \in \mathbb{R}^{d|\mathcal{N}|}$. Consider the potential function $V_1 : \mathbb{R}^{d|\mathcal{N}|} \to \mathbb{R}$ defined as $V_1(\boldsymbol{\delta}) = \frac{1}{2}\sum_{i=1}^{|\mathcal{N}|} ||\boldsymbol{\delta}_k||^2$, $\boldsymbol{\delta}_k = \boldsymbol{\delta}_i - \boldsymbol{\delta}_j$, where $\boldsymbol{\delta} = [\boldsymbol{\delta}_1^T, \ldots, \boldsymbol{\delta}_{|\mathcal{N}|}^T]^T$ is the velocity disagreement vector where each component $\boldsymbol{\delta}_i$ is given by $\boldsymbol{\delta}_i = \boldsymbol{v}_i - \bar{\boldsymbol{v}}$ with $\bar{\boldsymbol{v}}(t) := \dfrac{1}{|\mathcal{N}|}\sum\limits_{i=1}^{|\mathcal{N}|} \boldsymbol{v}_i(t) \in \mathbb{R}^d$ denoting the average velocity of the agents. Note that $\dot{\bar{\boldsymbol{v}}}(t) = 0$ and hence $\dot{\boldsymbol{\delta}}_i = \dot{\boldsymbol{v}}_i$, since $\bar{\boldsymbol{v}}$ is constant.

By considering the semi-definite function $V := V_0 + V_1$ as energy function for the double integrator flocking stabilization system (1), under Assumption 1, one can show local exponential convergence of the agents to desired shapes with flocking motion behaviour [3], [4].

Next, consider each agent $i \in \{1, \ldots, |\mathcal{N}|\}$ disturbed by an additive unknown dynamics given by

$$\begin{cases} \dot{\boldsymbol{q}}_i = \boldsymbol{v}_i, \\ \dot{\boldsymbol{v}}_i = \boldsymbol{u}_i + \boldsymbol{f}_i(\boldsymbol{q}_i, \boldsymbol{v}_i), \end{cases} \tag{2}$$

where $\boldsymbol{f}_i : \mathbb{R}^{2d} \to \mathbb{R}^d$ is a state-dependent unknown function. Potential source of $\boldsymbol{f}_i$ are wind fields in drone control, unmodeled nonlinearities such as friction or inhomogeneity in the agents' dynamics due to production inaccuracies. The time dependency of the states is omitted for simplicity of notation and the time dependency of the unknown input forces $\boldsymbol{f}_i$ might be also indirect, i.e. $\boldsymbol{f}_i(\boldsymbol{q}_i(t), \boldsymbol{v}_i(t))$.

In the following, we propose an online learning strategy for flocking motion, and an upper probabilistic bound for the error estimation between the learned and the true dynamics (i.e., the mean prediction of the GP). In preparation for the learning and control step, we introduce the estimate $\hat{\boldsymbol{f}}_i \colon \mathbb{R}^{2d} \to \mathbb{R}^d$ which can include existing prior knowledge about the unknown dynamics $\boldsymbol{f}_i$, e.g, using classical system identification modeling [25]. However, if no prior knowledge is available, the estimate $\hat{\boldsymbol{f}}_i$ is set to zero. Then, (2) can be written as

$$\begin{cases} \dot{\boldsymbol{q}} = \boldsymbol{v}, \\ \dot{\boldsymbol{v}} = \boldsymbol{u} + \boldsymbol{\rho}(\boldsymbol{q}) + \hat{\boldsymbol{f}}(\boldsymbol{q}, \boldsymbol{v}), \end{cases} \tag{3}$$

with the stacked vector of estimating functions $\hat{\boldsymbol{f}}(\boldsymbol{q}, \boldsymbol{v}) = [\hat{\boldsymbol{f}}_1(\boldsymbol{q}_1, \boldsymbol{v}_1)^\top, \ldots, \hat{\boldsymbol{f}}_{|\mathcal{N}|}(\boldsymbol{q}_{|\mathcal{N}|}, \boldsymbol{v}_{|\mathcal{N}|})^\top]^\top$ and the unknown dynamics $\boldsymbol{\rho} \colon \mathbb{R}^{2d|\mathcal{N}|} \to \mathbb{R}^{d|\mathcal{N}|}$ with elements defined by $\boldsymbol{\rho}_i(\boldsymbol{p}_i) = \boldsymbol{f}_i(\boldsymbol{q}_i, \boldsymbol{v}_i) - \hat{\boldsymbol{f}}_i(\boldsymbol{q}_i, \boldsymbol{v}_i)$, where $\boldsymbol{p}_i = [\boldsymbol{q}_i^\top, \boldsymbol{v}_i^\top]^\top$. In the next step, we employ a GP model for the learning of the unknown dynamics $\boldsymbol{\rho}$.

Gaussian processes are stochastic processes which are completely defined by a mean function $m_{\mathrm{GP}} : \mathbb{R}^p \to \mathbb{R}$, $p \in \mathbb{N}$ and a kernel function $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. Then, for all $f_{\mathrm{GP}}(\boldsymbol{x}) \sim \mathcal{GP}\left(m_{\mathrm{GP}}(\boldsymbol{x}), k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right)$, it is verified that the mean $m_{\mathrm{GP}}(\boldsymbol{x}) = \mathbb{E}\left[f_{\mathrm{GP}}(\boldsymbol{x})\right]$ and the variance $k\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \mathbb{E}\left[\left(f_{\mathrm{GP}}(\boldsymbol{x}) - m_{\mathrm{GP}}(\boldsymbol{x})\right)\left(f_{\mathrm{GP}}\left(\boldsymbol{x}'\right) - m_{\mathrm{GP}}\left(\boldsymbol{x}'\right)\right)\right]$, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^p, p \in \mathbb{N}$. One of the main strengths of GPs is that, when combined with Bayes' Theorem, they can provide statistical inference with function regression.

Consider the output $\boldsymbol{y}$ of a function $\boldsymbol{f} : \mathbb{R}^p \to \mathbb{R}^p$. Without loss of generality, the mean function has been set to zero. Measurements may be affected by Gaussian noise such that $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 I_p)$ with the p-dimensional identity matrix I_p. The set of input data, $X = \left[\boldsymbol{x}^{\{1\}}, \boldsymbol{x}^{\{2\}}, \ldots, \boldsymbol{x}^{\{m\}}\right] \in \mathbb{R}^{p \times m}$ and measured output data, $Y = \left[\boldsymbol{y}^{\{1\}}, \boldsymbol{y}^{\{2\}}, \ldots, \boldsymbol{y}^{\{m\}}\right] \in \mathbb{R}^{p \times m}$ constitute the training set $\mathcal{D} = \{X, Y\}$. Then, for a test input $\boldsymbol{x}^* \in \mathbb{R}^p$ the predictions of $\boldsymbol{f}(\boldsymbol{x}^*)$ are obtained by conditioning on the data which leads to the posterior distribution

$$\begin{aligned} \mu\left(f_i \,|\, \boldsymbol{x}^*, \mathcal{D}\right) &= \boldsymbol{k}\left(\boldsymbol{x}^*, X\right)^\top \left(K + I\sigma^2\right)^{-1} Y_{:,i}, \\ \operatorname{var}\left(f_i \mid x^*, \mathcal{D}\right) &= k\left(\boldsymbol{x}^*, \boldsymbol{x}^*\right) - \boldsymbol{k}\left(\boldsymbol{x}^*, X\right)^\top \\ &\quad \left(K + I\sigma^2\right)^{-1} \boldsymbol{k}\left(\boldsymbol{x}^*, X\right) \end{aligned} \tag{4}$$

for all $i \in \{1, \ldots, p\}$, where $Y_{:,i}$ denotes the i-th column of the outputs matrix Y. The kernel k measures the correlation of two inputs $(\boldsymbol{x}, \boldsymbol{x}')$. The function $K \colon \mathbb{R}^{p \times m} \times \mathbb{R}^{p \times m} \to \mathbb{R}^{m \times m}$ is called the Gram matrix and its elements are given by $K_{j',j} = k(X_{:,j'}, X_{:,j}) + \delta(j, j')\sigma^2$ for all $j', j \in \{1, \ldots, m\}$ with the delta function $\delta(j, j') = 1$ for $j = j'$ and zero, otherwise. The vector-valued function $\boldsymbol{k} \colon \mathbb{R}^p \times \mathbb{R}^{p \times m} \to \mathbb{R}^m$, with elements $k_j = k(\boldsymbol{x}^*, X_{:,j})$ for all $j \in \{1, \ldots, m\}$, expresses the covariance between $\boldsymbol{x}^*$ and the input training data X. The choice of the kernel and the determination of the corresponding hyperparameters can be seen as degrees of freedom of the regression procedure. One common and powerful kernel for GP models of physical systems is the squared exponential kernel. An overview about different kernels and their properties is provided in [19]. To simplify, identical kernels are selected for each output dimension. Nevertheless, the GP model can be easily adapted to different kernels for each output dimension.

For our purpose, each agent collects $m(t) \in \mathbb{N}$ training points based on its own dynamics (2) such that data sets

$$\mathcal{D}_{i,m(t)} = \{\boldsymbol{p}_i^{\{j\}}, \boldsymbol{y}_i^{\{j\}}\}_{j=1}^{m(t)} \tag{5}$$

are created. The output data $\boldsymbol{y}_i \in \mathbb{R}^d$ are given by $\boldsymbol{y}_i = \dot{\boldsymbol{v}}_i - \hat{\boldsymbol{f}}_i(\boldsymbol{q}_i, \boldsymbol{v}_i) - \boldsymbol{u}_i$. The number of training points $m(t)$ of the data sets $\mathcal{D}_{i,m(t)}, i \in \{1, \ldots, |\mathcal{N}|\}$ with $m \colon \mathbb{R}_{\geq 0} \to \mathbb{N}$ can change over time t, i.e., it allows *online learning*. Let $\mathcal{D}_{m(t)} = \{\mathcal{D}_{i,m(t)}, \ldots, \mathcal{D}_{|\mathcal{N}|,m(t)}\}$ be a set that contains all training set. We introduce the following assumption on the data collection.

Assumption 2 *There are only finitely many switches of $m(t)$ over time and there exists a time $T \in \mathbb{R}_{\geq 0}$ where $\mathcal{D}_{m(T)} = \mathcal{D}_{m(t)}, \forall t \geq T, \forall i \in \{1, \ldots, |\mathcal{N}|\}$.*

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.

Remark 1 Assumption 2 ensures that the switching between the data sets is not infinitely fast which is natural in real world applications.

To model the error, an assumption has to be made about the kernel function k of the GP model.

Assumption 3 *The continuous kernel k is chosen in such a way the functions ρ_i, $i \in \{1,\dots,d|\mathcal{N}|\}$ have a bounded reproducing kernel Hilbert Space (RKHS) norm on a compact set $\Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$, i.e. $\|\rho_i\|_k < \infty$ for all $i \in \{1,\dots,d|\mathcal{N}|\}$.*

Remark 2 The norm of a function in a RKHS is a smoothness measure relative to a kernel k that is uniquely connected with this RKHS. In particular, it is a Lipschitz constant with respect to the metric of the used kernel [26].

Under the previous consideration, the model error can be probabilistically bounded as written in the following lemma, which is a direct consequence of Lemma 1 given in [27].

Lemma 1 *Consider the system* (3) *and a GP model satisfying Assumptions 2 and 3. Then the model error is probabilistically bounded by*

$$\mathrm{P}\left\{\|\boldsymbol{\mu}(\boldsymbol{\rho} \mid \boldsymbol{p},\mathcal{D}_m) - \boldsymbol{\rho}(\boldsymbol{p})\| \leq \left\|\boldsymbol{\beta}^\top \Sigma^{\frac{1}{2}}(\boldsymbol{\rho} \mid \boldsymbol{p},\mathcal{D}_m)\right\|\right\} \geq \epsilon$$

for $\boldsymbol{p} \in \Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$ compact, with $\epsilon \in (0,1), \boldsymbol{\beta},\boldsymbol{\gamma} \in \mathbb{R}^d$, and denoting by m the number of entries in the data set $\mathcal{D}_m$,

$$\beta_j = \sqrt{2\left\|\rho_j\right\|_k^2 + 300\gamma_j \ln^3\left(\frac{m+1}{1-\epsilon^{1/(d|\mathcal{N}|)}}\right)}. \quad (6)$$

The variable $\gamma_j \in \mathbb{R}$ is the maximum information gain

$$\gamma_j = \max_{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}} \in \Omega} \frac{1}{2}\log\left|I + \sigma_j^{-2} K(\boldsymbol{x},\boldsymbol{x}')\right| \quad (7)$$

$$\boldsymbol{x},\boldsymbol{x}' \in \left\{\boldsymbol{p}^{\{1\}},\dots,\boldsymbol{p}^{\{m+1\}}\right\}. \quad (8)$$

Remark 3 An efficient algorithm can be used to find $\boldsymbol{\beta}$ based on the maximum information gain [28].

IV. Online learning for stable flocking control

Consider the potential function $V : \mathbb{R}^{2d|\mathcal{N}|} \to \mathbb{R}$ as described in Section III. In the absence of unknown disturbances, V allows to write the closed-loop system (1) as

$$\begin{cases} \dot{\boldsymbol{q}} = \nabla_{\boldsymbol{v}} V \\ \dot{\boldsymbol{v}} = -\nabla_{\boldsymbol{v}} V - \nabla_{\boldsymbol{q}} V. \end{cases} \quad (9)$$

Local exponential convergence to the set

$$\mathcal{S} = \{(\boldsymbol{q}^*,\boldsymbol{\delta}^*) \in \mathbb{R}^{2d|\mathcal{N}|} | \nabla_{\boldsymbol{q}} V(\boldsymbol{q}^*) = \boldsymbol{0}, \boldsymbol{\delta}^* = \boldsymbol{0}\} \quad (10)$$

for the system (9), has been shown in [3], [4]. Next, we design a decentralized data-driven control law by using GP's, such that, by learning and update the learning of the unknown disturbances, exponentially stabilizes the partially unknown motion of the agents (2) to a desired formation shape with flocking motion. We will proceed as in [29] for formation control without flocking behaviour, by analyzing an equivalent decoupled gradient systems, by employing a result from [30], that we will also use for the design of the decentralized learning-based control law, described as follow.

Consider the one-parameter family of systems with double integrator flocking stabilization dynamics $\mathcal{H}_\lambda$ given by

$$\begin{bmatrix}\dot{\boldsymbol{p}}\\ \dot{\boldsymbol{v}}\end{bmatrix} = \begin{bmatrix} -\lambda I_{d|\mathcal{N}|} & (1-\lambda) I_{d|\mathcal{N}|} \\ (\lambda-1) I_{d|\mathcal{N}|} & -\mathcal{L} I_{d|\mathcal{N}|}\end{bmatrix}\begin{bmatrix}\nabla_{\boldsymbol{p}} V \\ \nabla_{\boldsymbol{v}} V\end{bmatrix}, \quad (11)$$

where $\lambda \in [0,1]$. Equation (11) continuously interpolates all convex combinations between the dissipative system (9) for $\lambda = 0$ and a gradient system for $\lambda = 1$. The family $\mathcal{H}_\lambda$ has two important properties summarized in the following Lemma from [30].

Lemma 2 *(I) For all $\lambda \in [0,1]$, the equilibrium set of $\mathcal{H}_\lambda$ is given by the set of the critical points of the potential function V, and is independent of λ. (II) For any equilibrium of $\mathcal{H}_\lambda$ for all $\lambda \in [0,1]$, the numbers of the stable, neutral, and unstable eigenvalues of the Jacobian of $\mathcal{H}_\lambda$ are the same and independent of λ.*

Denote by $\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}} := (\boldsymbol{e},\boldsymbol{\delta})$ the stacked vector of relative positions errors and velocities disagreement vector for stabilization to desired formation shapes with flocking motion. The next theorem introduces the learning-based control law with guaranteed probabilistic boundedness of the error for stabilization under flocking motion.

Theorem 1 *Consider the system of agents* (3) *with unknown dynamics and GP models with data sets* (5) *satisfying Assumptions 2 and 3. Assume that the desired equilibrium set $\mathcal{S}$ given by* (10) *satisfies Assumption 1. Then, the control law*

$$\boldsymbol{u}(t) = -\mathcal{L}\boldsymbol{v} - R^\top(\boldsymbol{e})\boldsymbol{e}(\boldsymbol{z}) - \hat{\boldsymbol{f}}(\boldsymbol{q},\boldsymbol{v}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{p},\mathcal{D}_m) \quad (12)$$

guarantees that the solution trajectories converge locally exponentially fast to the equilibrium set $\mathcal{S}$ and are ultimately uniformly bounded in probability on Ω with $T_\epsilon \in \mathbb{R}_{\geq 0}$ by

$$\mathrm{P}\{||\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}(t)|| \leq \sqrt{2}\max_{p\in\Omega}\bar{\Delta}_{m(T)}(\boldsymbol{p}), \forall t \geq T_\epsilon\} \geq \epsilon, \quad (13)$$

where $\bar{\Delta}_m(\boldsymbol{p}) : \Omega \to \mathbb{R}_{\geq 0}$ defines an upper bound of the model error.

Remark 4 Note that the individual control law $\boldsymbol{u}_i(t)$ of each agent depends on the distance and velocity to its neighbors and the data set based on its own dynamics only.

Proof: Note that the relative position error satisfies $\dot{\boldsymbol{e}}_k = 2\boldsymbol{z}_k^T \cdot \boldsymbol{z}_k = 2D^T(\boldsymbol{z})\bar{B}(\boldsymbol{z})\boldsymbol{v}$, that is, $\dot{\boldsymbol{e}} = 2R(\boldsymbol{z})\boldsymbol{v}$. Using that $\dot{\boldsymbol{v}} = \dot{\boldsymbol{\delta}}$, the evolution for the relative position error can be written in terms of the rigidity matrix and the velocity disagreement vector as $\dot{\boldsymbol{e}} = 2R(\boldsymbol{z})\boldsymbol{\delta}$.

Denoting by $\mathcal{E}^\lambda_{\boldsymbol{e},\boldsymbol{\delta}}$ the stacked vector of errors $\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}$ from (11) for any $\lambda \in [0,1]$, which includes the closed-loop system (1) for $\lambda = 1$, we know that as a consequence of Lemma 2, $\mathcal{E}^\lambda_{\boldsymbol{e},\boldsymbol{\delta}}$ and $\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}$ share the same stability properties. By using Lemma 2, we will study the system (11) for

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.

$\lambda = 0.5$, without loss of generality, that is,

$$\dot{\boldsymbol{p}} = -\frac{1}{2}\overline{B}D(\boldsymbol{z})\boldsymbol{e} + \frac{1}{2}\boldsymbol{\delta} \tag{14}$$

$$\dot{\boldsymbol{z}} = -\frac{1}{2}\overline{B}^T\overline{B}D(\boldsymbol{z})\boldsymbol{e} + \frac{1}{2}\overline{B}^T\boldsymbol{\delta} \tag{15}$$

$$\dot{\boldsymbol{e}} = -D(\boldsymbol{z})^T\overline{B}^T\overline{B}D(\boldsymbol{z})\boldsymbol{e} + D(\boldsymbol{z})^T\overline{B}^T\boldsymbol{\delta} \tag{16}$$

$$\dot{\boldsymbol{\delta}} = -\frac{1}{2}\overline{B}D(\boldsymbol{z})\boldsymbol{e} - \mathcal{L}\boldsymbol{\delta}. \tag{17}$$

Consider the Lyapunov candidate function V for the system (3) with control law $\boldsymbol{u}(t)$ as in (12), given by the positive definite and radially unbounded function

$$V(\boldsymbol{e}, \boldsymbol{\delta}) = \frac{1}{2}||\boldsymbol{e}||^2 + ||\boldsymbol{\delta}||^2. \tag{18}$$

Note that the time-derivative of the Lyapunov function along the closed-loop trajectory satisfies

$$\begin{aligned}\dot{V} &= \boldsymbol{e}^\top\dot{\boldsymbol{e}} + 2\boldsymbol{\delta}^\top\dot{\boldsymbol{\delta}} \qquad (19)\\ &= -\begin{bmatrix}\boldsymbol{e} & \boldsymbol{\delta}\end{bmatrix}\begin{bmatrix}D(\boldsymbol{z})^\top\overline{B}^T\overline{B}D(\boldsymbol{z}) & -D(\boldsymbol{z})^\top\overline{B}^\top \\ \overline{B}D(\boldsymbol{z}) & \mathcal{L}\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \boldsymbol{\delta}\end{bmatrix}\\ &+ \boldsymbol{\delta}^\top(\boldsymbol{\rho}(\boldsymbol{p}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{p}, \mathcal{D}_m))\\ &= -\boldsymbol{e}^\top R(\boldsymbol{z})R(\boldsymbol{z})^\top\boldsymbol{e} - \boldsymbol{\delta}^\top\mathcal{L}\boldsymbol{\delta} + \boldsymbol{\delta}^\top(\boldsymbol{\rho}(\boldsymbol{p}) - \boldsymbol{\mu}(\boldsymbol{\rho}|\boldsymbol{p}, \mathcal{D}_m)).\end{aligned}$$

Denote by λ_{min} and λ_2 the minimum eigenvalue of $R(\boldsymbol{z})R^\top(\boldsymbol{z})$ and the non-zero minimum eigenvalue of $\mathcal{L}$, respectively. By Assumption 1 the rigidity matrix is full rank (except the non-generic cases, e.g., collinear or coplanar alignments of the agents in $\mathbb{R}^2$ or $\mathbb{R}^3$). Therefore $\lambda_{min} > 0$. Note also that, by Assumption 1, $\lambda_2 > 0$. Therefore, by employing Lemma 1 it follows that

$$\mathrm{P}\{\dot{V} \leq -\lambda_{\min}||\boldsymbol{e}||^2 - \lambda_2||\boldsymbol{\delta}||^2 + ||\boldsymbol{\delta}||\bar{\Delta}_m(\boldsymbol{p})\} \geq \epsilon, \tag{20}$$

where $\bar{\Delta}_m(\boldsymbol{p}) : \Omega \to \mathbb{R}_{\geq 0}$ is a bounded function such that $\|\boldsymbol{\beta}^\top\Sigma^{\frac{1}{2}}(\boldsymbol{\rho} \mid \boldsymbol{p}, \mathcal{D}_m)\| \leq \bar{\Delta}_m(\boldsymbol{p})$, which exists because the kernel function is continuous and therefore it is bounded on a compact set $\Omega \subset \mathbb{R}^{2d|\mathcal{N}|}$, and then the variance $\Sigma(\boldsymbol{\rho} \mid \boldsymbol{p}, \mathcal{D}_m)$ is bounded, see [31]. Then, the value of $\dot{V}$ is negative with probability ϵ for all $\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}$ with $||\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}|| > \max_{\boldsymbol{p}\in\Omega}\sqrt{2}\bar{\Delta}_m(\boldsymbol{p})$, where the maximum exists since $\bar{\Delta}_m(\boldsymbol{p})$ is bounded in Ω. By using Assumption 2, we define $T \in \mathbb{R}_{\geq 0}$ such that $\mathcal{D}_{m(T)} = \mathcal{D}_{m(t)}$ for all $t \geq T$. Then, V is uniformly ultimately bounded in probability by $\mathrm{P}\{||\mathcal{E}_{\boldsymbol{e},\boldsymbol{\delta}}|| \leq b, \forall t \geq T_\epsilon \in \mathbb{R}_{\geq 0}\} \geq \epsilon$ with probabilistic bound $b = \max_{\boldsymbol{p}\in\Omega}\sqrt{2}\bar{\Delta}_{m(T)}(\boldsymbol{p})$. □

V. Simulation Results

Next, we present a numerical simulation to illustrate the performance of the control law proposed in Theorem 1. We show the applicability of our method for learning flocking control of agents moving on the 3-dimensional space. We consider 4 agents forming a shape as depicted in Fig. 1.

Consider the perturbations over agents 1 and 3 given by $\boldsymbol{f}_1(\boldsymbol{v}_1) = [300\sin(0.2v_{1,y}), 300\cos(0.2v_{1,x}), 10]^\top$, $\boldsymbol{f}_3(\boldsymbol{v}_3) = [300\sin(0.2v_{3,y}), -200, 300\sin(0.2v_{3,y})]^\top$. Initial positions are $\boldsymbol{q}_1(0) = (100, 0, 0)$, $\boldsymbol{q}_2(0) = (0, 0, 200)$, $\boldsymbol{q}_3(0) = (0, -300, 0)$, $\boldsymbol{q}_4(0) = (100, 0, -300)$. Note the significant deterioration of the motion when the classical control law has to deal with the unknown forces, as observed in Figure 2.

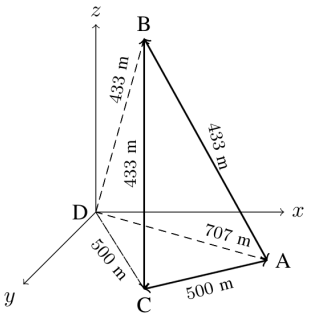


Fig. 1: Neighbor's relations and desired shape

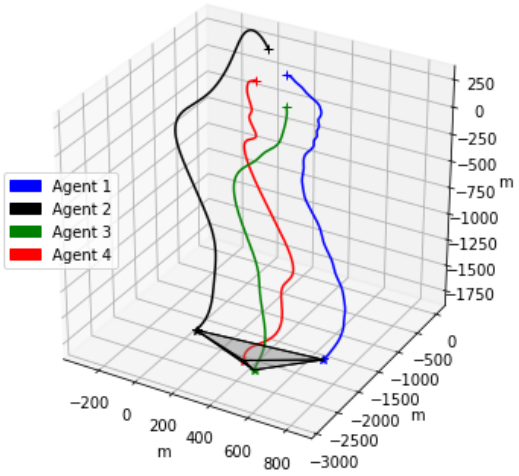


Fig. 2: Nominal control with unknown perturbations

The learning control law has been implemented using a square exponential kernel. At $t = 0$, the training set $\mathcal{D}_{m(0)}$ is empty and a new data point is added every 0.02s, until the final amount of 500 data points is reached at $t = 10$s. With the learning control law, it can be observed the agents converge to a close neighborhood around the desired formation as shown in Fig. 3. In order to better observe this behaviour, we also provide a video for the numerical experiment[2]. Finally, the evolution of the Lyapunov function in Fig. 4 highlights the superiority of the proposed control law as it allows the Lyapunov function to converge to a tight set around zero. Note that the evolution of the Lyapunov function is not always decreasing but bounded in a neighborhood around zero, see (20). The size of the set shrinks with improved accuracy of the GP model.

VI. Conclusions

We present an online learning-based control law for stable flocking of second-order agents with unknown dynamics. By employing Gaussian Processes, the error to achieve desired flocking motions is proven to be bounded in probability.

[2]https://youtu.be/pL-9MNnZ71k

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.

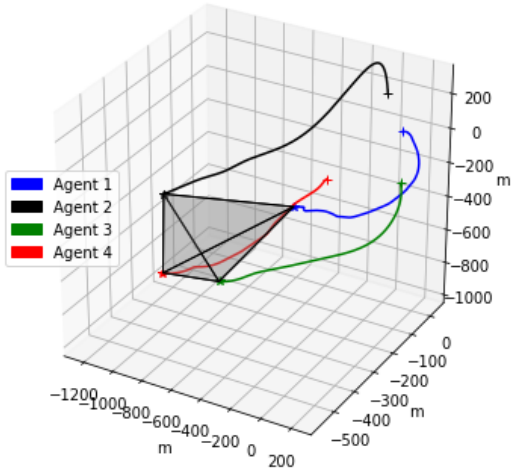


Fig. 3: The proposed learning-based control law can compensate the unknown perturbations.

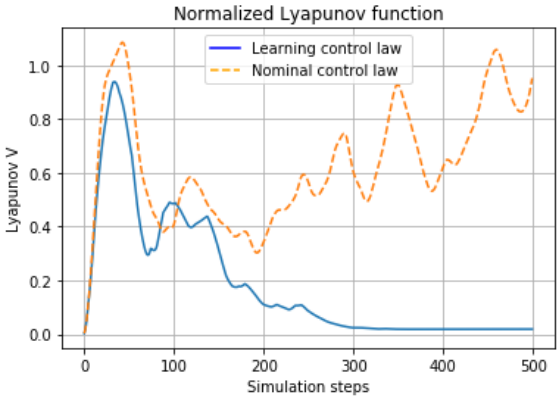


Fig. 4: Normalized Lyapunov function with the standard control law (dashed) and the proposed, learning-based control law (solid) which converges to a tight set around zero.

The online framework allows to improve the quality of the GPs over time and, thus, to improve stability performance. Finally, numerical examples visualize the effectiveness of the proposed control law.

REFERENCES

[1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[2] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, part i: Fixed topology," in *42nd IEEE Conference on Decision and Control (CDC)*, 2003, pp. 2010–2015.

[3] M. Deghat, B. Anderson, and Z. Lin, "Combined flocking and distance-based shape control of multi-agent formations." *IEEE Transactions on Automatic Control, 61*, pp. 1824–1837, 2016.

[4] Z. Sun and B. D. Anderson, "Rigid formation control systems modelled by double integrators: System dynamics and convergence analysis," in *5th Australian Control Conference (AUCC)*. IEEE, 2015, pp. 241–246.

[5] D. V. Dimarogonas and K. H. Johansson, "On the stability of distance-based formation control," in *47th IEEE Conference on Decision and Control (CDC)*, 2008, pp. 1200–1205.

[6] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[7] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[8] Y. Liu and Y. Jia, "An iterative learning approach to formation control of multi-agent systems," *Systems & Control Letters*, vol. 61, no. 1, pp. 148–154, 2012.

[9] C. Yuan, S. Licht, and H. He, "Formation learning control of multiple autonomous underwater vehicles with heterogeneous nonlinear uncertain dynamics," *IEEE Transactions on Cybernetics*, vol. 48, no. 10, pp. 2920–2934, 2017.

[10] T. Yan, X. Xu, Z. Li, and E. Li, "Flocking of multi-agent systems with unknown nonlinear dynamics and heterogeneous virtual leader," *International Journal of Control, Automation and Systems*, vol. 19, no. 9, pp. 2931–2939, 2021.

[11] G. Wen, C. L. P. Chen, and B. Li, "Optimized formation control using simplified reinforcement learning for a class of multiagent systems with unknown dynamics," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7879–7888, 2020.

[12] F. Schilling, J. Lecoeur, F. Schiano, and D. Floreano, "Learning vision-based flight in drone swarms by imitation," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4523–4530, 2019.

[13] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Conference on robot learning*. PMLR, 2020, pp. 671–682.

[14] Z. Yang, S. Sosnowski, Q. Liu, J. Jiao, A. Lederer, and S. Hirche, "Distributed learning consensus control for unknown nonlinear multi-agent systems based on gaussian processes," in *60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 4406–4411.

[15] T. Beckers, S. Hirche, and L. Colombo, "Online learning-based formation control of multi-agent systems with Gaussian processes," in *60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2197–2202.

[16] T. Z. Jiahao, L. Pan, and M. A. Hsieh, "Learning to swarm with knowledge-based neural ordinary differential equations," *arXiv preprint arXiv:2109.04927*, 2021.

[17] M. B. Bezcioglu, B. Lennox, and F. Arvin, "Self-organised swarm flocking with deep reinforcement learning," in *2021 7th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, 2021, pp. 226–230.

[18] Z.-S. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[19] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[20] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[21] J. Umlauft and S. Hirche, "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, 2020.

[22] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler–Lagrange systems," *Automatica*, vol. 103, pp. 390–397, 2019.

[23] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001, vol. 207.

[24] B. R. L. Asimow, "The rigidity of graphs, ii," *Journal of Mathematical Analysis and Applications*, vol. 68, no. 1, pp. 171–190, 1979.

[25] K. J. Åström and P. Eykhoff, "System identification—a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.

[26] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[27] T. Beckers, J. Umlauft, D. Kulic, and S. Hirche, "Stable Gaussian process based tracking control of Lagrangian systems," in *56th Conference on Decision and Control (CDC)*, 2017, pp. 5180–5185.

[28] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[29] K. Oh and H. Ahn, "Distance-based undirected formation of single-integrator and double-integrator modeled agents in n-dimensional space," *International Journal of Robust and Nonlinear Control*, pp. 1809–1820, 2014.

[30] F. Dörfler and F. Bullo, "On the critical coupling for kuramoto oscillators," *SIAM Journal on Applied Dynamical Systems*, vol. 10, no. 3, pp. 1070–1099, 2011.

[31] T. Beckers and S. Hirche, "Equilibrium distributions and stability analysis of Gaussian process state space models," in *55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 6355–6361.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:39:17 UTC from IEEE Xplore. Restrictions apply.