Distributed Control for Flocking Maneuvers via Acceleration-Weighted Neighborhooding

Shouvik Roy[1], Usama Mehmood[1], Radu Grosu[2], Scott A. Smolka[1], Scott D. Stoller[1], and Ashish Tiwari[3]

[1]Department of Computer Science, Stony Brook University, Stony Brook, NY
[2]Cyber-Physical Systems Group, Technische Universität Wien, Austria
[3]Microsoft Research, Seattle, WA

***Abstract*— We introduce the concept of Distributed Model Predictive Control (DMPC) with *Acceleration-Weighted Neighborhooding* (AWN) in order to synthesize a distributed and symmetric controller for high-speed flocking maneuvers (angular turns in general). Acceleration-Weighted Neighborhooding exploits the imbalance in agent accelerations during a turning maneuver to ensure that actively turning agents are prioritized. We show that with our approach, a flocking maneuver can be achieved without it being a global objective. Only a small subset of the agents, called *initiators*, need to be aware of the maneuver objective. Our AWN-DMPC controller ensures this local information is propagated throughout the flock in a scale-free manner with linear delays. Our experimental evaluation conclusively demonstrates the maneuvering capabilities of a distributed flocking controller based on AWN-DMPC.**

I. Introduction

We have recently shown how the problem of flight formation for multi-agent systems can be formulated in terms of distributed model-predictive control (DMPC) [1]–[4]. Our results are specifically for V-formation and flocking, but our approach is applicable to any flight-formation problem that has a *cost-function characterization*. This simply means that the desired formation has been collectively attained when a (distributed) cost function is minimized.

An additional benefit of this cost-function-based approach is that in the case of flocking, it proved rather straightforward to derive additional cost-function terms that capture other flight-formation goals, including obstacle avoidance, predator avoidance, target-location seeking, and leader following. Due to the high-level nature of this approach, we sometimes refer to it as *declarative flocking* (DF), in contrast to the classic rule-based approaches in [5], [6].

This paper presents a significant extension of the DF framework in the form of a *symmetric and distributed controller for high-speed turning maneuvers*. Realizing this capability requires two significant extensions of DF and even DMPC itself. The first of these is what we refer to as DMPC with *Acceleration-Weighted Neighborhooding*. In AWN-DMPC, a Velocity Matching (VM) term is added to the flocking distributed cost function [1], causing an agent to produce an acceleration that aligns its velocity with that of its neighbors, with priority given to neighbors with larger accelerations in the previous time step. We show that this technique allows a turn initiated by a small number of agents (even just one) to propagate rapidly through the flock in a scale-free manner with linear delays. Our method does not require any communication between agents; agents interact only by sensing the positions and velocities of their neighbors.

The second extension of DF we introduce to achieve high-speed flocking maneuvers is, as alluded to in the above discussion of AWN-DMPC, the concept of *turn initiators*. As with starlings [7], a high-speed, flock-wide turning maneuver is initiated by a small number of agents in close spatial proximity to one another, typically located at the elongated tips of the flock.

As the initiators start to turn, their neighbors (thanks to AWN-DMPC) sense the change in their velocities and respond by velocity-aligning with them. Since the velocities of all agents in a stable, pre-turn flocking formation are essentially equal, even a small change in the initiators' velocities induces a significant velocity-matching effect in neighboring agents. The neighbors of the initiators then influence their own neighbors, and so on, causing the turning maneuver to propagate throughout the flock. Technically, initiator agents have an additional *target velocity* term in their cost functions, causing them to initiate the desired flock-wide turning maneuver.

We conducted an extensive experimental evaluation of our AWN-DMPC-based framework. The presented experiments are numerical simulations. Results are presented for flocks consisting of 20, 30, and 40 agents. Both single-turn maneuvers and maneuvers spanning a series of turns (e.g., u-turns and zigzag turns) are considered. Using a technique known as Statistical Model Checking, we also conducted a statistical analysis of our approach's convergence rate, for both flock convergence and maneuver convergence. The results establish high confidence in our approach.

The rest of the paper is organized as follows. Section II reviews the declarative flocking framework. Section III introduces our DMPC-based approach to high-speed flocking maneuvers. Sections IV showcases the results of our experimental evaluation. Section V presents our Statistical Model Checking results. Section VI considers related work. Section VII offers our concluding remarks and directions for future research.

II. Background

We consider a set of N dynamic agents $\mathcal{A} = \{1, \ldots, N\}$. Let $p_i(k) \in \mathbb{R}^3$, $v_i(k) \in \mathbb{R}^3$, $a_i(k) \in \mathbb{R}^3$ be the position, velocity, and acceleration of agent $i \in \mathcal{A}$ at time step k. The discrete-time equations of motion for agent i are:

$$\begin{aligned} p_i(k+1) &= p_i(k) + dt \cdot v_i(k), \quad |v_i(k)| < \bar{v} \\ v_i(k+1) &= v_i(k) + dt \cdot a_i(k), \quad |a_i(k)| < \bar{a} \end{aligned} \tag{1}$$

where $dt \in \mathbb{R}^+$ is the duration of the time step. The magnitudes of velocities and accelerations are bounded by $\bar{v}$ and $\bar{a}$, respectively. Acceleration $a_i(k)$ is the control input for agent i at time step k. The acceleration is updated after every ct time steps; i.e., $ct \cdot dt$ is the control period. The *neighborhood* of agent i, denoted by $\mathcal{N}_i \subseteq \mathcal{A}$, contains its $\mathcal{N}$-nearest neighbors, i.e., the $\mathcal{N}$ other agents closest to it.

A. Model Predictive Control

Model Predictive Control (MPC) [8] is a well-known control technique that has been applied to the flocking problem [1], [9], [10]. At each time step, an optimization problem is solved to find the optimal sequence of control actions (agent accelerations in our case) that minimizes a given cost function with respect to a predictive model of the system. The first control action of the optimal control sequence is then applied to the system; the rest is discarded. In the computation of the cost function, the predictive model is evaluated for a finite prediction horizon of T control steps.

MPC-based flocking models can be categorized as *centralized* or *distributed* [1]. A *centralized* model assumes that complete information about the flock is available to a single "global" controller, which uses the states of all agents to compute their optimal accelerations. The following optimization problem is solved by a centralized MPC controller at each time step k:

$$\begin{aligned} \underset{a(k|k),\ldots,a(k+T-1|k)}{\arg\min} \quad & \sum_{t=0}^{T-1} J(k+t \mid k) \\ + \quad & \lambda \cdot \sum_{t=0}^{T-1} \| a(k+t \mid k) \|^2 \end{aligned} \tag{2}$$

$J(k+t \mid k)$ is the predicted value of the centralized (global) cost function J at time step $k+t$ given the current state at time k. The first term is the sum of the centralized cost function, evaluated for T control steps (this embodies the predictive aspect of MPC), starting at time step k. It encodes the control objective of minimizing the cost function. The second term, scaled by a weight $\lambda > 0$, penalizes large control inputs; the $a(k+t \mid k)$'s are the predictions made at time step k for the (bounded) accelerations at time step $k+t$.

In *distributed* MPC, each agent computes its acceleration based only on its own state and its local knowledge, e.g., information about its neighbors. The following optimization problem is solved by a distributed MPC controller at each time step k:

$$\begin{aligned} \underset{a_i(k|k),\ldots,a_i(k+T-1|k)}{\arg\min} \quad & \sum_{t=0}^{T-1} J_i(k+t \mid k) \\ + \quad & \lambda \cdot \sum_{t=0}^{T-1} \| a_i(k+t \mid k) \|^2 \end{aligned} \tag{3}$$

$J_i(k+t \mid k)$ is the distributed cost function for agent i, analogous to $J(k+t \mid k)$ in the centralized case, and the $a_i(k+t \mid k)$'s are the predictions made at time step k for the accelerations of agent i at time step $k+t$. Distributed cost function $J_i(k)$ is based on the simplifying assumption that each of agent i's neighbors has zero acceleration during the prediction horizon T. Other approaches are possible, such as assuming that a neighboring agent maintains its acceleration at time step $k-1$ over the prediction horizon.

B. Declarative Flocking

Declarative flocking (DF) is a high-level approach to designing flocking algorithms based on defining a suitable cost function for MPC [1]. This is in contrast to the operational approach, where a set of rules are used to capture flocking behavior, as in Reynolds model [5], [6]. The cost function J^C for centralized DF at time k is:

$$J^C(k) = \frac{1}{\mathcal{Z}} \cdot \sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{A}, i<j} \omega_c \cdot \|p_{ij}\|^2 + \frac{\omega_s}{\|p_{ij}\|^2} \tag{4}$$

where $\|p_{ij}\|$ is the Euclidean distance between agents i and j, ω_c and ω_s are the weights of the cohesion and separation terms. The cost function is normalized by the number of agent pairs $\mathcal{Z} = \frac{N \cdot (N-1)}{2}$; as such, the cost does not depend on the size of the flock. Agent i's cost function J_i^D for distributed DF is computed over its set of neighbors $\mathcal{N}_i$ at time k:

$$J_i^{\mathrm{D}}(k) = \frac{\omega_c}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \|p_{ij}\|^2 + \frac{\omega_s}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \frac{1}{\|p_{ij}\|^2} \tag{5}$$

III. DMPC with Acceleration-Weighted Neighborhooding

This section presents our control-theoretic approach to flocking maneuvers based on the new concept of Acceleration-Weighted Neighborhooding (AWN). We formulate AWN in the context of our DMPC controller to achieve high-speed flight maneuvers, using only a subset of agents to initiate a maneuver.

Our AWN-DMPC formulation of flocking maneuvers is an extension of the DF framework presented in Section II-B. The AWN-DMPC cost function $J_i(k)$ for agent i (at time step k) is obtained by adding an AWN-based velocity matching (VM) term to $J_i^D(k)$ defined in Eq. 5.

$$J_i(k) = J_i^D(k) + \sum_{j \in \mathcal{N}_i} \gamma_{ij} \cdot \|v_{ij}\| \tag{6}$$

where γ_{ij} is the AWN-based weight for neighbor j of agent i, and $\|v_{ij}\|$ is the difference in velocities between agents i and j. The centralized version $J(k)$ of $J_i(k)$, which we will use in Section V for experimental evaluation purposes, can be similarly defined by adding the AWN-based VM term to $J^C(k)$ (Eq. 4).

To define the weights γ_{ij}, we use a *softmax* formulation based on the exponential function, normalized so that they sum to 1 over agent i's neighborhood $\mathcal{N}_i$. Weight γ_{ij} for neighbor j is defined as:

$$\gamma_{ij} \;=\; \frac{e^{\eta \cdot \Delta v_j(k)}}{\sum_{j \in \mathcal{N}_i} e^{\eta \cdot \Delta v_j(k)}} \tag{7}$$

where $\Delta v_j(k) = \|v_j(k) - v_j(k-1)\|$ is the change in the magnitude of agent j's velocity between the two most recent time-steps (a measure of its acceleration), and η is a large constant used to stabilize the softmax function by amplifying the values of $\Delta v_j(k)$. The factor η is needed as the magnitude of $\Delta v_j(k)$ is almost always less than 1; it thus needs to be amplified for the softmax function to generate weights large enough to create a preferential bias among the neighbors.

Let $J_i(k+t \mid k)$ denote the local cost computed by agent i at time step $k+t$ given the current state at time k. The optimization problem defined in Eq. 3 is solved by an AWN-DMPC controller at each control step k to generate the control action $a_i(k)$ for each agent i.

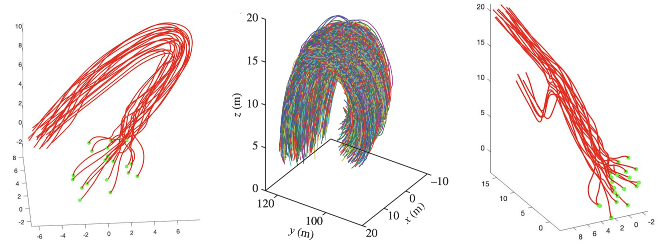
Turn Initiators: As with starlings [7], a high-speed, flock-wide turning maneuver is initiated by a small number of agents in close spatial proximity to one another, typically located at the sides of the flock. Moreover, these "initiators" are the only agents aware of the turn objective, including the desired turn angle. Since the velocities of all agents in a stable, pre-turn flocking formation are essentially equal, even a small change in the initiators' velocities causes a significant preferential imbalance in their neighboring agents. This forces these agents to velocity-align themselves with the initiators, leading to a cascading effect that spreads throughout the flock. The net effect is that the entire flock eventually (and rapidly) turns in the new direction.

Let agent $i \in \mathcal{I}$ be an initiator. The distributed cost function used by i to compute its accelerations upon the start of a turn is obtained by adding a weighted *target velocity* term $\omega_t \cdot \left\|v_i - \bar{v}_i^{\theta}\right\|$ to Eq. 6, where ω_t is the weight of the target-velocity term, v_i is initiator i's current velocity, and $\bar{v}_i^{\theta}$ is the average velocity of agent i's subflock when the turn is initiated, rotated by the desired turning angle θ. Agent i's subflock comprises i itself and its $\mathcal{N}$-nearest neighbors. The target velocity vector $\bar{v}_i^{\theta}$ for agent i is such that the angle between the new heading and the prior heading (average velocity vector of agent i's subflock at the start of the turn) is θ. The net effect of this new term is to put agent i on a trajectory towards completing a θ-degree turn with respect to the prior heading of its subflock.

IV. Experimental Evaluation

In our experimental setup, the AWN-DMPC control problem defined in Eqs. 3 and 6 is solved using the MATLAB fmincon optimizer. We use a flock of size $N = 20$ agents whose initial positions and velocities are uniformly sampled from $[-3,3]^3$ and $[0,1]^3$, respectively. The simulation time is 500 time steps, $dt = 0.05$, and $ct = 2$, where (recall) $ct \cdot dt$ is the control period. Agent velocity and acceleration bounds are $\bar{v} = 2.0$ and $\bar{a} = 1.0$.

We choose the set of initiators $\mathcal{I}$ at time-step $k = 200$, which is almost always enough time for the agents to have reached a flocking configuration. We use the following additional parameters: $|\mathcal{I}| = 4$, $|\mathcal{N}_i| = 6$, $\omega_c = 20$, $\omega_s = 30$, $\omega_t = 80$, $\eta = 100$, and $\theta = 170°$. To compute the target velocity $\bar{v}_i^{\theta}$ for an agent $i \in \mathcal{I}$, we generate a target velocity vector whose magnitude is the same as the magnitude of the average velocity vector of agent i's subflock immediately prior to the start of the turn, and the angle between these two vectors is θ. For experiments involving multiple turns, a new set of initiators is chosen at the beginning of each turn.



(a) Trajectories using AWN-DMPC (b) Starling Trajectories [7] (c) Trajectories w/out using AWN-DMPC

Fig. 1: Agent trajectories

Results: Fig. 1(a) depicts the trajectories of a 20-agent flock executing a 170° turn using AWN-DMPC. Fig. 1(b), from [11], shows the trajectories (recorded using high-speed video) for the same maneuver executed by a flock of 176 starlings in Rome, Italy. Fig. 1(c) depicts the failed trajectories of a 20-agent flock attempting a 170° turn using DMPC without AWN. The similarities between the two torus-shaped plots (Figs. 1(a) and (b)) are worth noting. Fig. 1(c) demonstrates the importance of AWN for the successful execution of the turning maneuver.

A pair of YouTube videos [12], [13] highlights the important role AWN plays in the successful execution of this turning maneuver. The first video shows the AWN-DMPC simulation that produced the flock-wide trajectories of Figure 1(a). The second video shows what happens when the AWN effect is turned off (something that is easily accomplished in our framework by setting η to zero in Eq. (7)). In this case, we have a traditional VM term with all neighbors treated equally. The effects are striking, with the initiators initiating the turn and then breaking away from the rest of the flock.

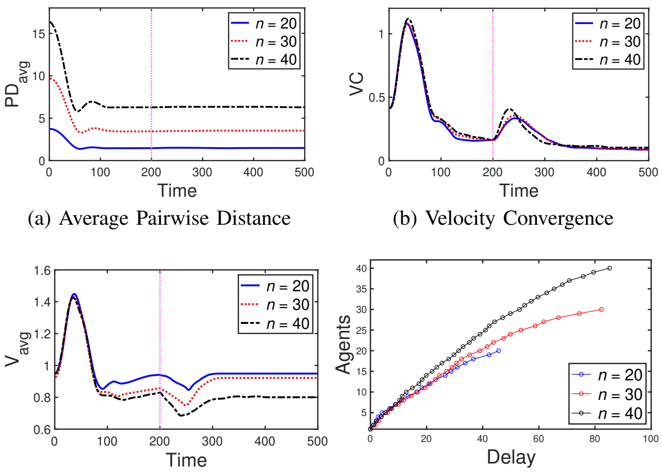
We use average pairwise distance and velocity convergence as performance metrics to evaluate the flocking quality of the AWN-DMPC controller. At any time step k, the average pairwise distance is:

$$PD_{avg}(k) = \frac{2}{N \cdot (N-1)} \sum_{(i,j) \in \mathcal{A}} \|p_{ij}\| \tag{8}$$

At any time step k, velocity convergence is defined as:

$$VC(k) = \frac{1}{N} \left(\sum_{i \in \mathcal{A}} \|v_i \;-\; \frac{1}{N} \sum_{j=1}^{N} v_j\|^2 \right) \tag{9}$$

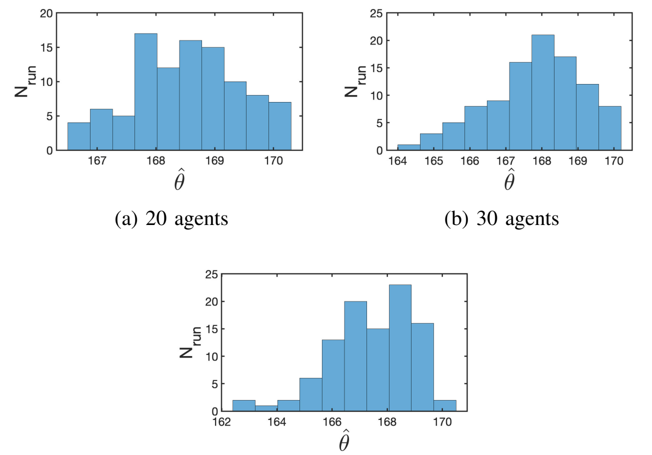
VC is the average of the squared magnitude of the discrepancy between each agent's velocities and the flock's average velocity. For these two metrics, lower values are better,



(a) Average Pairwise Distance (b) Velocity Convergence

(c) Average Velocity (d) Turning Delays

Fig. 2: Performance eval. of AWN-DMPC controller averaged over 100 runs. Turning maneuver commences at $k = 200$



(a) 20 agents (b) 30 agents

(c) 40 agents

Fig. 3: $\hat{\theta}$-distributions for AWN-DMPC, 100 runs, $\theta = 170°$

TABLE I: $\hat{\theta}$-statistics for AWN-DMPC, 100 runs, $\theta = 170°$

Agents	$\mu_{\hat{\theta}}$	$\sigma_{\hat{\theta}}$
20	168.53°	0.91
30	167.86°	1.34
40	166.91°	1.60

indicating a more coherent flock. All inter-agent distances, however, should be greater than d_{min}, the cutoff below which it is assumed that an inter-agent collision has occurred. Starting from a random initial configuration, a successful flocking controller ensures that the PD_{avg} and VC metrics stabilize for the flock. Similarly, such a controller ensures stabilization of these metrics for the post-maneuver flock.

Figs. 2(a)-(b) show the PD_{avg} and VC values for 20, 30, and 40 agents averaged over 100 runs with each runs executing a turning maneuver of 170° starting at time step $k = 200$. The vertical line in each of the plots indicates the start of the turn. There is no change in PD_{avg} after the start of the turn as the flock maintains its coherent structure. We also did not observe any flock fragmentation after initiating the turn. The VC value, however, shows a slight increase after the start of the turn, as the flock velocity undergoes a significant change due the flock's new heading. This effect is not long-lasting, with VC quickly converging once the turn is completed.

We use the average flock velocity V_{avg} and $\hat{\theta}$-distributions as metrics for evaluating AWN-DMPC's performance in executing flock-wide turns. Here $\hat{\theta}$ is the final turn angle achieved by the flock; i.e., the angle between the average velocity vector of the flock at the time of the turn and the average velocity vector of the flock at time step 500.

By plotting the average flock velocity V_{avg} before, during, and after a turn, we can show that the AWN-DMPC controller achieves *high-speed* flock maneuvers. In particular, Fig. 2(c) shows that V_{avg} slightly decreases after the start of the turn; the dip, however, is minimal and the flock quickly returns to its pre-turn velocity.

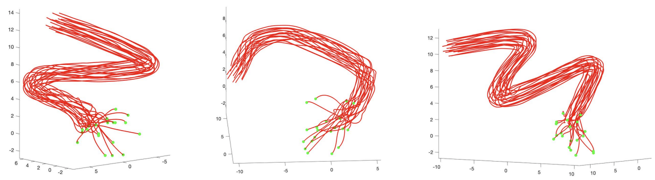
Fig. 2(d) plots the agent *turning delay* metric for flocks of 20, 30 and 40 agents, averaged over 100 runs. Let agent 0 be the first agent in the flock to initiate the turn, which it does at say time t_0. Then the turning delay for agent i is the time when i initiates its turn relative to t_0. Fig. 2(d) demonstrates that information about the maneuver propagates across the flock according to a *linear dispersion law*. Our results are consistent with the linear dispersion law observed in starling flocks during high-speed turns [11].

Fig. 3 shows the $\hat{\theta}$-distributions for 20, 30 and 40 agents where the desired flock turning angle $\theta = 170°$ and N_{run} is the number of runs. The $\hat{\theta}$-histograms (each divided into 10 bins) demonstrate that the final turn angles achieved by the flocks are statistically very close to θ. Table I captures the mean and standard deviation for the turn achieved by the flocks over 100 runs for 20, 30 and 40 agents.

Fig. 4 depicts the trajectories of a 20-agent flock executing *multiple turns*, thus demonstrating the wide-ranging maneuvering capabilities of an AWN-DMPC controller. In our experiments, although the initiators are chosen from among the agents located on the sides of the flock at the start of a turn, we also observed that successful flock-wide turns are possible independent of the distribution of the initiators.

V. Statistical Model Checking

We use Monte Carlo (MC) approximation as a form of Statistical Model Checking [14], [15] to compute confidence intervals for AWN-DMPC's flock convergence rate and maneuver convergence rate. The *flock convergence rate* is the fraction of successful flocks over M runs. We consider two types of flock convergence: *pre-maneuver flock convergence* (at time $k = 199$) and *post-maneuver flock convergence* (at time $k = 500$). The multi-agent system is said to have converged to a flock if $J(k) \leq \phi$, where recall that $J(k)$ is the centralized variant (see Section III) of the distributed cost function defined in Eq. 6. We used $\phi = 72$, a value obtained empirically by examining distribution plots for J similar in nature to those for $\hat{\theta}$ given in Fig. 3. The *maneuver convergence rate* is the fraction of successful maneuvers



(a) s-turn (b) u-turn (c) zigzag turn

Fig. 4: Agent trajectories with multiple turns

over M runs. A maneuver is considered to be successful if $\hat{\theta} \in \theta \pm \alpha$, where $\alpha = 5°$ is the angular error threshold.

The main idea of MC approximation is to use M random variables (RVs) $Z_1, \dots, Z_M$, also called samples, independent and identically distributed according to a random variable Z with mean μ_Z, and to take the expression $\tilde{\mu}_Z = (Z_1 + \dots + Z_M)/M$ as the value approximating the mean μ_Z. Since an exact computation of μ_Z is almost always intractable, an MC approach is used to compute an (ϵ, δ)-approximation of this quantity.

Additive Approximation [16] is an (ϵ, δ)-approximation scheme where the mean μ_Z of a RV Z is approximated with absolute error ϵ and probability $1 - \delta$:

$$Pr[\mu_Z - \epsilon \leq \tilde{\mu}_Z \leq \mu_Z + \epsilon] \geq 1 - \delta \qquad (10)$$

where $\tilde{\mu}_Z$ is an approximation of μ_Z. An important issue is to determine the number of samples M needed to ensure that $\tilde{\mu}_Z$ is an (ϵ, δ)-approximation of μ_Z. If Z is a Bernoulli variable whose value is expected to be large, one can use the Chernoff-Hoeffding instantiation of the Bernstein inequality and take M to be $M = 4\ln(2/\delta)/\epsilon^2$, as in [16]. This results in the *additive approximation algorithm* [15].

Algorithm 1: Additive Approximation Algorithm

Input: (ϵ, δ) with $0 < \epsilon < 1$ and $0 < \delta < 1$
Input: Random variables Z_i, IID
Output: $\tilde{\mu}_Z$ approximation of μ_Z
$M = 4\ln(2/\delta)/\epsilon^2$;
for *(i=0; i $\leq$ M; i++)* **do**
  $S = S + Z_i$;
$\tilde{\mu_Z} = S/M$; **return** $\tilde{\mu}_Z$;

We use Algorithm 1 to obtain (ϵ, δ)-approximations of AWN-DMPC's mean pre- and post-maneuver flock convergence rate ($\tilde{\mu}_{FC_1}$ and $\tilde{\mu}_{FC_2}$, respectively) and its mean maneuver convergence rate ($\tilde{\mu}_{MC}$). In each of these three cases, a sample Z_i is based on the result of executing the N-agent system starting from a random initial state. We take $Z = B$, where B is a Boolean variable indicating whether or not the agents converged accordingly. The assumptions about Z required for the applicability of the additive-approximation scheme hold, as RV B is a Bernoulli variable, the success rate is expected to be large (i.e., closer to 1 than to 0) in all three cases.

In our experiments using Algorithm 1, initial states are sampled from the same uniform random distributions used

TABLE II: SMC results for AWN-DMPC over M = 3,961 runs for $\theta = 170°$

Agents	$\tilde{\mu}_{FC_1}$	$\tilde{\mu}_{FC_2}$	$\tilde{\mu}_{MC}$
20	0.999	0.999	0.998
30	0.974	0.974	0.971
40	0.953	0.952	0.946

TABLE III: AWN-DMPC statistics for M = 3,961 runs, $\theta = 170°$

Agents	$\mu_{J_{pre}}$	$\sigma_{J_{pre}}$	$\mu_{J_{post}}$	$\sigma_{J_{post}}$	$\mu_{\hat{\theta}}$	$\sigma_{\hat{\theta}}$
20	70.17	0.35	70.18	0.35	168.51°	0.95
30	70.76	0.40	70.78	0.41	167.65°	1.41
40	71.14	0.46	71.15	0.46	166.83°	1.66

in Section IV. Also $\epsilon = 0.1$ and $\delta = 0.0001$, giving us M = 3,961 samples. We perform the required number of M simulations for 20, 30 and 40 agents.

Table II presents our SMC results, namely (ϵ, δ)-approximations $\tilde{\mu}_{FC_1}$, $\tilde{\mu}_{FC_2}$, and $\tilde{\mu}_{MC}$. The results demonstrate that AWN-DMPC performs very well over a range of flock sizes, even without changing the number of initiators or the size of an agent's neighborhood.

Table III presents the mean and standard deviation (over M runs) of the centralized cost function J upon pre-maneuver flock convergence (J_{pre}) and upon post-maneuver flock convergence (J_{post}). Also given are the μ and σ of the achieved turn angle $\hat{\theta}$. J_{pre} is computed at the time step prior to the start of maneuver ($k = 199$), and J_{post} and $\hat{\theta}$ are computed at the final time step ($k = 500$). The results of Table III demonstrate AWN-DMPC's high performance in executing substantial turning maneuvers over a range of flock sizes. They also showcase AWN-DMPC's ability to restabilize the flock post-maneuver.

VI. Related Work

A general distributed receding-horizon control framework for multi-agent systems is given in [17]. It does not specifically consider flocking or associated turning maneuvers. Prior work addresses the flock maneuvering problem using systems of dynamical equations to directly model such maneuvers. In [18], [19], a form of "distance-weighted neighborhooding" is used to generate agent accelerations; i.e., greater importance is given to closer neighbors. In contrast, we prioritize higher-accelerating neighbors. A flocking model for actual drones [20] incorporating an evolutionary optimization framework with carefully chosen order parameters and fitness functions is capable of performing flight maneuvers. The approach of [7], [11] analyzes the motion of flocks occurring in nature to create a model in the form of a Hamiltonian which follows a linear dispersion law for flock maneuvers. Multi-agent theory and finite-time control is used in [21] for flight maneuvers. They use finite-time disturbance observers to develop a finite-time distributed formation control strategy. Formation maneuver control is achieved in [22] using directed interaction graphs and leader-following.

VII. Conclusion

We introduced the concept of *acceleration-weighted neighborhooding* to synthesize a symmetric and distributed MPC controller for high-speed flocking maneuvers. Moreover, only a small number of agents are needed to initiate a maneuver, with the turning information propagating though the flock in a scale-free manner with linear delays. Our experimental evaluation used Statistical Model Checking to show that our controllers are capable of high-speed flocking maneuvers with high flock and maneuver convergence rates. We also demonstrated the *scalability* of our approach by successfully applying AWN-DMPC controllers on flocks of increasing size (number of agents).

For future work, we plan to implement AWN-DMPC on realistic quadrotor models. We also plan to apply AWN-DMPC to additional control objectives, such as predator and obstacle avoidance.

Acknowledgements: We thank the reviewers for their valuable comments. Research supported in part by NSF awards CPS-144683, CCF-1954837, CCF-191822, DCL-2040599 and ONR award N00014-20-1-2751.

References

[1] U. Mehmood, N. Paoletti, D. Phan, R. Grosu, S. Lin, S. D. Stoller, A. Tiwari, J. Yang, and S. A. Smolka, “Declarative vs rule-based control for flocking dynamics,” in *Proceedings of SAC 2018, 33rd Annual ACM Symposium on Applied Computing*, pp. 816–823, 2018.

[2] A. Lukina, A. Tiwari, S. A. Smolka, and R. Grosu, “Distributed adaptive-neighborhood control for stochastic reachability in multi-agent systems,” in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC ’19, (Limassol, Cyprus), pp. 914–921, ACM, 2019.

[3] U. Mehmood, S. Roy, R. Grosu, S. A. Smolka, S. D. Stoller, and A. Tiwari, “Neural flocking: MPC-based supervised learning of flocking controllers,” in *Foundations of Software Science and Computation Structures* (J. Goubault-Larrecq and B. König, eds.), (Cham), pp. 1–16, Springer International Publishing, 2020.

[4] S. Roy, U. Mehmood, R. Grosu, S. A. Smolka, S. D. Stoller, and A. Tiwari, “Learning distributed controllers for V-formation,” in *Proceedings of ACSOS 2020, First IEEE International Conference on Autonomic Computing and Self-Organizing Systems*, pp. 119–128, 2020.

[5] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *SIGGRAPH Comput. Graph.*, vol. 21, Aug. 1987.

[6] C. W. Reynolds, “Steering behaviors for autonomous characters,” in *Proceedings of Game Developers Conference 1999*, pp. 763–782, 1999.

[7] A. Attanasi, A. Cavagna, L. D. Castello, I. Giardina, A. Jelic, S. Melillo, L. Parisi, O. Pohl, E. Shen, and M. Viale, “Emergence of collective changes in travel direction of starling flocks from individual birds’ fluctuations,” *Journal of The Royal Society Interface*, vol. 12, 2015.

[8] E. F. Camacho and C. Bordons Alba, *Model Predictive Control*. Springer, 2007.

[9] H.-T. Zhang, Z. Cheng, G. Chen, and C. Li, “Model predictive flocking control for second-order multi-agent systems with input constraints,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 6, pp. 1599–1606, 2015.

[10] J. Zhan and X. Li, “Flocking of multi-agent systems via model predictive control based on position-only measurements,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 377–385, 2013.

[11] A. Attanasi, A. Cavagna, L. Castello, I. Giardina, T. Grigera, A. Jelić, S. Melillo, L. Parisi, O. Pohl, E. Shen, and M. Viale, “Information transfer and behavioural inertia in starling flocks,” *Nature physics*, vol. 10, pp. 615–698, 2014.

[12] S. Roy, “Flock maneuver with AWN.” `https://www.youtube.com/watch?v=_9z5LuPwF1I`, 2020.

[13] S. Roy, “Flock maneuver without AWN.” `https://www.youtube.com/watch?v=R-qIeQ0zc2E`, 2020.

[14] K. G. Larsen and A. Legay, “Statistical model checking: Past, present, and future,” in *6th International Symposium, ISoLA 2014*, (Corfu, Greece), Oct. 2014.

[15] R. Grosu, D. Peled, C. R. Ramakrishnan, S. A. Smolka, S. D. Stoller, and J. Yang, “Using statistical model checking for measuring systems,” in *6th International Symposium, ISoLA 2014*, (Corfu, Greece), Oct. 2014.

[16] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet, “Approximate probabilistic model checking,” in *Verification, Model Checking, and Abstract Interpretation* (B. Steffen and G. Levi, eds.), (Berlin, Heidelberg), pp. 73–84, Springer Berlin Heidelberg, 2004.

[17] W. B. Dunbar, *Distributed Receding Horizon Control of Multiagent Systems*. PhD thesis, California Institute of Technology, 2004.

[18] J. Ma, E. M. K. Lai, and W. Pang, “CS-CL: A flocking model that incorporates the bio-inspired chorus-line effect,” *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2018.

[19] J. Ma and E. M.-K. Lai, “Incorporating chorus line effect into a cucker-smale system for fast manoeuvre tracking,” in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS ’18, p. 2007–2009, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[20] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments,” *Science Robotics*, vol. 3, pp. 1–13, 07 2018.

[21] X. Zhang, J. Gao, W. Zhang, T. Zeng, and L. Ye, “Distributed formation control for multiple quadrotor based on multi-agent theory and disturbance observer,” *Mathematical Problems in Engineering*, vol. 2019, pp. 1–11, 2019.

[22] Y. Xu, S. Zhao, D. Luo, and Y. You, “Affine formation maneuver control of multi-agent systems with directed interaction graphs,” in *Proceedings of the 37th Chinese Control Conference (CCC)*, pp. 4563–4568, 2018.