

Nonlinear Distributed Model Predictive Flocking with Obstacle Avoidance^{*}

Philipp Hastedt^{*} Herbert Werner^{*}

^{*} *Hamburg University of Technology, Institute of Control Systems, 21073 Hamburg, Germany, {philipp.hastedt, h.werner}@tuhh.de*

Abstract: In this paper, we present a framework for nonlinear distributed model predictive flocking with obstacle avoidance, the pursuit of group objectives, and input constraints. While most existing predictive flocking frameworks are only applicable to agents with double-integrator dynamics, we propose a general framework for nonlinear agents that furthermore allows for the independent tuning of cohesive and repulsive inter-agent forces. To reduce the computational complexity, the resulting nonlinear program is solved as a sequential quadratic program with a limited number of iterations. The performance of the proposed algorithms is demonstrated in simulation and compared to a non-predictive flocking algorithm.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Nonlinear predictive control, Distributed nonlinear control, Multi-agent systems, Networked control systems, Constrained control

1. INTRODUCTION

In recent years, the control of networked autonomous multi-agent systems (MASs) has been studied intensively. One particular control strategy for MASs is flocking, a distributed technique for controlling the collective behavior of a swarm of agents. In flocking control, the rules for the agents' behavior are inspired by biological swarms, for example formulated as the three heuristic rules stated by Reynolds (1987): *cohesion*, *separation*, and *alignment*. Here, *cohesion* refers to the agents staying close to neighboring agents, *separation* describes the avoidance of collisions with other swarm members, and *alignment* refers to the incentive to match the velocities of neighboring agents. One of the first frameworks for flocking control was proposed by Olfati-Saber (2006). Based on the rules proposed by Reynolds (1987), the author employs carefully shaped potential functions to model the swarm agents, common group objectives, and obstacles to avoid.

One direction of research focuses on formulating the flocking problem as a model predictive control (MPC) problem. In MPC, optimal control inputs and trajectories are calculated online by solving an optimization problem based on predictions of the future system states. MPC furthermore allows for the explicit handling of constraints on states and inputs in an optimal fashion. For these reasons, MPC-based flocking algorithms are expected to yield a better performance compared to non-predictive flocking algorithms. In fact, this has been demonstrated in various publications (see Zhang et al. (2015), Hastedt and Werner (2023b)). The first centralized predictive flocking algorithm was proposed by Zhan and Li (2011b) and was then extended to a distributed model predictive flocking (DMPF) framework in Zhan and Li (2011a). In Yuan et al. (2017), the simulation results of Zhan and Li (2011a) were

validated in an outdoor experiment. DMPF with obstacle avoidance is presented in Huang et al. (2019). While all of the aforementioned frameworks consider cohesive and repulsive inter-agent forces of equal strength, a framework with asymmetric interaction forces is proposed in Hastedt and Werner (2023b).

In most DMPF frameworks, the flocking problem is formulated as a quadratic optimization problem by approximating or linearizing the nonlinear constraints that arise in flocking. Moreover, most of the existing MPF algorithms are formulated for agents with single- or double-integrator dynamics. This however excludes many types of vehicles that cannot be approximated well by these dynamics, such as fixed-wing aircrafts or wheeled ground robots. A nonlinear distributed model predictive flocking (NLDMPF) framework that can also handle these types of agents was proposed by Rochefort et al. (2012). However, in this work, the capabilities of predictive control are not used to its full extent as the authors consider a discrete decision space with a limited number of admissible control inputs. To further reduce the computational complexity, the authors also constrain the inputs to be constant over the prediction horizon.

In this paper, we present a NLDMPF framework for flocking with obstacle avoidance, the pursuit of group objectives, and input constraints. As proposed by Hastedt and Werner (2023b), the framework in this paper allows for an independent tuning of cohesive and repulsive inter-agent forces. In order to reduce computational complexity, the nonlinear optimization problem is solved as sequential quadratic program (SQP) with a limited number of iterations. The effect of the number of SQP iterations is investigated in simulation for a nonholonomic wheeled robot. Furthermore, the performance of the proposed algorithm is compared to that of the non-predictive flocking scheme proposed by Olfati-Saber (2006) in combination with feedback linearization.

^{*} This work was supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under project 20E1911.

The remainder of this paper is organized as follows: Section 2 provides the preliminaries for the NLDMPF algorithms presented in Section 3. Numerical examples are provided in Section 4. Section 5 concludes this paper.

2. PRELIMINARIES

2.1 Notation

The symbols 0 and I denote zero and identity matrices with compatible dimensions. Zero and one vectors of dimension n are denoted by 0_n and 1_n . The prediction of a variable x at time $k + i$ based on information at time k is denoted as $x(k + i|k)$. Additionally, the abbreviated notations x_{k+i} and $x_{k+i|k}$ are used. The euclidean norm is denoted by $\|\cdot\|$, with $\|x\|_M = \sqrt{x^\top M x}$ denoting the weighted 2-norm with $M \succeq 0$. Furthermore, define $(x_1, x_2) = [x_1^\top \ x_2^\top]^\top$.

2.2 Graph Theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with the elements of the vertex set $\mathcal{V} = \{1, 2, \dots, N\}$ representing the agents and the edge set $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ representing the communication topology. A graph is said to be undirected if $(i, j) \in \mathcal{E} \Rightarrow (j, i) \in \mathcal{E}$. Given a communication range $r_c > 0$ and the vector q of the positions q_i for all agents, $q = (q_1, \dots, q_N)$, the proximity graph is defined as $\mathcal{G}(q) = (\mathcal{V}, \mathcal{E}(q))$, with $\mathcal{E}(q) = \{(i, j) \in (\mathcal{V} \times \mathcal{V}) : \|q_i - q_j\| < r_c, j \neq i\}$. The set of neighboring nodes for a node i is then defined as $\mathcal{N}_i(q) = \{j \in \mathcal{V} : \|q_i - q_j\| < r_c, j \neq i\}$, with $\nu_i = |\mathcal{N}_i|$ denoting the number of neighbors.

2.3 Agent Dynamics

Consider a group of N homogeneous agents with smooth, nonlinear, discrete-time dynamics

$$x_{i,k+1} = f(x_{i,k}, u_{i,k}), \quad i = 1 \dots N, \quad (1)$$

with state $x_i \in \mathbb{R}^n$ and input $u_i \in \mathbb{R}^m$. Let q_i and p_i be the position and velocity of agent i , respectively. Given the prediction horizon H_p , the vectors of the future H_p states and inputs of agent i are given by

$$\begin{aligned} X_i(k+1) &= (x_i(k+1|k), \dots, x_i(k+H_p|k)), \\ U_i(k) &= (u_i(k|k), \dots, u_i(k+H_p-1|k)). \end{aligned}$$

Given the state $x_{i,k|k}$ and the input sequence $U_i(k)$, the future H_p states are calculated as

$$x_{i,k+\ell+1} = f(x_{i,k+\ell}, u_{i,k+\ell}), \quad \ell = 0 \dots H_p - 1. \quad (2)$$

With j_1, \dots, j_{ν_i} denoting the neighbors of agent i , stacked vectors of quantities for all neighbors of an agent are denoted by $\bar{*}_i = (*_{j_1}, \dots, *_{j_{\nu_i}})$. For example, $\bar{x}_{i,k}$ denotes the stacked neighbor states for agent i at time k . The vectors of states and inputs for the neighborhood of agent i stacked over the prediction horizon are then defined as

$$\begin{aligned} \bar{X}_i(k+1) &= (\bar{x}_i(k+1|k), \dots, \bar{x}_i(k+H_p|k)), \\ \bar{U}_i(k) &= (\bar{u}_i(k|k), \dots, \bar{u}_i(k+H_p-1|k)). \end{aligned}$$

2.4 Sequential Quadratic Programming

One particular method for solving nonlinear optimization problems is to solve a sequence of quadratic subproblems.

This method is called sequential quadratic programming (SQP) (Nocedal and Wright, 2006). Consider the nonlinear constrained optimization problem with optimization variable x ,

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, \\ & g(x) \leq 0, \end{aligned}$$

with f , h , and g being smooth functions. Given the current iterate x_k , the minimizer of the quadratic subproblem

$$\begin{aligned} \min_x \quad & \nabla f^\top(x_k)x + x^\top \nabla^2 f(x_k)x \\ \text{s.t.} \quad & \nabla h^\top(x_k)x + h(x_k) = 0, \\ & \nabla g^\top(x_k)x + g(x_k) \leq 0, \end{aligned}$$

is used to define the next iterate x_{k+1} . This procedure is then repeated until convergence.

3. NONLINEAR DISTRIBUTED PREDICTIVE FLOCKING

The core of flocking control is to model the interactions between agents based on the rules of cohesion, separation, and alignment. In the proposed NLDMPF framework, these interactions are modeled via the cost function and suitable constraints of a nonlinear optimization problem. At each time k , agents solve local, nonlinear optimization problems in the optimization variables $Z_{i,k}$. Here, $Z_{i,k}$ contains at least $U_{i,k}$, but depending on the scenario, it can also contain additional variables. The general optimization problem is then stated as follows:

Problem 1.

$$\min_{Z_{i,k}} J_i(x_{i,k}, \hat{X}_{i,k+1}, Z_{i,k}) \quad (3)$$

$$\text{s.t.} \quad h(x_{i,k}, \hat{X}_{i,k+1}, Z_k) \leq 0, \quad (3.a)$$

$$Z_{i,k} \in \mathcal{Z}. \quad (3.b)$$

Here, (3.a) describes the constraints of the optimization problem and (3.b) characterizes the admissible set for the optimization variables. Let $Z_{i,k}^*$ be the optimal solution at time k . The input applied by agent i is then $u_{i,k}^*$, the first input in the optimal input sequence $U_{i,k}^*$.

Note that the cost function J_i is a function of $\hat{X}_{i,k+1}$, an estimation of the neighbors' trajectories over the prediction horizon. As agents are solving their optimization problems in parallel, at time k , agent i is not aware of the decisions of its neighbors computed at the same time. Therefore, each agent estimates its neighbors' predicted trajectories based on the optimal solution at the previous time step. At time k , each agent transmits its current state $x_{i,k|k}$ and the previous optimal solution $U_{i,k-1}^*$ to its neighbors. Agents then shift the received optimal sequences one step in the future, i.e.

$$\begin{aligned} \hat{U}_j(k|k-1) &= (\hat{u}_{j,k}, \dots, \hat{u}_{j,k+H_p-1}) \\ &= (\bar{u}_{j,k|k-1}^*, \dots, \bar{u}_{j,k+H_p-2|k-1}^*, 0_{1 \times m\nu_i}). \end{aligned} \quad (4)$$

With $\hat{x}_{i,k} = \bar{x}_{i,k}$ and (4), each agent then computes an estimate $\hat{X}_{i,k+1}$ of its neighbors' trajectories according to

$$\hat{x}_{j,k+\ell+1} = f(\hat{x}_{j,k+\ell}, \hat{u}_{j,k+\ell}), \quad j \in \mathcal{N}_i, \quad \ell = 1 \dots H_p. \quad (5)$$

The NLDMPF problems are then solved using a SQP approach. To reduce computational complexity, the SQP algorithm is terminated after a specified number of N_{SQP} iterations (Cannon, 2004). Assuming ideal communication with an undirected proximity graph, the NLDMPF algorithm can be stated as follows:

Algorithm 1 NLDMPF algorithm for agent i

Initialization: $k \leftarrow 0$; $U_{i,k-1}^* = 0$

```

1: loop
2:   send  $x_{i,k}$  and  $U_{i,k-1}^*$  to all agents in  $\mathcal{N}_i$ 
3:   receive  $x_{j,k}$  and  $U_{j,k-1}^* \forall j \in \mathcal{N}_i$ 
4:   calculate  $\hat{X}_{i,k+1}$ 
5:    $n_{\text{SQP}} \leftarrow 0$ 
6:   repeat
7:     calculate  $Z_i^{n_{\text{SQP}}}(k)$  by performing one SQP iteration of the NLDMPF problem
8:      $n_{\text{SQP}} \leftarrow n_{\text{SQP}} + 1$ 
9:   until  $n_{\text{SQP}} = N_{\text{SQP}}$ 
10:  set  $U_i^*(k) = U_i^{N_{\text{SQP}}}(k)$ 
11:  apply  $u_k^*$ 
12:   $k \leftarrow k + 1$ 
13: end loop

```

Inspired by Olfati-Saber (2006), in the remainder of this section we address three problems: free flocking (Section 3.1), flocking with group objective (Section 3.2), and flocking with obstacle avoidance (Section 3.3).

3.1 Free Flocking

In free flocking, also called α -flocking, the objective of the swarm is to form a so-called α -lattice. Given a desired inter-agent distance d , an α -lattice is defined analogous to Olfati-Saber (2006).

Definition 1. An α -lattice is a configuration q satisfying $\|q_i - q_j\| = d, \forall (i, j) \in \mathcal{E}(q)$.

To model this objective mathematically, consider the vector of deviations from the desired distance d of an agent i to all of its neighbors given by

$$\delta(x_i, \bar{x}_i) = (\|q_{ij_{i,1}}\| - d, \dots, \|q_{ij_{i,\nu_i}}\| - d). \quad (6)$$

Here, q_{ij} is the displacement vector from agent i to agent j defined as

$$q_{ij} = q_j - q_i. \quad (7)$$

In (6), the formulation of the deviations from the desired distances involves the norm of q_{ij} , which is not continuously differentiable for $q_{ij} = 0$. As SQP requires smooth functions, the 2-norm is replaced by a continuously differentiable map, similar to the σ -norm in Olfati-Saber (2006). Defining the σ -map with the parameter $\epsilon > 0$ as

$$\sigma_\epsilon(z) = \frac{1}{\epsilon} \left(\sqrt{1 + \epsilon^2 \|z\|^2} - 1 \right), \quad (8)$$

one can show that this map is continuously differentiable everywhere and approaches the 2-norm for $\epsilon \rightarrow \infty$. The continuously differentiable vector of deviations from the desired distances is then defined as

$$\delta_\epsilon(x_i, \bar{x}_i) = (\sigma_\epsilon(q_{ij_{i,1}}) - \sigma_\epsilon(d), \dots, \sigma_\epsilon(q_{ij_{i,\nu_i}}) - \sigma_\epsilon(d)). \quad (9)$$

The vector of deviations stacked over the prediction horizon is given by

$$\Delta(X_{i,k+1}, \bar{X}_{i,k+1}) = (\delta_\epsilon(x_{i,k+1}, \bar{x}_{i,k+1}), \dots, \delta_\epsilon(x_{i,k+H_p}, \bar{x}_{i,k+H_p})). \quad (10)$$

As proposed in Hastedt and Werner (2023b), the desired behavior can then be formulated in the form of softened inequality constraints

$$-\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^-, \quad (11)$$

$$\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^+, \quad (12)$$

with $\sigma_{i,k+1}^-, \sigma_{i,k+1}^+ \in \mathbb{R}_{\geq 0}^{H_p \nu_i}$ being the stacked slack variables over the prediction horizon. By considering $\sigma_{i,k+1}^-$ and $\sigma_{i,k+1}^+$ as additional optimization variables in the NLDMPF problem, this formulation allows for the independent tuning of cohesive and repulsive inter-agent forces. Defining the vector of optimization variables as

$$Z_{i,k} = (U_{i,k}, X_{i,k+1}, \sigma_{i,k+1}^-, \sigma_{i,k+1}^+), \quad (13)$$

the NLDMPF problem for free flocking can be stated as

Problem 2. (free flocking)

$$\min_{Z_{i,k}} J_{i,\alpha}(x_{i,k}, \hat{X}_{i,k+1}, Z_{i,k}) \quad (14)$$

$$\text{s.t. } x_{i,k+\ell+1} = f(x_{i,k+\ell}, u_{i,k+\ell}), \ell = 0 \dots H_p - 1, \quad (14.a)$$

$$-\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^-, \quad (14.b)$$

$$\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^+, \quad (14.c)$$

$$\sigma_{i,k+1}^-, \sigma_{i,k+1}^+ \in \mathbb{R}_{\geq 0}^{H_p \nu_i}, \quad (14.d)$$

$$U_{i,k} \in \mathcal{U}, \quad (14.e)$$

with

$$J_{i,\alpha}(x_{i,k}, \hat{X}_{i,k+1}, Z_{i,k}) = \|U_{i,k}\|_R^2 + \|\sigma_{i,k+1}^-\|_{Q_\alpha^-}^2 + \|\sigma_{i,k+1}^+\|_{Q_\alpha^+}^2 + \sum_{j \in \mathcal{N}_i} \|X_{i,k+1} - \hat{X}_{j,k+1}\|_{Q_x}^2. \quad (15)$$

Here, $R, Q_\alpha^+, Q_\alpha^- \succ 0$, and $Q_x \succeq 0$ are the weighting matrices. The set of admissible inputs \mathcal{U} is defined as

$$\mathcal{U} := \{u_i : -|u_{\max}| \leq u_i \leq |u_{\max}|\}, \quad (16)$$

with maximum admissible input $u_{\max} \in \mathbb{R}^m$. In terms of the rules of flocking, the second and third term in (15), together with (14.b)-(14.d), describe the separation and cohesion that drive the agents towards the desired inter-agent distance. Alignment is modeled by the fourth term in (15). By a suitable choice of Q_x , the mismatch of states to be aligned, for example velocities, is penalized.

3.2 Flocking with Group Objective

Next, the NLDMPF framework is extended to the pursuit of group objectives in the form of virtual γ -agents. While the framework presented in this paper considers γ -agents in the form of static reference points, the proposed scheme can easily be adapted to dynamic references. Let x_i^r be the reference state for an agent i , containing the reference position q_i^r . Let $q_{ir,k} = q_i^r - q_{i,k}$ be the vector from agent i to the reference point. To limit the navigational penalty, especially when agents are far away from the reference point, each agent generates a virtual position reference

$$q_{i,k}^r = \begin{cases} q_i^r, & \text{if } \|q_{ir,k}\| \leq d_\ell, \\ q_{i,k} + d_\ell q_{ir,k} / \|q_{ir,k}\| & \text{otherwise.} \end{cases} \quad (17)$$

Here, d_ℓ is the look ahead distance on the line from the agent's current position to the reference point. Assuming

that the reference point is constant over the prediction horizon, the reference vector over the prediction horizon is given by

$$X_{i,k}^r = 1_{H_p} \otimes x_{i,k}^r. \quad (18)$$

Again using (13) as the vector of optimization variables, the NLDMPF problem for flocking with group objective can be formulated as

Problem 3. (flocking with group objective)

$$\begin{aligned} \min_{Z_{i,k}} J_{i,\alpha}(x_{i,k}, \hat{X}_{i,k+1}, Z_{i,k}) + J_{i,\gamma}(x_{i,k}, Z_{i,k}) \quad (19) \\ \text{s.t. (14.a) – (14.e),} \end{aligned}$$

with

$$J_{i,\gamma}(x_{i,k}, Z_{i,k}) = \|X_{i,k+1} - X_{i,k}^r\|_{Q_\gamma}^2. \quad (20)$$

Here, $Q_\gamma \succeq 0$ is the reference tracking weighing matrix.

3.3 Flocking with Obstacle Avoidance

This paper considers static, spherical obstacles, characterized by the obstacles' center positions q_o and radii r_o . Let $\mathcal{V}_\beta = \{o_1, o_2, \dots, o_{N_\beta}\}$ denote the set of N_β obstacles. As proposed in Olfati-Saber (2006), obstacles within a sensing radius r_β of an agent generate a so-called β -agent on the boundary of the obstacle at the point closest to the agent. Let x_i^o denote the state of the β -agent for obstacle o , generated by agent i . The vector from agent i to this β -agent is defined as $q_{io} = q_i^o - q_i$. The set of obstacles detected by agent i is given by

$$\mathcal{N}_i^\beta = \{o \in \mathcal{V}_\beta : \|q_{io}\| \leq r_\beta\}, \quad (21)$$

with $\omega_i = |\mathcal{N}_i^\beta|$. The stacked state vector of the obstacles $o_{i,1}, o_{i,2}, \dots, o_{i,|\mathcal{N}_i^\beta|}$ detected by agent i is

$$\bar{x}_{i,k}^o = (x_{i,k}^{o_1}, \dots, x_{i,k}^{o_{\omega_i}}). \quad (22)$$

Assuming that β -agents are constant over the prediction horizon, the obstacle state over the prediction horizon is given by

$$\bar{X}_{i,k}^o = 1_{H_p} \otimes \bar{x}_{i,k}^o. \quad (23)$$

Let d_β be the desired separation from the obstacles. The continuously differentiable deviations from the desired obstacle separation are then defined analogous to (9) as

$$\delta_\epsilon^\beta(x_i, \bar{x}_i^o) = (\sigma_\epsilon(q_{io_1}) - \sigma_\epsilon(d_\beta), \dots, \sigma_\epsilon(q_{io_{\omega_i}}) - \sigma_\epsilon(d_\beta)). \quad (24)$$

The vector of deviations stacked over the prediction horizon is then given by

$$\Delta^\beta(X_{i,k+1}, \bar{X}_{i,k}^o) = (\delta_\epsilon^\beta(x_{i,k+1}, \bar{x}_{i,k}^o), \dots, \delta_\epsilon^\beta(x_{i,k+H_p}, \bar{x}_{i,k}^o)). \quad (25)$$

To integrate obstacle avoidance into the NLDMPF problem, similar to (11) and (12), a softened inequality constraint

$$-\Delta^\beta(X_{i,k+1}, \bar{X}_{i,k}^o) \leq \sigma_{i,k+1}^\beta \quad (26)$$

is formulated. Here, $\sigma_{i,k+1}^\beta \in \mathbb{R}_{\geq 0}^{H_p \omega_i}$ is the vector of corresponding slack variables. Considering $\sigma_{i,k+1}^\beta$ as optimization variable in the NLDMPF problem and defining the vector of optimization variables as

$$Z_{i,k} = (U_{i,k}, X_{i,k+1}, \sigma_{i,k+1}^-, \sigma_{i,k+1}^+, \sigma_{i,k+1}^\beta), \quad (27)$$

the NLDMPF problem with obstacle avoidance can be stated as

Problem 4. (flocking with obstacle avoidance)

$$\begin{aligned} \min_{Z_{i,k}} J_{i,\alpha}(x_{i,k}, \hat{X}_{i,k+1}, Z_{i,k}) + J_{i,\gamma}(x_{i,k}, Z_{i,k}) \\ + J_{i,\beta}(x_{i,k}, Z_{i,k}) \quad (28) \end{aligned}$$

$$\text{s.t. } x_{i,k+l+1} = f(x_{i,k+l}, u_{i,k+l}), \ell = 0 \dots H_p - 1, \quad (28.a)$$

$$-\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^-, \quad (28.b)$$

$$\Delta(X_{i,k+1}, \hat{X}_{i,k+1}) \leq \sigma_{i,k+1}^+, \quad (28.c)$$

$$-\Delta(X_{i,k+1}, \bar{X}_{i,k}^o) \leq \sigma_{i,k+1}^\beta, \quad (28.d)$$

$$\sigma_{i,k+1}^-, \sigma_{i,k+1}^+ \in \mathbb{R}_{\geq 0}^{H_p \nu_i}, \quad (28.e)$$

$$\sigma_{i,k+1}^\beta \in \mathbb{R}_{\geq 0}^{H_p \omega_i}, \quad (28.f)$$

$$U_{i,k} \in \mathcal{U}, \quad (28.g)$$

with

$$J_{i,\beta}(x_{i,k}, Z_{i,k}) = \|\sigma_{i,k+1}^\beta\|_{Q_\beta}^2. \quad (29)$$

Here, $Q_\beta \succ 0$ is the obstacle avoidance weighing matrix.

Remark 1. The cost functions of all three presented optimization problems (Problems 2, 3 and 4) are quadratic in the optimization variables. Hence, the generation of the quadratic subproblems in the SQP algorithm only requires a linearization of the constraints.

4. NUMERICAL EXAMPLE

In this section, the performance of the proposed NLDMPF algorithms is investigated in simulation. Two scenarios are presented. In the first one, the effect of the number of SQP iterations is investigated in a free flocking scenario. In the second scenario, the performance of the proposed algorithm is demonstrated in a scenario with multiple obstacles and compared to the performance of a non-predictive flocking algorithm. The simulation results are generated with the open-source Matlab MAS simulation library presented in Hespe et al. (2023). The code for reproducing the simulations is provided in Hastedt and Werner (2023a).

Consider the nonholonomic wheeled robot depicted in Fig. 1. The nonlinear, discrete-time equations of motion with sampling time T can be written as

$$x_{k+1} = \begin{bmatrix} q_{k+1}^x \\ q_{k+1}^y \\ \psi_{k+1} \\ v_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} q_k^x \\ q_k^y \\ \psi_k \\ v_k \\ \omega_k \end{bmatrix} + T \begin{bmatrix} \cos(\psi_k) v_k \\ \sin(\psi_k) v_k \\ \psi_k \\ F_k/M \\ \tau_k/I_z \end{bmatrix}. \quad (30)$$

Here, M is the mass and I_z is the inertia about the z -axis. The center of mass (COM) is denoted by $q = (q^x, q^y)$, ψ

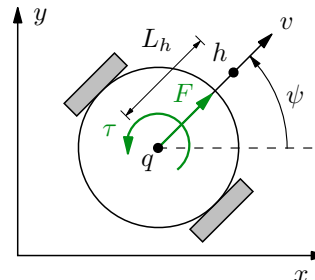


Fig. 1. Nonholonomic wheeled robot.

is the orientation, v is the velocity in heading direction, and ω is the angular velocity. The control inputs are the applied force F and the applied torque τ . Furthermore, h denotes a handle point located at a distance L_h from q in the heading direction.

4.1 Free Flocking

Consider a group of $N = 10$ agents with initial positions $q_{i,0} \in [0, 25] \times [0, 25]$, initial orientations $\psi_{i,0} \in [0, 2\pi]$, and initial velocities $v_{i,0} \in [-1, 1]$ and $\omega_{i,0} = 0$. The objective of the swarm is to form a static α -lattice without γ - or β -agents. The sampling time is chosen as $T = 0.3$ s. The model parameters are $M = 1$, $I_z = 0.25$, $F_{\max} = 1$, and $\tau_{\max} = 1$. The simulation parameters are $\epsilon = 10$, $H_p = 5$, $r_c = 8.4$, $d = 7$, with tuning matrices $R = I_{H_p} \otimes \text{diag}(0.2, 0.1)$, $Q_{\alpha}^- = I$, $Q_{\alpha}^+ = I$, and $Q_x = I_{H_p} \otimes \text{diag}(0, 0, 0, 0.002, 0.002)$. As performance measure, the α -lattice irregularity

$$J_q = \sum_{(i,j) \in \mathcal{E}(q)} \frac{|||q_{ij}|| - d|^2}{|\mathcal{E}(q)|} \quad (31)$$

is used, where $|\mathcal{E}(q)|$ denotes the number of edges. In Fig. 2, the α -lattice irregularity is displayed for different maximum numbers of SQP-iterations.

When comparing the speeds of convergence, $N_{\text{SQP}} = 3$, $N_{\text{SQP}} = 10$, and $N_{\text{SQP}} = 100$ initially show very similar performances, converging faster compared to only one iteration. However, with $N_{\text{SQP}} = 1$ and $N_{\text{SQP}} = 3$, the α -lattice irregularity does not converge to zero, but $J_{q,\text{final}} = 0.24$, and $J_{q,\text{final}} = 0.18$, respectively. For $N_{\text{SQP}} = 10$ and $N_{\text{SQP}} = 100$, there is only an insignificant difference in the performances with both converging to a zero α -lattice irregularity. Hence, for the investigated wheeled robot, $N_{\text{SQP}} = 10$ appears to be a good trade-off between performance and computational complexity.

4.2 Obstacle Avoidance

In this scenario, the performance of the proposed NLDMPF scheme is compared to that of a non-predictive flocking algorithm. For this purpose, a feedback-linearized model of the nonholonomic wheeled robot is used in combination with the flocking algorithm proposed in Olfati-Saber (2006). Since the model in (30) is not controllable for $v = 0$, instead of directly controlling the position of the COM, the position of the handle point h is controlled with

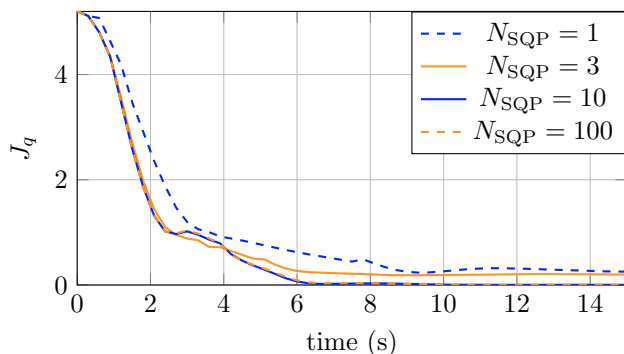


Fig. 2. Comparison of α -lattice irregularities for different numbers of SQP iterations.

$L_h = 0.5$ (see Fig. 1). The equations of motion for the handle point and the feedback linearization laws can be found in Lawton et al. (2003). Here, we consider a group of $N = 20$ agents tasked with reaching a target position $q^r = (85, 85)$ with zero velocities $v^r = 0$, $\omega^r = 0$, avoiding a cluster of spherical obstacles on the way. The obstacles are described by the center positions q_o and radii r_o

$$q_o = \begin{bmatrix} 35 & 60 & 30 & 55 & 75 & 15 \\ 35 & 35 & 60 & 55 & 45 & 45 \end{bmatrix}, \quad r_o = [6 \ 3 \ 2 \ 1 \ 2 \ 1].$$

The agents are initialized with initial positions $q_{i,0} \in [-15, 15] \times [-15, 15]$, initial orientations $\psi_{i,0} \in [0, 2\pi]$, and initial velocities $v_{i,0} \in [-1, 1]$ and $\omega_{i,0} = 0$. Model and tuning parameters are chosen as in Section 4.1, except $Q_{\alpha}^+ = 0.1I$. As a result of the investigation in the previous section, the number of SQP iterations is selected as $N_{\text{SQP}} = 10$. Additional parameters are $r_{\beta} = 8.4$, $d_{\beta} = 6$, $Q_{\beta} = 1.2I$, $Q_{\gamma} = I_{H_p} \otimes \text{diag}(0.1, 0.1, 0, 0.002, 0.001)$, and $d_{\ell} = 2$. To get comparable results, both algorithms are tuned to use similar amounts of input in terms of the root-mean-square (RMS) value. Furthermore, since the algorithm in Olfati-Saber (2006) cannot handle input constraints explicitly, saturation is added after feedback linearization.

To evaluate and compare the performances, we consider the minimum inter-agent and agent-to-obstacle distances

$$d_a = \min_{i,j \in \mathcal{V}, i \neq j} \|q_{ij}\|, \quad d_o = \min_{i \in \mathcal{V}, o \in \mathcal{V}_{\beta}} \|q_{io}\|,$$

depicted in Fig. 3. For both algorithms, these distances are computed from the agents' centers of mass. Furthermore t_o , the time it takes the swarm to clear the field obstacles, is considered as the time interval in which the obstacle is within the interaction range of at least one agent. The agents' trajectories for the proposed NLDMPF algorithm as well as the non-predictive flocking algorithm with feedback linearization are depicted in Fig. 4. The simulation performances are summarized in Table 1.

In both cases, the agents are able to clear the field of obstacles and form an α -lattice at the reference point within 200 s. In comparison to the non-predictive algorithm, with the proposed scheme, the swarm passes the field of obstacles 33 s (36%) faster. With both algorithms however, d_a does not converge to the desired distance, but a smaller distance (see Fig. 3). This is caused by the conflicting objectives of inter-agent separation while at the same time minimizing the distance to the γ -agent. For the non-predictive algorithm, this offset is even larger as it

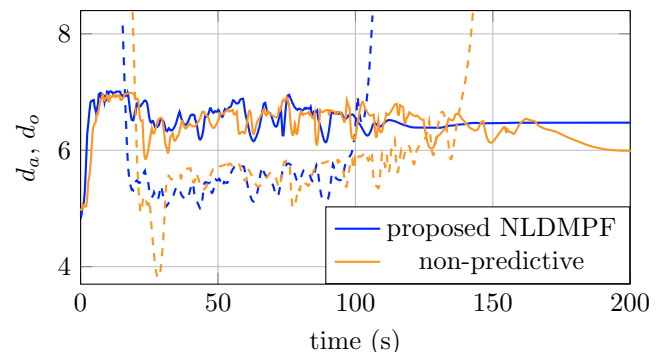


Fig. 3. Distances d_a (solid lines) and d_o (dashed lines) for obstacle avoidance scenario.

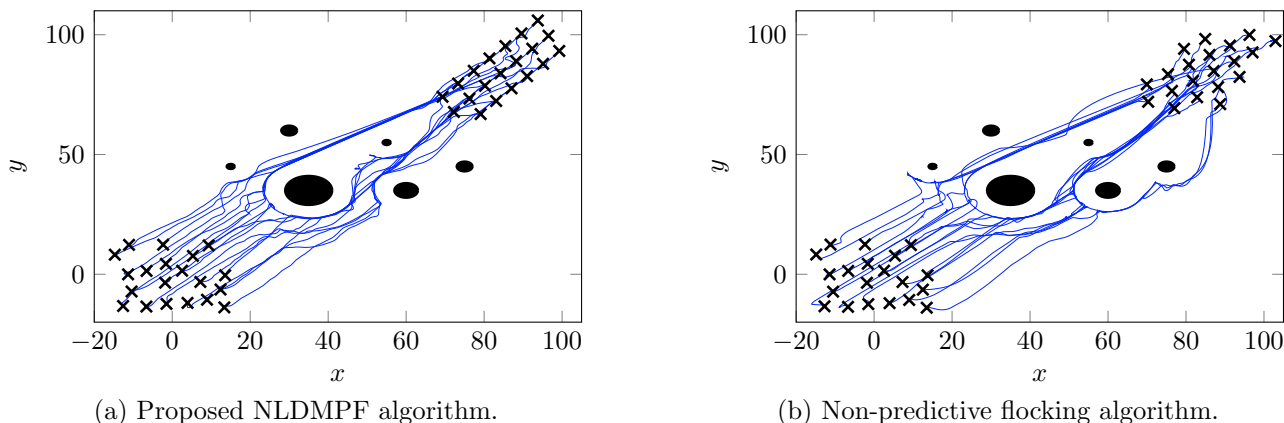


Fig. 4. Agent trajectories for predictive and non-predictive algorithms with markers for initial and final positions.

considers the handle point position h instead of the COM. In terms of the minimum inter-agent distances, both algorithms show a similar performance with a slightly larger minimum d_a for the predictive scheme. However, when comparing the minimum distances to the obstacles, with the non-predictive algorithm, at least one agent gets significantly closer to the obstacles compared to the NLDMPF approach. In the case of the non-predictive algorithm, the separation performance is again affected by considering the handle point instead of the COM. In conclusion, the advantages of predictive control in combination with the possibility of directly controlling the position of the COM, result in a superior performance when using the proposed NLDMPF algorithm.

Table 1. Performance comparison summary

	$d_{a,final}$	$\min(d_a)$	$\min(d_o)$	t_o
proposed	6.48	6.14	5	90.9 s
non-predictive	6.01	5.85	3.81	123.9 s

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework for nonlinear distributed model predictive flocking with obstacle avoidance and reference tracking. The resulting nonlinear optimization problems are solved with a predefined number of SQP iterations. Using the example of a nonlinear, nonholonomic wheeled robot, we demonstrated in simulation that a small number of SQP iterations is sufficient to achieve the desired performance. Furthermore, in comparison to a non-predictive flocking algorithm, the proposed scheme achieves a superior performance, especially in terms of the obstacle clearance time and the separation from obstacles. As the performance of the proposed algorithms has so far only been investigated heuristically, future research involves an investigation of the stability and convergence properties of the proposed scheme.

REFERENCES

- Cannon, M. (2004). Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28(2), 229–237.
- Hastedt, P. and Werner, H. (2023a). Code for Paper: Nonlinear Distributed Model Predictive Flocking with Obstacle Avoidance. URL <https://doi.org/10.5281/zenodo.7801765>.
- Hastedt, P. and Werner, H. (2023b). Distributed Model Predictive Flocking with Obstacle Avoidance and Asymmetric Interaction Forces. In *2023 American Control Conference (ACC)*.
- Hespe, C., Datar, A., Schneider, D., Saadabadi, H., Werner, H., and Frey, H. (2023). Distributed H_2 Controller Synthesis for Multi-Agent Systems with Stochastic Packet Loss. In *2023 American Control Conference (ACC)*.
- Huang, D., Yuan, Q., and Li, X. (2019). Decentralized flocking of multi-agent system based on MPC with obstacle/collision avoidance. In *Proceedings of the 38th Chinese Control Conference*, 5587–5592. IEEE.
- Lawton, J., Beard, R.W., and Young, B.J. (2003). A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6), 933–941.
- Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. SpringerLink. Springer New York, New York, NY, second edition edition.
- Olfati-Saber, R. (2006). Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Transactions on Automatic Control*, 51(3), 401–420.
- Reynolds, C.W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Comput. Graph.*, 21(4), 25–34.
- Rocheffort, Y., Bertrand, S., Piet-Lahanier, H., Beauvois, D., and Dumur, D. (2012). Cooperative Nonlinear Model Predictive Control for Flocks of Vehicles. *IFAC Proceedings Volumes*, 45(1), 169–174.
- Yuan, Q., Zhan, J., and Li, X. (2017). Outdoor flocking of quadcopter drones with decentralized model predictive control. *ISA transactions*, 71(Pt 1), 84–92.
- Zhan, J. and Li, X. (2011a). Decentralized flocking protocol of multi-agent systems with predictive mechanisms. In *Proceedings of the 30th Chinese Control Conference*, 5995–6000.
- Zhan, J. and Li, X. (2011b). Flocking of Discrete-time Multi-Agent Systems with Predictive Mechanisms. *IFAC Proceedings Volumes*, 44(1), 5669–5674.
- Zhang, H.T., Cheng, Z., Chen, G., and Li, C. (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), 1599–1606.