Declarative vs Rule-based Control for Flocking Dynamics

Usama Mehmood
Department of Computer Science, Stony Brook University, USA

Nicola Paoletti
Department of Computer Science, Stony Brook University, USA

Dung Phan
Department of Computer Science, Stony Brook University, USA

Radu Grosu
Cyber-Physical Systems Group, Technische Universitat Wien, Austria

Shan Lin
Department of Electrical and Computer Engineering, Stony Brook University, USA

Scott D. Stoller
Department of Computer Science, Stony Brook University, USA

Ashish Tiwari
SRI International, USA

Junxing Yang
Department of Computer Science, Stony Brook University, USA

Scott A. Smolka
Department of Computer Science, Stony Brook University, USA

ABSTRACT

The popularity of rule-based flocking models, such as Reynolds classic flocking model, raises the question of whether more declarative flocking models are possible. This question is motivated by the observation that declarative models are generally simpler and easier to design, understand, and analyze than operational models. We introduce a very simple control law for flocking based on a cost function capturing cohesion (agents want to stay together) and separation (agents do not want to get too close). We refer to it as *declarative flocking* (DF). We use model-predictive control (MPC) to define controllers for DF in centralized and distributed settings. A thorough performance comparison of our DF-based approach with Reynolds model, and with more recent flocking models that use MPC with a cost function based on lattice structures, demonstrate that DF-MPC yields the best cohesion and least fragmentation, and maintains a surprisingly good level of geometric regularity while still producing natural flock shapes similar to those produced by Reynolds model. We also show that DF-MPC has high resilience to sensor noise.

ACM Reference Format:
Usama Mehmood, Nicola Paoletti, Dung Phan, Radu Grosu, Shan Lin, Scott D. Stoller, Ashish Tiwari, Junxing Yang, and Scott A. Smolka. 2018. Declarative vs Rule-based Control for Flocking Dynamics. In *SAC 2018: SAC 2018: Symposium on Applied Computing , April 9–13, 2018, Pau, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3167132.3167222

1 INTRODUCTION

Flocking is a collective behavior exhibited by a large number of interacting agents possessing a common group objective [14]. The term is most commonly associated with birds, and more recently, drones. Examples include foraging for food, executing a predator-avoidance maneuver, and engaging in migratory behavior.

With the introduction of Reynolds model [21, 22], *rule-based control* became the norm in the flocking community. Specifically, in this model, at each time-step, each agent executes a control law given in terms of the weighted sum of three competing forces to determine its next acceleration. Each of these forces has its own rule: *separation* (keep a safe distance away from your neighbors), *cohesion* (move towards the centroid of your neighbors), and *alignment* (steer toward the average heading of your neighbors). As the descriptions suggest, these rules are executed by each agent in a distributed environment with limited-range sensing and no communication.

The popularity of Reynolds model and its many variants raises the question: Is there a more abstract *declarative* form of control for flocking? This question is important because declarative models are generally simpler and easier to design, understand, and analyze than operational models. This is analogous to declarative programs (e.g., functional programs and logic programs) being easier to write and verify than imperative programs.

We show that the answer to this question is indeed positive by providing a very simple control law for flocking based on a *cost function* comprising two main terms: *cohesion* (the average squared distance between all pairs of agents) and *separation* (the sum of inverse squared distances between all pairs of agents). That is it. For example, no term representing velocity alignment is required. The cost function specifies what we want as the goal, and is hence declarative. In contrast, Reynolds model does not formalize a goal. Its update rules are operational, and are designed to achieve goals described informally as cohesion, separation, and velocity alignment.

Executing declarative control amounts to finding the right balance between attracting and repelling forces between agents. We refer to this approach as *Declarative Flocking* (DF). We use model-predictive control (MPC) to define controllers for DF, and refer to this approach as *DF-MPC*. We define a centralized version of DF-MPC, which requires communication, and a distributed version, which does not.

Previous MPC-based aproaches to flocking exist [28, 29, 31] and are designed to conform to the α-lattice model of flocking proposed

SAC 2018, April 9–13, 2018, Pau, France

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167222

in [14]. α-lattices impose a highly regular structure on flocks, requiring all neighboring agents to be distance d apart, for a specified constant d. This kind of structure is seen in e.g. beehives, but is not witnessed in many other natural and engineered settings, and it is not imposed by Reynolds model.

In this paper, we show, via a thorough experimental evaluation, how the performance of controllers based on centralized and distributed DF-MPC compares to that of controllers based on Reynolds model [21, 22], Olfati-Saber's potential-based approach [14], (a variant of) Zhan and Li's centralized lattice-based MPC approach [27, 28], and Zhang *et al.*'s distributed lattice-based MPC approach [29]. We consider performance measures that capture multiple aspects of flocking behavior: number of sub-flocks (flock fragmentation), maximum sub-flock diameter (cohesion), velocity convergence, and a new parameter-free measure of the geometric regularity of the formation.

Our experimental results demonstrate that DF-MPC yields the best cohesion and least fragmentation, and produces natural flock shapes like those seen in Reynolds model. Also, distributed DF-MPC maintains a surprisingly good level of geometric regularity. We also analyze the resiliency of DF-MPC and lattice-based MPC to sensor noise. Our results demonstrate a remarkably high level of resiliency on the part of DF-MPC in comparison with these other approaches.

The rest of the paper is organized as follows. Section 2 presents the rule-based, potential-based, and lattice-based MPC approaches mentioned above. Section 3 defines our declarative flocking approach. Section 4 introduces our performance measures for flocking models. Section 5 presents our experimental results and performance evaluation. Section 6 discusses related work. Section 7 offers concluding remarks and directions for future work.

2 MODELS OF FLOCKING BEHAVIOR

We consider a set of dynamic *agents* $\mathcal{A} = \{1, \ldots, n\}$ that move according to the following discrete-time equation of motion:

$$x_i(k+1) = x_i(k) + dt \cdot v_i(k), \ |v_i(k)| \leq \bar{v} \tag{1}$$

$$v_i(k+1) = v_i(k) + dt \cdot a_i(k), \ |a_i(k)| \leq \bar{a}, \tag{2}$$

where $x_i(k), v_i(k), a_i(k) \in \mathbb{R}^m$ are respectively the position, velocity and acceleration of agent $i \in \mathcal{A}$, in m-dimensional space at time k, and $dt \in \mathbb{R}^+$ is the time step. Velocities and accelerations are bounded by $\bar{v}$ and $\bar{a}$, respectively.

In most flocking models, agents update their motion by changing their acceleration. In this sense, $a_i(k)$ represents the control input for agent i at time k. Accelerations are computed every η time steps, $\eta \in \mathbb{Z}^+$, and the time step at which accelerations are computed is called the *control step*. Accelerations remain constant during the interval from one control step to another.

The *configuration* of all agents is described by the vector $\mathbf{x}(k) = [x_1^T(k) \ \ldots \ x_n^T(k)]^T \in \mathbb{R}^{m \cdot n}$. Let $\mathbf{v}(k) = [v_1^T(k) \ \ldots \ v_n^T(k)]^T \in \mathbb{R}^{m \cdot n}$, and $\mathbf{a}(k) = [a_1^T(k) \ \ldots \ a_n^T(k)]^T \in \mathbb{R}^{m \cdot n}$. Then the equation of motion for all agents can be expressed as

$$\mathbf{x}(k+1) = \mathbf{x}(k) + dt \cdot \mathbf{v}(k), \tag{3}$$

$$\mathbf{v}(k+1) = \mathbf{v}(k) + dt \cdot \mathbf{a}(k), \tag{4}$$

The local neighborhood of agent i is defined by the set of other agents, called *neighbors*, within a given distance from i, mimicking the agent's visibility sphere. For an *interaction radius* $r > 0$ and configuration $\mathbf{x}$, the set of *spatial neighbors* of agent i, $N_i(\mathbf{x}) \subseteq \mathcal{A}$, is given by:

$$N_i(\mathbf{x}) = \left\{ j \in \mathcal{A} \mid j \neq i \wedge \|x_i - x_j\| < r \right\}, \tag{5}$$

where $\| \cdot \|$ denotes the Euclidean norm.

For configuration $\mathbf{x} \in \mathbb{R}^{m \cdot n}$, we define the associated *proximity net* $G(\mathbf{x}) = (\mathcal{A}, \mathcal{E}(\mathbf{x}))$ as the graph that connects agents within their interaction radius:

$$\mathcal{E}(\mathbf{x}) = \left\{ (i, j) \in \mathcal{A} \times \mathcal{A} \mid \|x_i - x_j\| < r, i \neq j \right\}, \tag{6}$$

To capture the regular geometry of flocks, Olfati-Saber introduced the notions of *α-lattices*, i.e. configurations where each agent is equally distant from its neighbors, and *quasi α-lattices*, i.e. configurations that are α-lattices modulo a small error in the distances [14]. The *scale* parameter d defines the ideal inter-agent distance.

Definition 2.1 (α-lattice [14]). A configuration $\mathbf{x} \in \mathbb{R}^{m \cdot n}$ is called *α-lattice* if for all $i \in \mathcal{A}$ and all $j \in N_i(\mathbf{x})$, $\|x_i - x_j\| = d$, where $d \in \mathbb{R}^+$ is the scale of the α-lattice. For tolerance $\delta \in \mathbb{R}^+$, a configuration $\mathbf{x} \in \mathbb{R}^{m \cdot n}$ is called a *quasi α-lattice* if for all $i \in \mathcal{A}$ and all $j \in N_i(\mathbf{x})$, $|\|x_i - x_j\| - d| \leq \delta$.

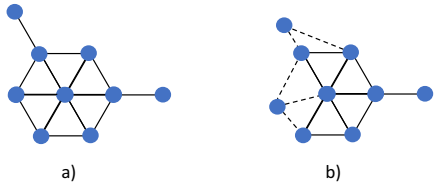


Figure 1: Examples of α-lattice a) and quasi α-lattice b). Solid lines connect agents in the same neighborhood that have distance d. Dashed lines connect those with have distance $d \pm \epsilon$ for $\epsilon \leq \delta$ (the tolerance).

2.1 Sensing noise

We extend the classical equations of motion, Eqs. (1)–(2), with *sensing noise* affecting how each agent perceives positions and velocities of its neighbors. Existing work has put little focus on flocking dynamics subject to noise, which is unfortunately unavoidable in realistic natural and engineered flocks.

For actual positions $\mathbf{x}(k)$ and velocities $\mathbf{v}(k)$ at step k, let $\tilde{\mathbf{x}}(k)$ and $\tilde{\mathbf{v}}(k)$ denote the possibly noisy versions of these quantities sensed by some agent, defined by:

$$\tilde{\mathbf{x}}(k) = \mathbf{x}(k) + \mathbf{nx}(k) \text{ and } \tilde{\mathbf{v}}(k) = \mathbf{v}(k) + \mathbf{nv}(k), \tag{7}$$

where $\mathbf{nx}(k)$ and $\mathbf{nv}(k)$ in $\mathbb{R}^{m \cdot n}$ are vectors of independent and identically distributed (i.i.d.) random variables. The position noise $\mathbf{nx}(k)$ and velocity noise $\mathbf{nv}(k)$ are distributed according to Gaussian distributions with mean 0 and standard deviation σ_x and σ_v, respectively. We stress the dependency on k because noise variables are independent across control intervals. (In our performance evaluation, we ran experiments for 10 noise levels, with $-0.2 \leq \sigma_x \leq 0.2$ and $0.1 \leq \sigma_v \leq 1.0$.)

In centralized flocking algorithms, where agent decisions are computed by a single controller with information about the whole

population, we use Eq. 7 to define noisy measurements. In the distributed setting, sensing noise is independent for each agent. We denote the noisy measurement of agent i by $\tilde{\mathbf{x}}^{\triangleright i}(k)$ and $\tilde{\mathbf{v}}^{\triangleright i}(k)$, where positions and velocities are noisy for all agents except agent i

$$\tilde{\mathbf{x}}^{\triangleright i}(k) = [\tilde{x}_1^T(k) \dots x_i^T(k) \dots \tilde{x}_n^T(k)]^T \text{ and} \quad (8)$$

$$\tilde{\mathbf{v}}^{\triangleright i}(k) = [\tilde{v}_1^T(k) \dots v_i^T(k) \dots \tilde{v}_n^T(k)]^T, \quad (9)$$

with $\tilde{x}_1(k), \dots, \tilde{x}_n(k)$ and $\tilde{v}_1(k), \dots, \tilde{v}_n(k)$ defined as per (7); implicitly, for each agent i and each other agent j, the noise distribution is sampled independently to compute the $\tilde{x}_j^T(k)$ component of $\tilde{\mathbf{x}}^{\triangleright i}(k)$.

2.2 Reynolds rule-based model

In Reynolds rule-based distributed model [21, 22], agents follow simple rules to compute their accelerations from the positions and velocities of their neighbors. The rules are illustrated in Figure 2. They do not explicitly specify the desired flocking formation as an objective; rather, flocking emerges from the interaction rules.

Specifically, each agent $i \in \mathcal{A}$ updates its acceleration $a_i(k)$ at control step k by considering the following three components (adapted to include sensing noise):

(1) *Alignment*: agents match their velocities with the average velocity of nearby agents.

$$a_i^{al}(k) = w_{al} \cdot \left(\left(\frac{1}{|N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))|} \cdot \sum_{j \in N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))} \tilde{v}_j(k) \right) - v_i(k) \right) \quad (10)$$

(2) *Cohesion*: agents move towards the centroid of the agents in the local neighborhood.

$$a_i^{c}(k) = w_{c} \cdot \left(\left(\frac{1}{|N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))|} \cdot \sum_{j \in N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))} \tilde{x}_j(k) \right) - x_i(k) \right) \quad (11)$$

(3) *Separation*: agents move away from nearby neighbors.

$$a_i^{s}(k) = w_{s} \cdot \frac{1}{|N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))|} \cdot \left(\sum_{j \in N_i(\tilde{\mathbf{x}}^{\triangleright i}(k))} \frac{x_i(k) - \tilde{x}_j(k)}{\|x_i(k) - \tilde{x}_j(k)\|^2} \right) \quad (12)$$

The cohesion and alignment rules help form and maintain a closely packed, flock-like formation. The separation rule prevents agents from coming too close to each other, thus reducing crowding and collisions.

Non-negative constants w_{al}, w_c and w_s are the weights for each acceleration component. Typically, a smaller interaction radius (hence a smaller neighborhood) is used for the separation rule, because it is significant only when agents are very close to each other. The overall acceleration in Reynolds model is given by:

$$a_i(k) = a_i^{al}(k) + a_i^{c}(k) + a_i^{s}(k). \quad (13)$$

2.3 Olfati-Saber's potential-based model

In potential-based flocking models, the interaction between a pair of agents is modeled by a potential field. It is assumed that an agent is a point source, and it has a potential field around it, which exerts a force, equal to its gradient, on other agents in its range of influence. The potential field has circular symmetry and hence is a function of distance from the source. In the work of Olfati-Saber [14], the potential function ψ_α for a pair of agents has its minimum at the desired inter-agent distance d of the desired α-lattice. Outside the interaction radius r, the potential function is constant, so the potential field exerts no force. The exact definition of ψ_α is complicated: it is the definite integral of an "action function" ϕ_α that is the product of a "bump function" ρ_h and an uneven sigmoidal function ϕ. The control law computes an agent's acceleration based on the sum of the forces from all other agents in its neighborhood and a velocity alignment term.

2.4 MPC-based models

Model predictive control (MPC) is a well-established control technique that works as follows [4]: at each control step k, it computes the optimal control sequence (agents accelerations in our case) that minimizes a given cost function with respect to a predictive model of the controlled system and a finite prediction horizon of T control steps. Then, the first control input of the optimal sequence is applied (the remainder of the sequence is unused), and the algorithm proceeds with a new iteration.

Two main kinds of MPC-based flocking models exist, *centralized* and *distributed*. Centralized models assume that information about positions and velocities of all agents is available to compute their optimal accelerations. Formally, at each control step k, it solves the following optimization problem:

$$\min_{\substack{\mathbf{a}(k|k), \mathbf{a}(k+\eta|k), \dots, \\ \mathbf{a}(k+(\eta-1)\cdots T|k) \in A}} J(k) + \lambda \cdot \sum_{k'=0}^{T-1} \|\mathbf{a}(k + \eta \cdot k' \mid k)\|^2 \quad (14)$$

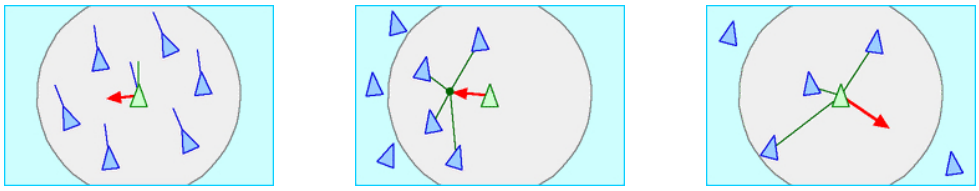
where $\mathbf{a}(k + \eta \cdot k' \mid k)$ is the control input (accelerations) for all agents at predicted control step $k + \eta \cdot k'$ starting from control step k. The first term $J(k)$ is the primary model-specific cost function that the controller seeks to optimize within the prediction horizon; it is implicitly a function of the predicted configurations during the prediction horizon for time step k. The second term is standard for MPC problems and penalizes large control inputs, with weight $\lambda > 0$.

In distributed flocking models, each agent computes its optimal acceleration based only on information about its neighbors. Each agent i solves an optimization problem of the form:

$$\min_{\substack{a_i(k|k), a_i(k+\eta|k), \dots, \\ a_i(k+(\eta-1)\cdot T|k) \in A}} J_i(k) + \lambda \cdot \sum_{k'=0}^{T-1} \left\| a_i(k + \eta \cdot k' \mid k) \right\|^2 \quad (15)$$

where $a_i(k + \eta \cdot k' \mid k)$ is the acceleration for agent i at predicted control step $k + \eta \cdot k'$ starting from step k, and $J_i(k)$ is the model-specific cost function for agent i. In distributed MPC, an agent has no way to know current or future control decisions of its neighbors, which are needed to make accurate predictions about their behavior. To address this problem, some approaches allow agents to communicate their local control decisions or future positions (e.g. [28, 31]), or assume that neighbors follow some default motion law, e.g., they move with constant velocities. We adopt the second strategy, because it does not require any communication.

The majority of existing MPC-based approaches to flocking are designed to optimize the regularity of the flock, by penalizing configurations where neighboring agents are not exactly distance d apart, i.e., configurations that differ from an α-lattice [27–29, 31].



(a) Alignment (b) Cohesion (c) Separation

Figure 2: Interaction rules for flocking behavior in Reynolds Model. Image taken from [19].

We call these approaches *lattice-based MPC*. Next we describe representative centralized and distributed lattice-based MPC flocking models, which we extend to account for sensing noise. The centralized model is a variant of a model by Zhan and Li [27, 28]. The distributed model is by Zhang *et* al. [29].

2.4.1 Centralized lattice-based MPC flocking. The centralized lattice-based MPC problem is defined as:

$$\min_{\substack{\mathbf{a}(k\mid k),\mathbf{a}(k+\eta\mid k),\ldots,\\ \mathbf{a}(k+(\eta-1)\cdots T\mid k)\in A}} \sum_{k'=1}^{T} \|g\left(\mathbf{x}(k+\eta\cdot k'\mid k)\right)\|^2+\lambda\cdot\sum_{k'=0}^{T-1}\|\mathbf{a}(k+\eta\cdot k'\mid k)\|^2 \tag{16}$$

where $\mathbf{x}(k+\eta\cdot k'\mid k)$ is the configuration of the system at predicted control step $k+\eta\cdot k'$ starting from control step k, following the dynamics:

$$\begin{aligned} x_i(k\mid k) &= \tilde{x}_i(k) \quad v_i(k\mid k)=\tilde{v}_i(k)\\ x_i(k+\eta\cdot k'+1\mid k) &= x_i(k+\eta\cdot k'\mid k)+dt\cdot v_i(k+\eta\cdot k'\mid k)\\ v_i(k+\eta\cdot k'+1\mid k) &= v_i(k+\eta\cdot k'\mid k)+dt\cdot a_i(k+\eta\cdot k'\mid k), \end{aligned} \tag{17}$$

where the initial state of the prediction window is given by noisy measurements.

For configuration $\mathbf{x}$, $g(\mathbf{x})$ captures the α-lattice irregularity as the total deviation between agent distances and d:

$$\|g(\mathbf{x})\|^2 = \sum_{(i,j)\in\mathcal{E}(\mathbf{x})}\left\|x_{ji}-\frac{d\cdot x_{ji}}{\|x_{ji}\|}\right\|^2, \text{ with } x_{ji}=x_j-x_i. \tag{18}$$

This model is inspired by [27] and [28] but differs from both: in [27] the cost function also contains a velocity alignment term, which the same authors removed in their subsequent work, while in [28], "impulsive MPC" is used, which means that agents directly control their velocities (instead of accelerations), an abstraction that allows physically unrealizable accelerations.

2.4.2 Distributed lattice-based MPC flocking. In the distributed MPC flocking model of Zhang *et* al. [29], each agent i controls its acceleration based on position and velocity measurements of the neighbors and assumes they have constant velocity (zero acceleration) during the prediction horizon. Similarly, the set of neighbors of i is assumed invariant during the prediction horizon, and we denote it by $N_i(k)=N_i(\mathbf{x}(k))$. The control law for agent i is:

$$\min_{\substack{a_i(k\mid k),a_i(k+\eta\mid k),\ldots,\\ a_i(k+(\eta-1)\cdot T\mid k)\in A}} \sum_{k'=1}^{T}\|g_i\left(\mathbf{x}(k+\eta\cdot k'\mid k)\right)\|^2+ \lambda\cdot\sum_{k'=0}^{T-1}\|a_i(k+\eta\cdot k'\mid k)\|^2. \tag{19}$$

Similar to the centralized flocking, future dynamics of i is determined by equation (17).

while i's neighbors $j\in N_i(k)$ have constant velocity:

$$x_j(k\mid k)=\tilde{x}_j(k) \quad x_j(k+\eta\cdot k'+1\mid k)=x_j(k+\eta\cdot k'\mid k)+dt\cdot\tilde{v}_j(k).$$

For configuration $\mathbf{x}$, $g_i(\mathbf{x})$ is defined in a similar way to Eq. (18) and quantifies how much i's neighborhood $N_i(k)$ deviates from an α-lattice:

$$\|g_i(\mathbf{x})\|^2 = \sum_{j\in N_i(k)}\left\|x_{ji}-\frac{d\cdot x_{ji}}{\|x_{ji}\|}\right\|^2. \tag{20}$$

3 DECLARATIVE FLOCKING

This section introduces centralized and distributed versions of our Declarative Flocking (DF) model, and presents a flocking algorithm based on MPC. Our formulation is declarative in that it consists of just two simple terms: (1) a cohesion term based on the average squared distance between pairs of agents, to keep the flock together, and (2) a separation term based on the inverse squared distances between pairs of agents, to avoid crowding. These two terms represent opposing forces on agents, causing agents to move towards positions in which these forces are balanced. Unlike the majority of existing MPC-based approaches that are designed to optimize conformance to an α-lattice, our design does not impose a specific geometric structure.

3.1 Centralized DF model

The cost function J for our centralized DF model contains the two terms described above. Both the cohesion term and the separation term consider all pairs of agents. , The weight ω of the separation term provides control over the density of the flock.

$$J^{\mathrm{C}}(\mathbf{x}) = \frac{2}{|\mathcal{A}|\cdot(|\mathcal{A}|-1)}\cdot\sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{A},i<j}\|x_{ij}\|^2+\omega\cdot\frac{1}{\|x_{ij}\|^2} \tag{21}$$

The control law is Eq. (14) with $J(k)$ equal to $\sum_{k'=1}^{T}J^{\mathrm{C}}(\mathbf{x}(k+\eta\cdot k'\mid k))$.

3.2 Distributed DF model

The cost function J for our distributed DF model is similar to the centralized one, except that both terms are limited to consider pairs of agents that are neighbors.

$$J_i^{\mathrm{D}}(\mathbf{x}) = \frac{1}{|N_i(k)|} \cdot \sum_{j \in N_i(k)} \|x_{ij}\|^2 + \omega \cdot \sum_{j \in N_i(k)} \frac{1}{\|x_{ij}\|^2} \tag{22}$$

The control law for agent i is Eq. (15) with $J_i(k)$ equal to $\sum_{k'=1}^{T} J_i^{\mathrm{D}}(\mathbf{x}(k + \eta \cdot k' \mid k))$.

4 MEASURES OF FLOCKING PERFORMANCE

We introduce four key measures of flocking performance. A single measure is insufficient, because flocking is indeed characterized by multiple desirable properties, such as aligned velocities and cohesion. Olfati-Saber introduces four main properties for flocking [14], informally described as:

(1) the group of agents stays *connected* in a unique flock, i.e., no sub-flocks and fragmentation should emerge;
(2) the group remains *cohesive*, in a close-knit formation;
(3) the group moves in a coherent way as if it was a unique body, i.e., agents' velocities are aligned; and
(4) the group maintains a regular geometry (in the α-lattice sense).

We introduce the following four measures to capture these four requirements. An important concept in these definitions is a *sub-flock*, which is a set of interacting agents that is too far apart from other agents to interact with them. Formally, a sub-flock in a configuration $\mathbf{x}$ corresponds to a connected component of the proximity net $G(\mathbf{x})$. Let $CC(\mathbf{x}) \subseteq 2^{\mathcal{A}}$ be the set of connected components of the proximity net $G(\mathbf{x})$.

(1) The *number of connected components* of the proximity net quantifies connectedness—or, equivalently, fragmentation—of the flock. There is no fragmentation when $|CC(\mathbf{x})| = 1$. Fragmentation exists when $|CC(\mathbf{x})| > 1$. Fragmentation may be temporary or, if sub-flocks move in different directions, permanent.

(2) The *maximum component diameter*, denoted $D(\mathbf{x})$, quantifies cohesion. It is defined by

$$D(\mathbf{x}) = \max_{\mathcal{A}' \in CC(\mathbf{x})} D(\mathbf{x}, \mathcal{A}') \tag{23}$$

where $D(\mathbf{x}, \mathcal{A}')$ is the diameter of connected component $\mathcal{A}'$:

$$D(\mathbf{x}, \mathcal{A}') = \max_{\substack{(i,j) \in \mathcal{A}' \times \mathcal{A}' \\ i \neq j}} \|\mathbf{x}_{ij}\|. \tag{24}$$

Note that when all agents are isolated, i.e., $CC(\mathbf{x}) = \bigcup_{i \in \mathcal{A}} \{\{i\}\}$, $D(\mathbf{x}) = -\infty$ because the domain of the max function in Equation 24 is empty when $\mathcal{A}'$ is a singleton. Note that we consider the maximum diameter of a sub-flock in order to make this measure more independent of connectedness. If we instead considered the overall diameter of the entire (possibly fragmented) flock, any flocking model that did poorly on connectedness would also do very poorly on this measure.

(3) The *velocity convergence* measure, adopted from [29], quantifies the average discrepancy between each agent's velocity and the average velocity of the flock. In particular, we extend the measure of [29] to average velocity convergence values across sub-flocks:

$$VC(\mathbf{x}, \mathbf{v}) = \frac{\sum_{\mathcal{A}' \in CC(\mathbf{x})} \left\| \sum_{i \in \mathcal{A}'} v_i - \left(\frac{\sum_{j \in \mathcal{A}'} v_j}{|\mathcal{A}'|} \right) \right\|^2 \Big/ |\mathcal{A}'|}{|CC(\mathbf{x})|} \tag{25}$$

(4) To measure the regularity of the geometric structure of a flock, as reflected in the inter-agent spacing, we introduce a parameter-free and model-independent *irregularity* measure $I(\mathbf{x})$. For a connected component (sub-flock) $\mathcal{A}'$, it is defined as the sample standard deviation of the distances between each agent in $\mathcal{A}'$ and its closest neighbor. Thus, the measure penalizes configurations where there is dispersion in inter-agent distances, while not imposing any fixed distance between them (unlike α-lattices).

Let $CC'(\mathbf{x}) = CC(\mathbf{x}) \setminus \bigcup_{i \in \mathcal{A}} \{\{i\}\}$ be the set of connected components where isolated agents are excluded. For $|CC'(\mathbf{x})| = 0$ (or equivalently, $|CC(\mathbf{x})| = |\mathcal{A}|$), i.e., all agents are isolated, we set the irregularity $I(\mathbf{x}) = 0$, which is the optimal value. This reflects the fact that a single point is a regular structure on its own. Moreover, such a configuration is already highly penalized by $|CC(\mathbf{x})|$ and $VC(\mathbf{v})$. For $|CC'(\mathbf{x})| > 0$, the measure is defined by:

$$I(\mathbf{x}) = \frac{\sum_{\mathcal{A}' \in CC'} \sigma\left(\biguplus_{i \in \mathcal{A}'} \min_{j \neq i} \|x_{ij}\| \right)}{|CC'|}. \tag{26}$$

where $\sigma(S)$ is the standard deviation of the multiset of samples S and $\biguplus$ is the sum operator (or disjoint union) for multisets.

An α-lattice (see Def. 2.1) has the optimal value of $I(\mathbf{x})$, i.e., $I(\mathbf{x}) = 0$, since all neighboring agents are located at the same distance d from each other, leading to zero standard deviation for the term $\sigma(\{d, d, \ldots, d\})$. This shows that $I(\mathbf{x})$ captures the regularity underlying the concept of α-lattice.

We introduce this measure because previous measures of regularity or irregularity, such as those in [14, 28, 29], measure deviations from an α-lattice with a specified inter-agent distance d and are therefore inapplicable to flocking models, such as Reynolds model and our DF models, that are not based on α-lattices and do not have a specified target inter-agent distance. Also, our irregularity measure is more flexible than those based on α-lattices, because it gives an optimal score to some configurations that are geometrically regular but not α-lattices. For example, consider a configuration $\mathbf{x}$ in which the agents are on the vertices of a grid with edge length e, and the interaction radius is equal to the length of the diagonal of a box in the grid. This configuration has an optimal value for our irregularity measure, i.e., $I(\mathbf{x}) = 0$, because the distance from every agent to its nearest neighbor is e. This configuration is not an α-lattice and hence does not nave an optimal value for the irregularity measures used in prior work.

5 PERFORMANCE EVALUATION

We compare the performance of the models of Section 2 with the newly introduced DF flocking models in the 2-dimensional setting. In the first set of experiments (Section 5.1), we evaluate the performance measures illustrated in Section 4. In the second set of experiments (Section 5.2), we analyze the resilience of the algorithms to sensor noise. It is important to note here that we do not

evaluate performance for specific motion objectives such as total traveled distance. Only flocking behavior is considered here.

By Olfati-Saber model we refer to Algorithm 1 that is defined in equation (23) in [15]. This model does not have a group objective, in accordance with the flocking models considered in this paper.

For consistency with the experimental settings of [29], the lattice-based MPC problems are solved using the interior point method implemented in MATLAB's `fmincon` function. Our DF-MPC problems are solved using gradient descent optimization. Unless otherwise specified, the population size is $n = 30$, the simulation length is 100, $dt = 0.3$, $\bar{v} = 8$, $\bar{a} = 1$, $r = 8.4$, $d = 7$, $T = 3$, and $\lambda = 1$. These parameter values are the same ones reported in [29]. The control interval is 0.9, i.e., $\eta = 3$, for our centralized and distributed DF-MPC. For other models, $\eta = 1$ as they do not distinguish control interval from time step. Following the settings in the OpenSteer project [20], the parameters for Reynolds model are $r_c = 9$, $r_s = 5$, $r_{al} = 7.5$, $w_c = 8$, $w_s = 12$, and $w_{al} = 8$. The weight ω of the separation term in our centralized and distributed DF-MPC is 2000 and 30, respectively. As in [29], initial positions and initial velocities of agents are uniformly sampled from $[-15, 15]^2$ and $[0, 2]^2$, respectively.

5.1 Performance Comparison of Flocking Algorithms

Fig. 3 shows examples of final formations for all flocking models. In particular, we chose configurations where fragmentation did not occur. We observe that the formations for lattice-based MPC algorithms have spread-out, rigid structures, consistent with the design objective of maximizing the α-lattice regularity. On the other hand, Reynolds and our DF MPC models result in more natural flock shapes.

In Fig. 4, we compare the performance measures for the different flocking models. The graphs are obtained by averaging the performance measures at each time-step for 100 runs. Regarding the number of connected components (sub-flocks), our centralized DF-MPC registers the best behavior, rapidly stabilizing to an average of 1 component (see plot a). The next best are Reynolds Model and our distributed DF-MPC with 1.2 sub-flocks and 1.6 sub-flocks on average, respectively The lattice-based MPCs and Olfati-Saber instead lead to constant fragmentation, with more than 2 sub-flocks for the distributed lattice-based MPC, 6 for the centralized lattice-based MPC, and more than 8 for Olfati-Saber's model.

This ranking is confirmed by the diameter measure (plot b), where our centralized and distributed DF-MPC and Reynolds model show the best cohesion, outperforming the lattice-based approaches. Recall that this measure indicates the maximum diameter over all sub-flocks, not the diameter of the entire population. As a consequence, fragmentation tends to decrease diameter values since it produces sub-flocks with fewer individuals. This explains why Olfati-Saber's model has smaller diameter measure than centralized lattice-based MPC, which in turn has smaller diameter measure than the distributed variant.

In our centralized DF-MPC, oscillations can be seen in the max component diameter (plot b) and irregularity (plot c) from time steps 5 to 18. This is because our centralized controller initially applies aggressive velocity adjustments (accelerations) and consequently overcompensates. The overshoot can be decreased both by increasing the length of the MPC prediction horizon and by decreasing the size of the control step.

As expected, Olfati-Saber and the lattice-based MPCs perform very well in terms of irregularity (plot c), since they are designed to achieve the geometric regularity of an α-lattice. Somewhat surprisingly, our distributed DF-MPC and centralized DF-MPC perform almost as well as these approaches on this measure. Reynolds model exhibits the least regularity in its formations.

For velocity convergence (plot d), we find that all models perform comparably well and are able to achieve flocks with consistent velocities fairly quickly after a sharp initial spike. The spike owes its presence to the fact that to quickly maneuver into a flock-like formation from a semi-random initial, velocity convergence has to be compromised.

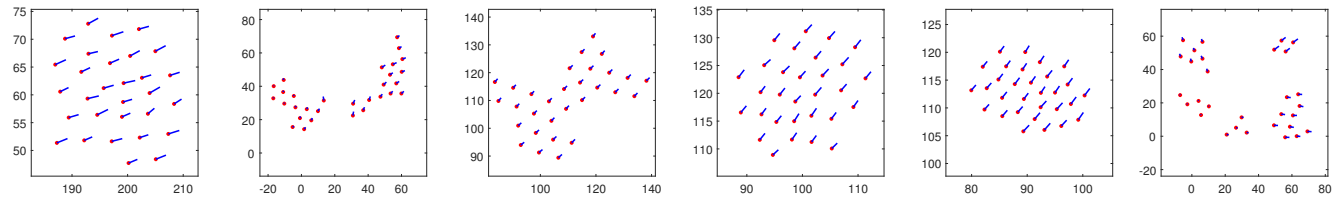
5.2 Robustness to Sensing Noise

To evaluate the resiliency of the models to sensor noise, we performed 20 runs for each model at 10 noise levels. The noise levels are numbered from 1 to 10, and noise level i has $\sigma_x = 0.2i$ and $\sigma_v = 0.1i$. For each performance metric, we averaged its final values over 20 runs for each noise level. The results are plotted in Fig. 5. Of the six models, Olfati-Saber's model is the most vulnerable to sensing noise: the number of sub-flocks $|CC|$ in Olfati-Saber's model quickly increases to nearly 30, rendering other metrics irrelevant. The lattice-based MPC models also exhibit high fragmentation, leading to nominally good but largely irrelevant values for the other performance metrics. Our distributed DF-MPC has the best resiliency to sensing noise, as it exhibits stable profiles in all metrics. Furthermore, distributed DF-MPC, centralized DF-MPC and, Reynolds model remarkably maintain almost a single connected component with a nearly constant component diameter for all noise levels. On the irregularity measure, our distributed DF-MPC performs significantly better then the centralized DF-MPC.

6 RELATED WORK

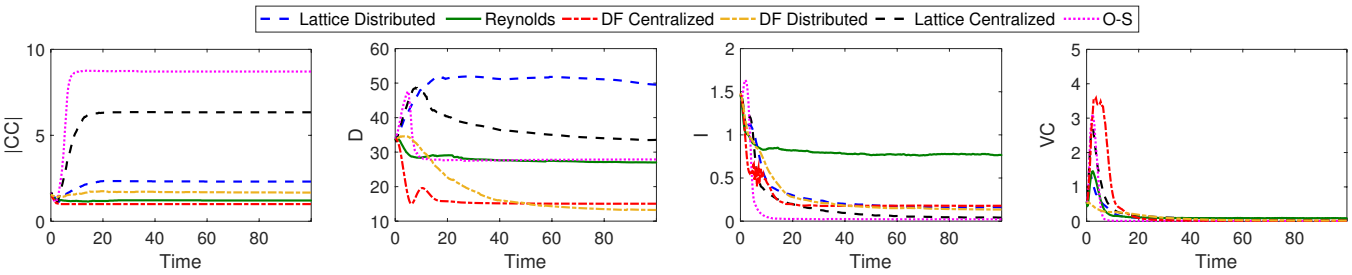
We discuss related work on rule-based and MPC-based flocking algorithms, as well as research on flocking under noise and stochastic dynamics. For an overview of flocking and more general consensus and formation-control problems, we refer the reader to the reviews in [13, 15].

Since the introduction of Reynolds rule-based flocking model [21], a number of other models have been proposed. Vicsek *et al.* [25] studied flocking from a physical perspective, by introducing a minimal discrete-time model of motion where agents move at a constant velocity and align their directions with the average directions of their respective neighbors (subject to noisy updates). This model can explain the phase transition from disordered to ordered/collective motion. Cucker and Smale [7] extended Vicsek's model in such a way that the contributions of neighbors to the velocity of an agent are inversely proportional to their distances to that agent, and proved velocity convergence under different parameterizations. This work has been further extended in [6] to provide collision-avoidance guarantees. Another strategy is considered in the work of Pearce *et al.* [16], where agents move to maximize their view out



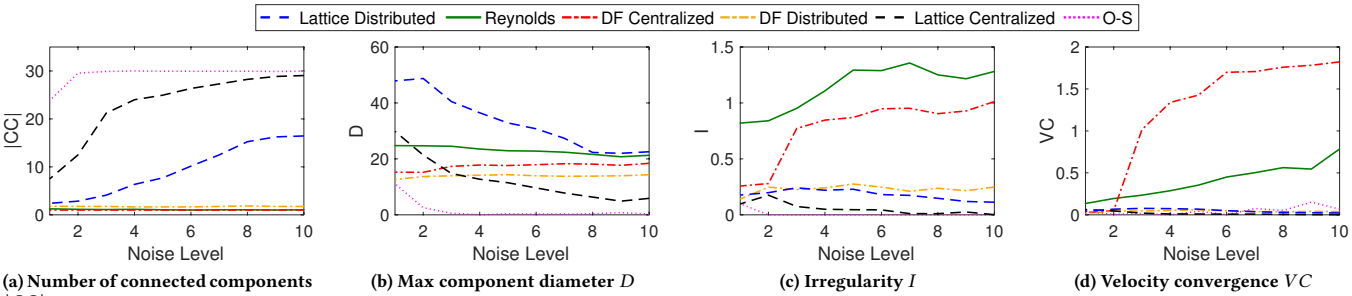
(a) Reynolds (b) Lattice-based centralized MPC (c) Lattice-based distributed MPC (d) DF centralized MPC (e) DF distributed MPC (f) Olfati-Saber

Figure 3: Examples of final formations for different flocking models. The red dots are the agent positions. The blue lines denote the agent velocities; the line lengths are proportional to the speeds.



(a) Number of connected components $|CC|$ (b) Max component diameter D (c) Irregularity I (d) Velocity convergence VC

Figure 4: Comparison of performance measures obtained with 100 runs for each flocking algorithm.



(a) Number of connected components $|CC|$ (b) Max component diameter D (c) Irregularity I (d) Velocity convergence VC

Figure 5: Comparison of the final values of the performance measures obtained with 20 runs for each flocking algorithm and for each noise level. The noise levels are numbered from 1 to 10, and noise level i has $\sigma_x = 0.2i$ and $\sigma_v = 0.1i$.

of the flock. Artificial potential fields have long been deployed in motion planning [10] and, more recently, applied to flocking models to represent inter-agent influences in terms of attractive/repulsive forces [14, 17, 18, 23].

In contrast to the above methods, our declarative flocking approach does not consider a fixed set of rules or dynamical equations, but agent motion is the result of an optimal control problem, solved with MPC techniques. Our cost functions incorporate notions of cohesion and separation that are at the core of many rule-based models.

Most of the existing MPC-based approaches to flocking are designed to conform to the α-lattice structure introduced by Olfati-Saber [14], resulting in formations with strict geometric regularity. Examples of such lattice-based MPC approaches include [27–29, 31]. Our declarative flocking method is also based on MPC but is designed to optimize a combination of cohesion and separation, leading to more natural flock geometry resembling those produced by Reynolds model, yet maintaining a good level of geometric regularity (unlike Reynolds model). Note that our approach achieves velocity alignment without directly optimizing for it, in contrast to other MPC flocking approaches such as [2, 5], where velocity alignment is explicitly included in the cost function. MPC was also used in [12, 24] to study another kind of collective behavior, V-formation.

Prior work exists that extend well-established models like Cucker-Smale and Olfati-Saber to study flocking under noise/stochasticity in sensing, actuation and in the agents' environment; see e.g. [1, 3,

8, 9, 11, 26]. In this paper we similarly compare the robustness of different flocking models under increasing levels of sensing noise, evaluating the effect of noise on fragmentation, cohesion, velocity convergence, and regularity.

7 CONCLUSIONS

We have presented an abstract, declarative form of control for flocking behavior along with the results of a thorough performance comparison of centralized and distributed versions of our MPC-based declarative flocking with four other flocking models. Our results demonstrate that DF-MPC yields the best cohesion and least fragmentation, and produces natural flock shapes like those produced by Reynolds rule-based model. Our resiliency analysis shows that the distributed version of DF-MPC is highly robust to sensor noise.

As future work, we plan to study resilience of the flocking models with respect to additional stochastic scenarios, such as actuation noise (i.e., noise affecting acceleration) and faulty agents with deviant behavior. We also plan to investigate smoothing techniques to increase resilience to sensor noise.

Additionally, to reduce the online computational overhead associated with MPC-based techniques such as our declarative flocking, we plan to learn the associated control laws in the form of a Deep Neural Network (DNN); this is motivated in part by Zhang *et al.*'s success using MPC to train a DNN-based controller for collision avoidance [30]. After training, the online computational cost is reduced to the cost of evaluating the DNN on an input.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This material is based on work supported in part by AFOSR Grant FA9550-14-1-0261, NSF Grants CPS-1446832, IIS-1447549, CNS-1445770, CNS-1421893, CNS-1553273, CNS-1463722, CNS-1423298, and CCF-1414078, FWF-NFN RiSE Award, and ONR Grant N00014-15-1-2208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these agencies.

REFERENCES

[1] Shin Mi Ahn and Seung-Yeal Ha. 2010. Stochastic flocking dynamics of the Cucker-Smale model with multiplicative white noises. *J. Math. Phys.* 51, 10 (2010), 103301. https://doi.org/10.1063/1.3496895

[2] G. Albi and L. Pareschi. 2016. Selective model-predictive control for flocking systems. *ArXiv e-prints* (2016). arXiv:math.OC/1603.05012

[3] Giacomo Albi, Lorenzo Pareschi, and Mattia Zanella. 2015. Uncertainty quantification in control problems for flocking models. *Mathematical problems in Engineering* 2015 (2015).

[4] Eduardo F. Camacho and Bordons Alba Carlos. 2007. *Model predictive control.* Springer.

[5] Zhaomeng Cheng, Hai-Tao Zhang, Ming-Can Fan, and Guanrong Chen. 2015. Distributed consensus of multi-agent systems with input constraints: A model predictive control approach. *IEEE Transactions on Circuits and Systems I: Regular Papers* 62, 3 (2015), 825–834.

[6] Felipe Cucker and Jiu-Gang Dong. 2011. A general collision-avoiding flocking framework. *IEEE Trans. Automat. Control* 56, 5 (2011), 1124–1129.

[7] F. Cucker and S. Smale. 2007. Emergent Behavior in Flocks. *IEEE Trans. Automat. Control* 52, 5 (May 2007), 852–862. https://doi.org/10.1109/TAC.2007.895842

[8] Radek Erban, Jan Haskovec, and Yongzheng Sun. 2016. A Cucker–Smale Model with Noise and Delay. *SIAM J. Appl. Math.* 76, 4 (July 2016), 1535–1557. https://doi.org/10.1137/15M1030467

[9] Seung-Yeal Ha, Kiseop Lee, Doron Levy, et al. 2009. Emergence of time-asymptotic flocking in a stochastic Cucker-Smale system. *Communications in Mathematical Sciences* 7, 2 (2009), 453–469.

[10] Oussama Khatib. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research* 5, 1 (1986), 90–98.

[11] H. M. La and W. Sheng. 2010. Flocking control of multiple agents in noisy environments. In *2010 IEEE International Conference on Robotics and Automation*. 4964–4969. https://doi.org/10.1109/ROBOT.2010.5509668

[12] Anna Lukina, Lukas Esterle, Christian Hirsch, Ezio Bartocci, Junxing Yang, Ashish Tiwari, Scott A. Smolka, and Radu Grosu. 2017. ARES: Adaptive Receding-Horizon Synthesis of Optimal Plans. In *Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017*. 286–302. https://doi.org/10.1007/978-3-662-54580-5_17

[13] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. 2015. A survey of multi-agent formation control. *Automatica* 53 (2015), 424–440.

[14] Reza Olfati-Saber. 2006. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on automatic control* 51, 3 (2006), 401–420.

[15] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. 2007. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* 95, 1 (2007), 215–233.

[16] Daniel J. G. Pearce, Adam M. Miller, George Rowlands, and Matthew S. Turner. 2014. Role of projection in the control of bird flocks. *Proceedings of the National Academy of Sciences* 111, 29 (2014), 10422–10426.

[17] Naomi Ehrich Leonard Peter Ogren. 2004. Cooperative control of mobile sensor networks:Adaptive gradient climbing in a distributed environment. *IEEE transactions on Automatic Control* 49, 8 (2004).

[18] John H. Reif and Hongyan Wang. 1999. Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems* 27, 3 (1999), 171 – 194. https://doi.org/10.1016/S0921-8890(99)00004-4

[19] Craig Reynolds. 2001. Boids. Background and Update. http://www.red3d.com/cwr/boids/. (2001). Accessed: 5 Dec 2017.

[20] Craig Reynolds. 2004. OpenSteer, steering behaviors for autonomous characters. (2004). http://opensteer.sourceforge.net/

[21] Craig W. Reynolds. 1987. Flocks, Herds and Schools: A Distributed Behavioral Model. *SIGGRAPH Comput. Graph.* 21, 4 (Aug. 1987), 25–34. https://doi.org/10.1145/37402.37406

[22] Craig W. Reynolds. 1999. Steering Behaviors For Autonomous Characters. In *Proceedings of Game Developers Conference 1999*. 763–782.

[23] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. 2003. Stable flocking of mobile agents part I: dynamic topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, Vol. 2. 2016–2021 Vol.2.

[24] Ashish Tiwari, Scott A. Smolka, Lukas Esterle, Anna Lukina, Junxing Yang, and Radu Grosu. 2017. Attacking the V: On the Resiliency of Adaptive-Horizon MPC. In *Automated Technology for Verification and Analysis - 15th International Symposium, ATVA 2017*. 446–462. https://doi.org/10.1007/978-3-319-68167-2_29

[25] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. 1995. Novel type of phase transition in a system of self-driven particles. *Physical review letters* 75, 6 (1995), 1226.

[26] Christian A Yates, Radek Erban, Carlos Escudero, Iain D Couzin, Jerome Buhl, Ioannis G Kevrekidis, Philip K Maini, and David JT Sumpter. 2009. Inherent noise can facilitate coherence in collective swarm motion. *Proceedings of the National Academy of Sciences* 106, 14 (2009), 5464–5469.

[27] Jingyuan Zhan and Xiang Li. 2011. Flocking of discrete-time multi-agent systems with predictive mechanisms. *IFAC Proceedings Volumes* 44, 1 (2011), 5669–5674.

[28] Jingyuan Zhan and Xiang Li. 2013. Flocking of multi-agent systems via model predictive control based on position-only measurements. *IEEE Transactions on Industrial Informatics* 9, 1 (2013), 377–385.

[29] Hai-Tao Zhang, Zhaomeng Cheng, Guanrong Chen, and Chunguang Li. 2015. Model predictive flocking control for second-order multi-agent systems with input constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers* 62, 6 (2015), 1599–1606.

[30] Tianhao Zhang, Gregory Kahn, Sergey Levine, and Pieter Abbeel. 2016. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*. 528–535. https://doi.org/10.1109/ICRA.2016.7487175

[31] Lifeng Zhou and Shaoyuan Li. 2017. Distributed model predictive control for multi-agent flocking via neighbor screening optimization. *International Journal of Robust and Nonlinear Control* 27, 9 (2017), 1690–1705.