

Flock navigation with dynamic hierarchy and subjective weights using nonlinear MPC

Nag, Aneek

Department of Electrical Engineering, Kyushu University

Huang, Shuo

Department of Electrical Engineering, Kyushu University

Themelis, Andreas

Department of Electrical Engineering, Kyushu University

Yamamoto, Kaoru

Department of Electrical Engineering, Kyushu University

<https://hdl.handle.net/2324/6625739>

出版情報 : 2022 IEEE Conference on Control Technology and Applications (CCTA), 2022. Institute of Electrical and Electronics Engineers: IEEE

バージョン :

権利関係 : © 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Flock navigation with dynamic hierarchy and subjective weights using nonlinear MPC

Aneek Nag

Shuo Huang

Andreas Themelis

Kaoru Yamamoto

Abstract—We propose a model predictive control (MPC) based approach to a flock control problem with obstacle avoidance capability in a leader-follower framework, utilizing the future trajectory prediction computed by each agent. We employ the traditional Reynolds’ flocking rules (cohesion, separation, and alignment) as a basis, and tailor the model to fit a navigation (as opposed to formation) purpose. In particular, we introduce several concepts such as the credibility and the importance of the gathered information from neighbors, and dynamic trade-offs between references. They are based on the observations that near-future predictions are more reliable, agents closer to leaders are implicit carriers of more educated information, and the predominance of either cohesion or alignment is dictated by the distance between the agent and its neighbors. These features are incorporated in the MPC formulation, and their advantages are discussed through numerical simulations.

I. INTRODUCTION

Nature often exhibits collective behavior, such as in a school of fish or a flock of birds, in which each member seemingly follows simple principles using only local information and still achieves a global task. This is an attractive feature when dealing with the control of a large number of autonomous vehicles, especially if each one has limited computational capability. In fact, in such a system each agent acts based solely on the information it receives (or detects) from other agents in its near proximity, without the need to be aware of the state of the entire flock.

In 1987, Reynolds introduced a model where each agent, referred to as a *boid*, obeys the following three simple rules to achieve *flocking* [9]: *cohesion*, promoting proximity among boids by biasing each one towards the local center of mass, *separation*, which imposes a minimum safety distance so as to prevent collisions, and *alignment*, encouraging agreement of velocity among neighboring boids. Because of its simplicity and scalability, Reynolds’ flocking has received much attention ever since, and many attempts have been made to employ this model for the joint control of large numbers of vehicles and mobile robots [1]. Each boid’s behavior can be captured by imposing an artificial potential field representing attractive/repulsive forces and velocity consensus to the neighbors; see, e.g., [7, 15, 16]. However, potential-field based methods tend to exhibit oscillatory behavior as each

agent reacts pretty much passively to the current state of the surroundings only. Instead, we may consider a “smarter” flock, in which each boid accounts for future stages and acts accordingly. This is precisely the idea behind MPC, a closed-loop control technique that provides optimal inputs by computing solutions of a finite-horizon problem at each sampling time, and then implementing the control law in accordance to a receding horizon strategy. In doing so, it can account for complex problem formulations, explicitly handles state constraints, and in considering long prediction horizons it favors smoother control actions. The applicability of MPC is however tightly linked to the efficiency of the optimization method employed to solve each problem, as well as the capability of the computing platform at hand. Nevertheless, the outstanding computing advancements and the development of efficient tailored algorithms [6, 8, 12, 14, 17] have made a reliable employment of this control technology a reality, even on embedded devices.

Attracted by the decentralized nature of flocks and the potential of MPC, several works for flock control using MPC strategies have been proposed in the last decade in which agents cooperatively exchange information to minimize the deviation from Reynolds’ rules throughout long prediction horizons [5, 13, 20, 21, 22, 23, 24].

A. Goals and proposed methodology

Inspired by this trend, in this paper we aim at investigating how (suitable modifications of) flocking rules can conveniently be exploited for *navigating* unknown environments towards a desired destination. Specifically, we consider the scenario in which the destination is unknown to some (possibly all but one) agents, yet the enforcement of a boids-type behavior allows the whole swarm information to be passively passed upon each single agent. In this regard, we refine the purpose of merely achieving *flocking*, as in forming and maintaining a stable configuration, and focus instead on a distributed *navigation* control. To seamlessly achieve this task, we introduce several factors to the proposed MPC formulation and to the flock modeling.

1) *Processing the neighbors’ predicted information*: Similarly to and inspired by [4, 13, 18], we propose an MPC-based approach where at each sampling time neighboring agents exchange their predicted configurations to facilitate the achievement of a cooperative task. To utilize such information wisely, we introduce a discount factor to reduce the influence of distant-future prediction as it is less accurate than the near-future one. Possible MPC horizons mismatch and delayed information are handled suitably.

Kyushu University, Department of Electrical Engineering, 744 Motooka, Nishi-ku, 819-0395 Fukuoka, Japan. This work is supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI grants n. JP19H02161 JP20K14766 and JP21K17710. The first author is supported by the Japanese government (MEXT) scholarship.
nag.aneek.364@s.kyushu-u.ac.jp, huangshuo0415@outlook.com,
andreas.themelis@ees.kyushu-u.ac.jp, yamamoto@ees.kyushu-u.ac.jp

2) *Flock modeling*: As a means to expedite information flow, we design a real-time adjustment of the flocking rules that, while retaining a fully decentralized nature, biases the instantaneous cohesion and alignment configurations to better reflect the navigation objective of the entire swarm. In particular, we introduce the following three adjustments to the original boids model: (i) time-varying leader-follower graph-distance-based hierarchy, (ii) subjective neighbor weighting and flocking references, and (iii) cohesion/alignment dynamic trade-off. We note that hierarchy structures in flocking have been studied in [11] for strict hierarchy (a lower-ranked agent does not affect a higher-ranked agent at all) and in [3, 11] for time-invariant hierarchy. We instead consider a non-strict and time-variant hierarchy that each agent can compute in a decentralized manner at negligible cost.

The paper focuses on the flock modeling, and assumes that the problem instances can be solved to *local* optimality in real time. While not tied to a particular optimization method, we point out the OpEn solver [12] as a candidate well suited to our purposes, which we use in the simulations.

B. Paper organization

Section II describes in detail the investigated setting and defines the MPC problems that each boid agent needs to solve. Technical considerations on the information exchange are also discussed. Section III presents the proposed dynamic modifications of the flocking rules, elaborating on their *hierarchical* and *subjective* nature. In Section IV we showcase the advantages of the proposed modifications with numerical simulations. Section V concludes the paper.

II. PROBLEM SETTING

Our goal is the design of a distributed control strategy for a fleet of N agents, divided into *leaders*, that know (or are in charge of planning) the trajectory to reach a desired destination, and *followers*, which are instead only aware of their immediate surroundings. Specifically, we aim at controlling these latter ones under the assumption that the leaders dispatch their near future trajectory and velocity predictions to the followers, yet without necessarily taking into account information received from them. Desirable features of the control action include minimizing the time to reach destination, as well as the number of required leaders and the amount of constraints on their motion for ensuring that no follower is left behind. The revisited flocking rules developed in Section III are designed in this perspective.

A. Notational conventions

We bold-face variables indexed along the time steps spanning the prediction horizon, e.g., $\mathbf{u}_i^t = \mathbf{u}_i^{t|t} \dots \mathbf{u}_i^{t+T_i-1|t}$ shall denote a sequence of inputs of agent i and similarly \mathbf{x}_i^t will represent a sequence of states throughout the prediction horizon T_i of agent i . At every time instant t , agent i is aware of a set of obstacles \mathcal{O}_i^t in its proximity and receives a sequence of predicted outputs \mathbf{y}_{ji}^t from each of its current in-neighbors $j \in \mathcal{N}_i^t \setminus \{i\}$. Outputs comprise position \mathbf{p}_{ji}^t and velocity \mathbf{v}_{ji}^t ; the indexing ji accounts for possible mismatches in the same information collected by different agents.

B. MPC formulation

The discrete time step is considered normalized as $\Delta t = 1$ time units for notational simplicity. At each time step t , agent i needs to solve the following problem

$$\begin{aligned} & \underset{\mathbf{u}_i^t \in \mathcal{U}_i \times \dots \times \mathcal{U}_i}{\text{minimize}} && J(\mathbf{u}_i^t) \\ & \text{subject to} && \left\{ \begin{array}{l} \mathbf{x}_i^{t+k+1|t} = \mathbf{f}_i(\mathbf{x}_i^{t+k|t}, \mathbf{u}_i^{t+k|t}) \\ \mathbf{x}_i^{t+k+1|t} \in \mathcal{X}_i \\ \mathbf{p}_{ji}^{t+k+1|t} \notin \mathcal{O}, \quad \mathcal{O} \in \mathcal{O}_i^t \\ d_{ji}^{t+k+1|t} \geq d_{i,\text{sep}}, \quad j \in \mathcal{N}_i^{t+k+1|t} \end{array} \right. \quad \begin{array}{l} k < T_i \\ k < T_{i,\text{sep}} \end{array} \end{aligned} \quad (1a)$$

where $\mathbf{f}_i : \mathbb{R}^{n_{x,i}} \times \mathbb{R}^{n_{u,i}} \rightarrow \mathbb{R}^{n_{x,i}}$ are agent i 's discrete state dynamics, $\mathcal{U}_i \subseteq \mathbb{R}^{n_{u,i}}$ and $\mathcal{X}_i \subseteq \mathbb{R}^{n_{x,i}}$ are input and state constraint sets, and $T_{i,\text{sep}} < T_i$ and $d_{i,\text{sep}} > 0$ are modeling parameters that account for the separation rule (see Section II-D). Furthermore, at every time step $\tau \in [t+1, t+T_i]$,

$$d_{ji}^{\tau|t} := \|\mathbf{p}_i^{\tau|t} - \mathbf{p}_{ji}^{\tau-1|t}\|^2$$

is the predicted (square) distance from neighbor j , $\bar{\mathbf{y}}_i^{\tau|t}$ is a reference output comprising of vectors

$$\bar{\mathbf{p}}_i^{\tau|t} := \sum_{j \in \mathcal{N}_i^{\tau|t}} w_{ji}^{\tau|t} \cdot \mathbf{p}_{ji}^{\tau|t} \quad \text{and} \quad \bar{\mathbf{v}}_i^{\tau|t} := \sum_{j \in \mathcal{N}_i^{\tau|t}} \omega_{ji}^{\tau|t} \cdot \mathbf{v}_{ji}^{\tau|t}, \quad (2)$$

namely weighted averages of positions and velocities among the available neighbors $\mathcal{N}_i^{\tau|t}$ with (positive, summing to one) cohesion and alignment weights $w_{ji}^{\tau|t}$ and $\omega_{ji}^{\tau|t}$ (cf. Section III-B), and

$$J(\mathbf{u}_i^t) = \|\mathbf{u}_i^t\|_{R_i}^2 + \sum_{k=0}^{T_i-1} \gamma_i^k \|\mathbf{y}_i^{t+k+1|t} - \bar{\mathbf{y}}_i^{t+k|t}\|_{Q_i^t}^2 + \rho_{i,\text{sep}} \sum_{k>T_{i,\text{sep}}} \gamma_i^k \sum_{j \in \mathcal{N}_i^{t+k|t} \setminus \{i\}} d_{ji}^{t+k|t}$$

is the objective. Here, $\gamma_i \in (0, 1]$ is a discount factor, $Q_i^t \in \mathbb{R}^{4 \times 4}$ and $R_i \in \mathbb{R}^{n_{u,i} T_i \times n_{u,i} T_i}$ are (positive semidefinite, diagonal) weight matrices, and $\rho_{i,\text{sep}} \geq 0$ is a penalty parameter for soft-constraining (or neglecting, if chosen null) separation at late stages. The overall information exchange and data processing is synopsized in Algorithm 1.

Algorithm 1 Conceptual MPC workflow

At the beginning of stage t , boid agent i

- 1: DETECTS obstacles \mathcal{O}_i^t and in-neighbors \mathcal{N}_i^t (including itself) collecting their last
 - hierarchy level π_j^{t-1} and
 - time-stamped predicted outputs \mathbf{y}_{ji}^{t-1} , $j \in \mathcal{N}_i^t$
 - 2: SHIFTS neighbors outputs based on discrepancy with own time-stamp and/or MPC horizon, and refines future neighbors $\mathcal{N}_i^{t+k|t}$, $k = 1, \dots, T_i - 1$, as in §II-C
 - 3: estimates OWN HIERARCHY LEVEL π_i^t as in (3)
 - 4: assigns INDIVIDUAL WEIGHTS $w_{ji}^{t+k|t}$ and $\omega_{ji}^{t+k|t}$ as in §III-B, $k = 0, \dots, T_i - 1$, $j \in \mathcal{N}_i^{t+k|t}$, thus creating
 - reference positions $\bar{\mathbf{p}}_i^{t+k|t}$ and
 - reference velocities $\bar{\mathbf{v}}_i^{t+k|t}$ as in (2)
 - 5: selects COHESION/ALIGNMENT TRADE-OFF WEIGHT Q_i^t as in (9)
 - 6: COMPUTES OPTIMAL INPUTS \mathbf{u}_i^t by solving problem (1), and store predicted outputs \mathbf{y}_i^t
 - 7: ACTUATES $\mathbf{u}_i^{t|t}$, $t \leftarrow t + 1$, and RESTARTS from step 1
-

C. Horizons mismatch and time delays

The dependency on τ for the set of neighboring agents $\mathcal{N}_i^{\tau|t}$ accounts for the possibility of different prediction horizons T_i among neighbors and/or delayed information. The rationale is as follows. If agent i receives information from an agent $j \in \mathcal{N}_i^{\tau|t}$ which has a longer prediction horizon $T_j > T_i$, then any prediction from step T_i onward can be discarded. In case $T_j < T_i$, instead, agent j can be removed from the *virtual* set of neighbors $\mathcal{N}_i^{\tau|t}$ for $\tau \geq T_j$, since no information is anymore available after that time.

The strategy of *virtually* removing neighbors in the prediction can also be adopted to account for delayed information. If agent i receives information from agent j with a delay of δ time steps, then the received output sequence is $\mathbf{y}_{ji}^{t-\delta}$, of which only the last $T_i - \delta$ elements will be taken into account, leading to $\mathbf{y}_{ji}^t = \mathbf{y}_{ji}^{t-\delta}, \dots, \mathbf{y}_{ji}^{t-\delta+T_i|t-\delta}$, at which point any shortcomings can be accounted for by suitably removing agent j from late virtual neighborhoods.

Note that, by adopting the convention that $i \in \mathcal{N}_i^{\tau|t}$ for all τ , in absence of external information the average $\bar{\mathbf{y}}_i^{\tau|t}$ reduces to agent i 's previously predicted output, so that the term $\|\mathbf{y}_i^{t+k|t} - \bar{\mathbf{y}}_i^{t+k|t}\|_{\mathcal{Q}_i}^2$ in the cost function promotes consistency with the information that was dispatched to the neighbors.

D. Discount factor and late-stage soft constraints

In accordance with the receding horizon principle of MPC, although a sequence of T_i optimal inputs is computed every time it is only the first one that is actuated. This strategy follows the logic that predictions on late stages are much less accurate and more susceptible to environmental changes, which is why the problem is continuously updated to take new information into account. Following this spirit and as a way to prevent too much conservatism, the separation rule (i.e., collision avoidance) is enforced only on early time instants, and is otherwise only discouraged as a soft penalty in the cost function for later stages with a quadratic term $\rho_{i,\text{sep}} d_{ji}^{\tau|t}$. For the same reason, a discount factor γ_i^k , with $\gamma_i \in (0, 1]$ weighs the k -th per-stage cost in the objective so as to prioritize early stages in the decision process and give less importance to longer term estimates which are likely less accurate. Avoidance of detected obstacles, instead, is hard-constrained for the entire prediction horizon, owing to the fact that obstacles are assumed static and thus not subject to (unknown) changes in position.

III. FLOCKING RULES REVISITED

In this section we propose some adaptations of the classical flocking rules that are better suited for navigating in an unknown environment. The advantages of each of them will be showcased with simulations in the following section.

A. Leader-follower graph-distance hierarchy

Because of their predominant role and more accurate and longer term spatial information, it is reasonable to give more importance to data received from leaders. For the same reason, followers that are directly connected to a leader implicitly benefit for more educated estimates than those which are

not, and their dispatched data should likewise be given more weight. As a way to enforce this chain of priority, we may endow the connectivity graph of a weight on the (directed) edges based on the graph distance from the closest leader. Although each follower is only aware of a limited neighborhood and this quantity may thus not be readily available, it is nevertheless easy to well estimate it through the information exchanges. Each leader ℓ is assigned a hierarchy level $\pi_\ell^t \equiv 0$, constant over time, while each follower i , starting from a predefined upper bound $\pi_i^0 = \bar{\pi} \in \mathbb{N}_{\geq 1}$, updates its value over time as

$$\pi_i^t = \min \left\{ \bar{\pi}, 1 + \min_{i \neq j \in \mathcal{N}_i^t} \pi_j^{t-1} \right\}. \quad (3)$$

The cap $\bar{\pi}$ is a model parameter, possibly different among different agents and whose role is merely to prevent arbitrarily low weights. Although the rule is only based on local information, it is easy to see that it accurately estimates the graph distance from the set of leader agent nodes in few sampling times. This claim can be substantiated with a trivial induction. To this end, let $(\mathcal{V}, \mathcal{E}^t)$ denote the connectivity graph at time t among agents \mathcal{V} , where $(j, i) \in \mathcal{E}^t \subseteq \mathcal{V} \times \mathcal{V}$ iff $j \in \mathcal{N}_i^t$. The graph distance between agents $i, j \in \mathcal{V}$ at time t is defined as

$$\text{dist}^{\mathcal{E}^t}(i, j) = \min \left\{ \eta \mid \exists (i, j_0), (j_0, j_1), \dots, (j_{\eta-1}, j) \in \mathcal{E}^t \right\},$$

while the *exact* hierarchy level of i at time t as

$$D_i^{\mathcal{E}^t} = \min \left\{ \bar{\pi}, \text{dist}^{\mathcal{E}^t}(i, \ell) \mid \ell \in \mathcal{L} \right\},$$

where $\mathcal{L} \subseteq \mathcal{V}$ is the set of leader agents and $\bar{\pi}$ is a chosen upper bound as in (3).

Lemma 1. *Suppose that the delay in exchanging information is bounded by $\delta \geq 0$ timesteps, and let $D \in \{0, \dots, \bar{\pi}\}$. If \mathcal{E}^t has not changed in the last $(\delta + 1)D$ steps before time \bar{t} , then for every agent $i \in \mathcal{V}$*

$$\pi_i^{\bar{t}} \begin{cases} = D_i^{\mathcal{E}^{\bar{t}}} & \text{if } D_i^{\mathcal{E}^{\bar{t}}} \leq D \\ > D & \text{otherwise.} \end{cases} \quad (4)$$

Proof. We proceed by induction on D . If $D = 0$, then the claim holds trivially, since $D_i^{\mathcal{E}^t} = 0$ holds iff i is a leader and is strictly greater otherwise. Suppose that the claim holds for $D - 1 \in [0, \bar{\pi} - 1]$, and suppose that \mathcal{E}^t remained constant in the last $(1 + \delta)D$ steps before \bar{t} . Let $i \in \mathcal{V}$; without loss of generality, $D_i^{\mathcal{E}^{\bar{t}}} \geq D$, for otherwise there is nothing to show. For $j \in \mathcal{N}_i^{\bar{t}}$, let δ_{ji} be the delay with which i receives the information from j at time \bar{t} , and observe that necessarily $D_j^{\mathcal{E}^{\bar{t}}} \geq D - 1$. Since $\delta_{ji} + 1 \leq (\delta + 1)D$, it follows from the induction that

$$\pi_j^{\bar{t}-\delta_{ji}-1} \begin{cases} = D_j^{\mathcal{E}^{\bar{t}}} & \text{if } D_j^{\mathcal{E}^{\bar{t}}} \leq D - 1 \quad (\Leftrightarrow D_j^{\mathcal{E}^{\bar{t}}} = D - 1) \\ \geq D & \text{otherwise,} \end{cases} \quad (5)$$

where we used the assumption that $\mathcal{E}^{\bar{t}-\delta_{ji}-1} = \mathcal{E}^{\bar{t}}$. Then, by the update rule (3),

$$\pi_i^{\bar{t}} = \min \left\{ \bar{\pi}, 1 + \min_{i \neq j \in \mathcal{N}_i^{\bar{t}}} \pi_j^{\bar{t}-\delta_{ji}-1} \right\} \geq D. \quad (6)$$

If $D_i^{\mathcal{E}^{\bar{t}}} = D$, then there exists $j \in \mathcal{N}_i^{\bar{t}}$ with $D_j^{\mathcal{E}^{\bar{t}}} = D - 1$. If instead $D_i^{\mathcal{E}^{\bar{t}}} > D$, then $D_j^{\mathcal{E}^{\bar{t}}} \geq D$ holds for all $j \in \mathcal{N}_i^{\bar{t}}$. In both cases, (4) follows by combining (5) and (6). \square

Note that including $\bar{\pi}$ in the minimum also accounts for possible instances of isolatedness of agent i or detachedness of the cluster it belongs from a leader, while excluding π_i^{t-1} is necessary in order to guarantee an up-to-date (as opposed to a best-so-far) hierarchy estimate. The values of the own and neighbors' hierarchy levels are used to customize cohesion references, as described next.

B. Subjective neighbor weighting and flocking references

In order to promote a flow towards the leader, each follower i can bias the reference center of mass dictating the cohesion rule based on the hierarchy level of all agents in its neighborhood, including itself. To this end, simplifying the notation as $\pi_j := \pi_i^t$ if $j = i$, and $\pi_j := \pi_j^{t-1}$ otherwise, cohesion weights

$$w_{ji}^{\tau_{lt}} = \frac{2^{-\pi_j}}{\sum_{\ell \in \mathcal{N}_i^{\tau_{lt}}} 2^{-\pi_\ell}} \quad (7)$$

can be chosen to bias $\bar{p}_i^{\tau_{lt}}$ as in (2).

While the same can be done for the alignment weights $\omega_{ji}^{\tau_{lt}}$, a more refined criterion can be implemented to discourage unwanted behaviors. For instance, in the Cucker-Smale model [2, 11, 23], ω_{ji} is inversely proportional to the distance between agents i and j . While the rationale is perfectly reasonable and intuitive, when navigating an obstructed environment it is hardly helpful, as will be showcased in the simulations section. Differently from flocking, navigation presupposes a heading direction that is not the result of an agreement among agents, and a navigation-friendly alignment rule should be able to reflect this bias. Suppose that neighboring agent j is traveling *ahead* of agent i , and changes direction because of an obstacle in its way. In biasing its velocity accordingly, agent i is better prepared for avoiding the same obstacle in a later stage, even in case it is not currently detected. On the contrary, if agent j is *behind*, then the obstacle is of no concern to agent i , at least in the near future, and indulging into a possibly abrupt violation of the flow is not desirable in this case. Given that followers obey a flocking behavior and are unaware of the entire flock configuration, notions of *ahead* and *behind* are merely based on agents' instantaneous velocity. In this context, we may define subjective orientations by comparing the displacement between agents with the individual movement direction, and say that, at time t and relative to agent i ,

$$j \in \mathcal{N}_i^{\tau_{lt}} \begin{cases} \text{is ahead} & \text{if } \langle v_i^{\tau_{lt}}, p_{ji}^{t-1|t-1} - p_i^{\tau_{lt}} \rangle \geq 0 \\ \text{is behind} & \text{otherwise.} \end{cases}$$

(The time difference between own and neighbor j 's positions $p_i^{\tau_{lt}}$ and $p_{ji}^{t-1|t-1}$ reflects the fact that data exchange among agents happens only once before the MPC problem is addressed; cf. workflow in Algorithm 1.) Once a behind weight $\omega_{i,b} \in [0, 1]$ is fixed, we may define alignment weights as

$$\omega_{ji}^{\tau_{lt}} := \begin{cases} 1 & \text{if } j \text{ is ahead wrt } i \\ \omega_{i,b} & \text{otherwise} \end{cases} \quad (8)$$

to bias $\bar{v}_i^{\tau_{lt}}$ as in (2). Our experiments suggest that values $\omega_{i,b} \leq 0.2$ tend to work well in all scenarios.

C. Cohesion/alignment dynamic trade-off

Once the reference positions and velocities are determined, it remains to decide if and how much tracking either quantity is to be favored. This is done by suitably choosing the weight matrix Q_i^t in the cost function J (recall that y_i^t comprises position and velocity vectors). If agent i is far from the reference center of mass $\bar{p}_i^{\tau_{lt}}$, aligning with distant agents serves no purpose; in this case, *cohesion* should be the priority reference. On the contrary, if agent i is very close to the center of mass, then cohesion is already well achieved and sticking to neighboring agents' *alignment* becomes more reasonable. To dynamically adjust this trade-off, a Cucker-Smale-type weight rule [2] can be adopted, e.g.,

$$Q_i^t := \begin{pmatrix} 1 - q_i^t & & \\ & 1 - q_i^t & \\ & & q_i^t \\ & & & q_i^t \end{pmatrix} \quad \text{with} \quad q_i^t := \frac{q_{i,\text{st}}}{1 + c_i \|p_i^{\tau_{lt}} - \bar{p}_i^{\tau_{lt}}\|^2}, \quad (9)$$

for some $c_i > 0$ and $q_{i,\text{st}} \in (0, 1)$; by setting $c_i = 0$ one recovers a conventional *static* trade-off $q_i^t \equiv q_{i,\text{st}}$. Although with the same principle stage-dependent weights $Q_i^{\tau_{lt}}$ could also be considered, experimental evidence has shown that this modification hardly makes any difference. This is somewhat expected, given that within short sampling times the changes in position and velocity are negligible. Note that, although based on the same distance inverse proportionality principle, this weighting rule is substantially different from the original Cucker-Smale criterion. In fact, while the latter is used to determine which velocity to align to, the proposed weight (9) quantifies the importance of said reference.

IV. SIMULATIONS

To demonstrate the advantages of the purpose-oriented rules introduced in Section III, we consider a system of planar point-mass boids with double integrator dynamics and one leader which navigate obstructed environments. Except for the scenario depicted in Figure 3, which considers (the shortcomings of) a single-stage formulation, MPC horizons are set to $T_i = 8$ for all boids, while separation horizons as $T_{i,\text{sep}} = 4$; avoidance of predicted collisions at stages higher than $T_{i,\text{sep}}$ is soft constrained with penalty $\rho_{i,\text{sep}} = 100$. The alignment weights of behind agents as in (8) are set to $\omega_{i,b} = 0.2$, while in the trade-off rule (9) we use a default static coefficient $q_{i,\text{st}} = 0.5$ and a dynamic weight $c_i = 2$. An upper cap $\bar{\pi} = 10$ is chosen for the hierarchy levels as in (3), $\gamma_i = 0.5$ is used as discount factor in the cost, and $\Delta t = \frac{1}{40}$ s is set as sampling time.

The leader shares its predicted positions and velocities, but is otherwise indifferent to other agents. We consider a proximity-based connectivity, i.e., $j \in \mathcal{N}_i^t$ iff $\|p_i^t - p_j^t\| \leq r$ for some detection radius r , assumed the same for all agents for simplicity. We consider bounded box constraints with radius 2 for (acceleration) inputs and velocity states, and obstacles of the form $O = \{x \in \mathbb{R}^2 \mid h^O(x) < 0\}$ for some smooth mapping $h^O : \mathbb{R}^2 \rightarrow \mathbb{R}^{n_o}$.

A. Implementation details

The experiments are run in Python, and the MPC problems are solved with the parametric optimizer OpEn [12]

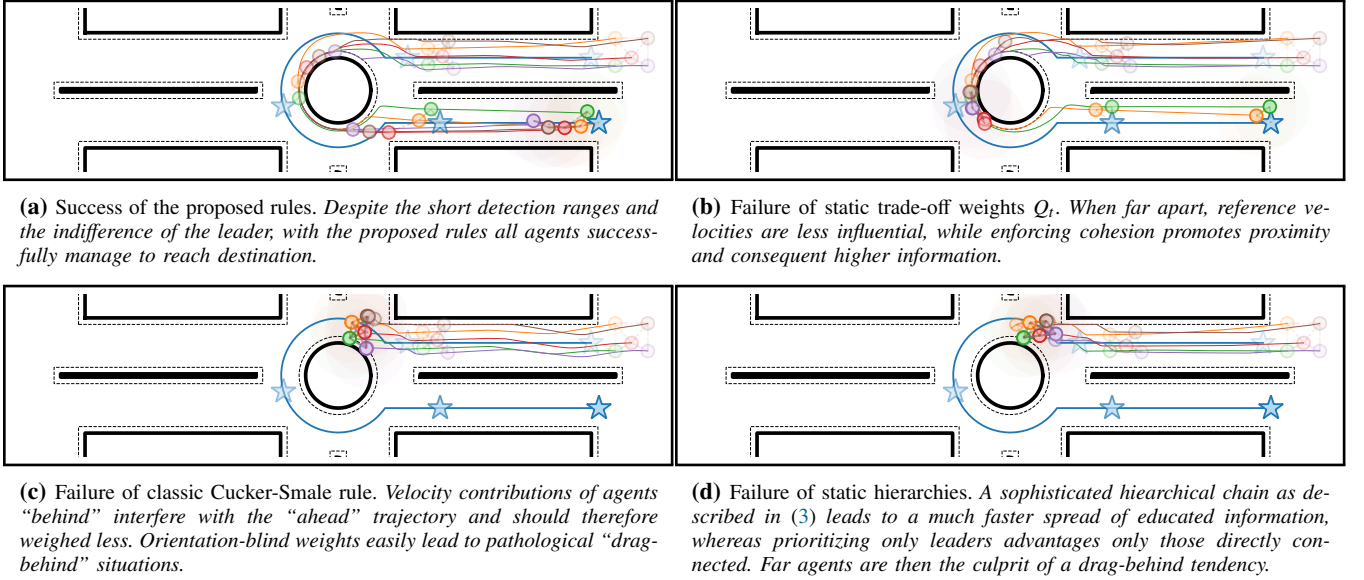


Fig. 1: U-turn at a roundabout. A leader travelling at a constant speed can be well tracked by all followers till destination. Neglecting some of the rules may result in disconnected agents left behind.

with direct interface. Following [10], avoidance of obstacle O is modeled as the smooth and real-valued equation $(\prod_{i=1}^{n_o} \min\{0, h_i^O\})^2 = 0$, and enforced with OpEn’s penalty method, while velocity constraints and collision avoidance are enforced with the augmented Lagrangian method. Obstacles are slightly enlarged as a safety precaution; the enlargements are represented by dashed lines in the figures. Parameters passed to the solver are initial state (position and velocity) $s_i^{it} \in \mathbb{R}^4$, cohesion and alignment references $\bar{p}_i^t, \bar{v}_i^t \in \mathbb{R}^{2T}$, predicted positions of the $J = 5$ closest neighbors (excluding i itself) $\mathbf{p}_{ji}^t \in \mathbb{R}^{2T}$, and cohesion/alignment trade-off weight q_i^t , which we saturate in $[0.2, 0.8]$ as numerical evidence suggests that too extreme trade-offs may lead to erratic behaviors. The limit on the number of neighbors to separate from is imposed due to modeling restrictions, in account of the non variable size of the parameter vector. For the same reason, when $\mathcal{N}_i^{it} \setminus \{i\}$ has less than J agents, parameters are filled with 10^4 distance units for dummy positions.

B. Problems description

We consider two scenarios that are hostile to conventional flocking control problems; one is taking a U-turn at a roundabout, and the other is going through an obstructed narrow passage. In these situations, the traditional flocking rules may not be meaningful or even disadvantageous, as they discourage changes in the formation which are instead essential in navigation: a U-turn imposes a quick change of direction subject to road constraints, i.e., lanes orientation and boundaries, while obstructed environments make it easier for boids to get stuck by an obstacle and then either dragging back the rest of the flock or ending up disconnected from it. Instead, in the sequel we demonstrate that the proposed rules well cope with these scenarios, even when tracking an imprecise

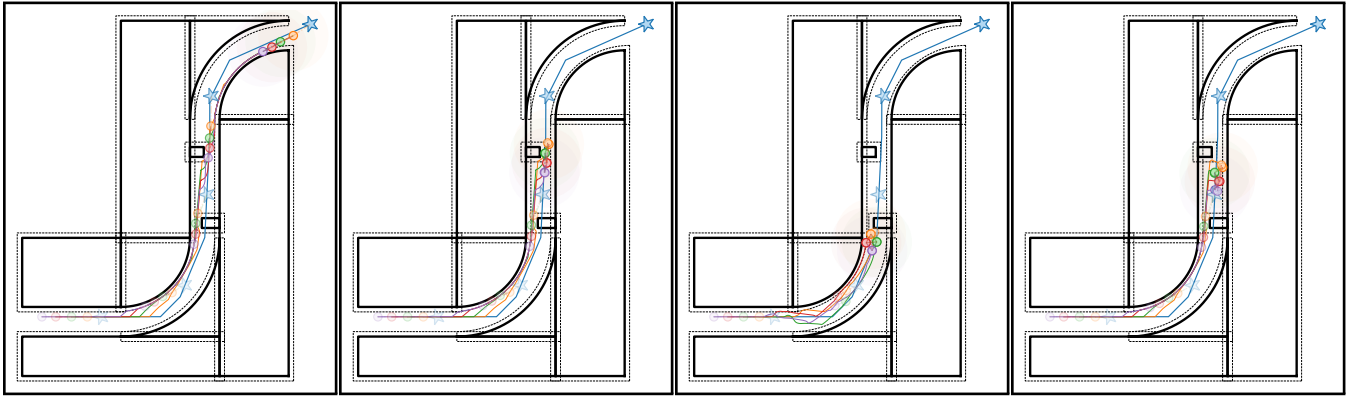
leading reference. Indeed, when any of the rules is inactive, the flock fails to reach the destination, while the complete method successfully achieves the task in both cases.

The leader agent is denoted with a blue star, while the followers with colored circles. The leader proceeds along a predefined path at a velocity compatible with the followers’ limits, but is otherwise indifferent to their states. Snapshots of the configurations at different time instants are shown, where faded colors indicate the earlier time. Agents are assumed dimensionless point particles positioned in the center of the circles, the radiuses of which indicate the separation distance $d_{i,sep}$. Dashed lines between agents indicate the (time-varying, distance-based) connectivity graph; the communication ranges, depicted as transparent circles in the last snapshot, are set very small to make the problem more challenging. Obstacles are represented by a combination of rectangles and circles, shown in black, with respective enlargements as dotted lines. Results of the simulations are described in the captions of the respective Figures 1, 2 and 3.

Only few agents are considered for the sake of graphical clarity. We however emphasize that the advantage of the proposed rules is more evident when considering large fleets, which are more prone to exhibiting disconnections in the communication graph.

V. CONCLUSIONS

A distributed control strategy using MPC has been proposed for a flock navigation problem in unknown obstructed environments. Conventional flocking rules have been suitably modified, thus introducing a dynamic hierarchy strategy, subjective neighbor weighting, and cohesion/alignment dynamic trade-off that are all computed using local information only. Numerical examples demonstrate that these rules effectively guide a flock in challenging environments, and



(a) Success of the proposed rules. All agents seamlessly manage to navigate the impervious environment until destination. (b) Failure of static trade-off weights Q_1 . At the second obstacle, the flock loses momentum by not prioritizing cohesion. (c) Failure of classic Cucker-Smale rule. “ α -lattice” formations [7] resulting from traditional flocking are not fit for obstructed environments. (d) Failure of static hierarchies. Similarly to the situation in Fig. 1d, trailing agents drag the flock behind, unless a smart hierarchy is established.

Fig. 2: Tracking an inaccurate virtual reference through an obstructed narrow passage. The leader is here a virtual reference which is not accurately detected or computed by the followers. As a result, it violates the first obstacle on the way.

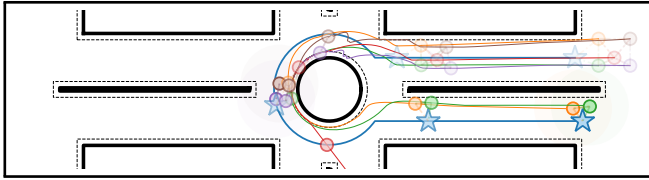


Fig. 3: Importance of multi-stage control. The same round-about scenario of Figure 1 fails without taking future predictions into account. In this example, the red follower loses all its neighbors and ends up pursuing a wrong direction.

without imposing severe restrictions on leaders’ trajectories. Possible future directions include accounting for noisy/faulty communication and investigating optimal cooperative leaders’ control actions. It would also be interesting to extend the analysis of (static) *layered-path graphs* of [19] to our dynamic hierarchy setting, so as to develop strategies to minimize connectivity loss in case of agents leaving the flock.

REFERENCES

- [1] L.E. Beaver and A.A. Malikopoulos. An overview on optimal flocking. *Annual Reviews in Control*, 51:88–99, 2021.
- [2] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007.
- [3] Y. Jia and T. Vicsek. Modelling hierarchical flocking. *New Journal of Physics*, 21(9):093048, 2019.
- [4] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos. A scalable distributed collision avoidance scheme for multi-agent UAV systems. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9212–9218, 2021.
- [5] Y. Lyu, J. Hu, B.M. Chen, C. Zhao, and Q. Pan. Multivehicle flocking with collision avoidance via distributed model predictive control. *IEEE Transactions on Cybernetics*, 51(5):2651–2662, 2021.
- [6] T. Ohtsuka. A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4):563–574, 2004.
- [7] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.
- [8] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.
- [9] C.W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, aug 1987.
- [10] A.S. Sathya, P. Sopasakis, R. Van Parys, A. Themelis, G. Pipeleers, and P. Patrinos. Embedded nonlinear model predictive control for obstacle avoidance using PANOC. In *2018 European Control Conference (ECC)*, pages 1523–1528, 2018.
- [11] J. Shen. Cucker–Smale flocking under hierarchical leadership. *SIAM Journal on Applied Mathematics*, 68(3):694–719, 2008.
- [12] P. Sopasakis, E. Fresk, and P. Patrinos. OpEn: Code generation for embedded nonconvex optimization. *IFAC-PapersOnLine*, 53(2):6548–6554, 2020. 21st IFAC World Congress.
- [13] E. Soria, F. Schiano, and D. Floreano. Distributed predictive drone swarms in cluttered environments. *IEEE Robotics and Automation Letters*, 7(1):73–80, 2022.
- [14] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos. A simple and efficient algorithm for nonlinear model predictive control. In *2017 IEEE 56th Conference on Decision and Control (CDC)*, pages 1939–1944, 2017.
- [15] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, part I: fixed topology. In *2003 IEEE 42nd Conference on Decision and Control (CDC)*, volume 2, pages 2010–2015 Vol.2, 2003.
- [16] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents part II: dynamic topology. In *2003 IEEE 42nd Conference on Decision and Control (CDC)*, volume 2, pages 2016–2021 Vol.2, 2003.
- [17] A. Themelis. *Proximal Algorithms for Structured Nonconvex Optimization*. PhD thesis, KU Leuven, 12 2018.
- [18] T. Yang, Z. Liu, H. Chen, and R. Pei. Formation control and obstacle avoidance for multiple mobile robots. *Acta Automatica Sinica*, 34(5):588–593, 2008.
- [19] R. Yoshise and K. Yamamoto. Algebraic connectivity of layered path graphs under node deletion. *arXiv e-prints*, 2022.
- [20] J. Zhan and X. Li. Flocking of discrete-time multi-agent systems with predictive mechanisms. *IFAC Proceedings Volumes*, 44(1):5669–5674, 2011. 18th IFAC World Congress.
- [21] J. Zhan and X. Li. Flocking of multi-agent systems via model predictive control based on position-only measurements. *IEEE Transactions on Industrial Informatics*, 9(1):377–385, 2013.
- [22] H. 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*, 62(6):1599–1606, 2015.
- [23] H. Zhang, B. Liu, Z. Cheng, and G. Chen. Model predictive flocking control of the Cucker-Smale multi-agent model with input constraints. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(8):1265–1275, 2016.
- [24] L. Zhou and S. Li. Distributed model predictive control for multi-agent flocking via neighbor screening optimization. *International Journal of Robust and Nonlinear Control*, 27(9):1690–1705, 2017.