

# A Decentralized Event-based Predictive Navigation Scheme for Air-Traffic Control

Spyros Maniatopoulos, Dimos V. Dimarogonas and Kostas J. Kyriakopoulos

**Abstract**—This paper presents an approach for the multi-agent navigation and conflict resolution problem, that considers the issue of performance. We present a decentralized predictive navigation scheme that combines the Decentralized Navigation Functions methodology with the Model Predictive Control (MPC) framework while preserving the former’s collision avoidance and convergence guarantees. Aircrafts flying at constant altitude are modeled as unicycles. Performance criteria are encoded in a cost functional. Due to decentralization, each agent does not take into account the decisions of others in the control law calculation, resulting in performance discrepancies. Therefore we employ event-triggered executions in our scheme. The improved performance is demonstrated through simulations.

## I. INTRODUCTION

The problem of multi-agent navigation has attracted a lot of attention during the last years. Major applications include multiple robotic vehicles operating in a common workspace and Air Traffic Control (ATC). The problem’s main concern is convergence of each agent to a target destination along collision-free trajectories. Our motivation for this work is the performance aspect of the problem. By performance we refer to task related criteria, e.g. minimum control effort or deviation from a nominal path, not computation time.

A class of methods used to solve path planning problems is artificial potential fields [1]. In particular, Decentralized Navigation Functions (DNFs) [2] are a multi-agent extension of the Navigation Functions (NFs) methodology [3]. Other approaches to the multi-agent navigation or coordination problem employ MPC (see for example [4]–[8]).

Most relevant to this paper are Refs. [5] and [9]. In [5], a distributed approach where MPC is used to generate intermediate way-points for DNFs is proposed. At each iteration, each agent solves an optimization problem and broadcasts its solution to others. In [9], the authors present a framework for the navigation of a single robot that combines receding horizon control and control Lyapunov functions. Control inputs are parametrized using a perturbation on the direction of a potential field’s gradient. The Finite Horizon Optimal Control Problem (FHOCP) is addressed using randomized algorithms [10], and Lyapunov-like stability conditions are given in [11]. Finally, event-triggered (E-T) nonlinear MPC has appeared in [12], [13].

Spyros Maniatopoulos and Kostas J. Kyriakopoulos are with the Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, 9 Heron Polytechniou Street, Zografou 15780, Greece [smaniato@ieee.org](mailto:smaniato@ieee.org), [kkyria@mail.ntua.gr](mailto:kkyria@mail.ntua.gr)

Dimos V. Dimarogonas is with the ACCESS Linnaeus Centre, School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden [dimos@kth.se](mailto:dimos@kth.se). He is also affiliated with the KTH Centre for Autonomous Systems and is supported by the Swedish Research Council (VR) through contract 2009-3948.

This paper extends ideas from [9] to a multi-agent setting by employing dipolar DNFs [2]. The aircrafts (agents) are abstracted to unicycles with lower bounded linear velocities. We consider planar flight since heading changes are preferred over altitude changes in ATC. We account for performance requirements by introducing a deviation from the direction of the DNF’s gradient. This deviation is calculated by solving a FHOCP using the randomized algorithms in [10]. The overall scheme is decentralized [14], i.e., agents’ predictive controllers do not communicate during navigation. The only additional information required, compared to the original scheme using just DNFs, is the broadcast of each agent’s target configuration to other agents at the beginning of the manoeuvre. This information exchange is minimal compared to the broadcast of state trajectories. Decentralization introduces uncertainty wrt each agent. This motivates the use of event-triggered recalculations of the FHOCP. An execution rule, based on the discrepancy between the predicted and actual performance, generates the events. Each agent derives its own event times independently, in a decentralized and asynchronous manner.

The rest of this paper is organized as follows: an introduction to DNFs and MPC is given in §II, followed by §III where the multi-agent predictive navigation scheme is presented. Decentralization and event-triggering are treated in §IV. In §V, the navigation properties of the proposed scheme are analyzed. Comparative simulation results are given in §VI and §VII summarizes our conclusions.

## II. PRELIMINARIES

### A. Agents’ Model

The case of  $N$  aircrafts vehicles (agents) navigating in a two dimensional workspace  $W \subset \mathbb{R}^2$  is considered. Each agent will be described by the unicycle kinematic model:

$$\dot{\mathbf{q}}_i = \begin{bmatrix} \dot{\mathbf{p}}_i \\ \dot{\phi}_i \end{bmatrix} = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\phi}_i \end{bmatrix} = \begin{bmatrix} v_i \cos \phi_i \\ v_i \sin \phi_i \\ \omega_i \end{bmatrix}, \quad (1)$$

where  $i \in \mathcal{N} = \{1, \dots, N\}$  and  $\mathbf{I}_i \triangleq [\cos \phi_i \ \sin \phi_i]^\top$ . Furthermore,  $\mathbf{p}_i = [x_i \ y_i]^\top$  is the position vector of agent  $i$  wrt an earth fixed frame and  $\phi_i \in (-\pi, \pi]$  its heading angle, i.e., the angle between agent  $i$ ’s longitudinal axis and the earth-fixed  $X$ -axis. The configuration of each agent is then  $\mathbf{q}_i \in Q = W \times (-\pi, \pi]$ . The control vector  $\mathbf{u}_i$  consists of the linear and the angular velocities  $\mathbf{u}_i = [v_i \ \omega_i]^\top$ .

Each aircraft’s protected zone is a disk of radius  $r_i$  centered at  $\mathbf{p}_i$ . The agents’ workspace is assumed to be a disk of radius  $R_w$ , centered at the origin of the earth-fixed frame.

Its boundary  $\partial W$  is considered an obstacle. In the following, the term *conflict* will refer to the intersection of two or more agents' protected zones. Finally, each agent's linear velocity is lower bounded by a nominal speed  $U_i$ , given as follows:

$$U_i = \begin{cases} U_{id}, & \|\mathbf{p}_i - \mathbf{p}_{id}\| > r_0 \\ \frac{\|\mathbf{p}_i - \mathbf{p}_{id}\|}{r_0} \cdot U_{id}, & \|\mathbf{p}_i - \mathbf{p}_{id}\| \leq r_0 \end{cases} \quad (2)$$

where  $U_{id}$  is a desired absolute cruise speed for a given aircraft and altitude (constant or independent of this scheme).

### B. Decentralized Navigation Functions

Navigation Functions [3] are a special case of artificial potential fields [1]. A NF  $\Phi$  is a mapping with  $\Phi = 0 \Leftrightarrow \mathbf{p} = \mathbf{p}_d$  (unique minimum at target destination  $\mathbf{p}_d$ ), and is uniformly maximal ( $\Phi = 1$ ) on the boundary of  $W$ . NFs are not suitable for the control of nonholonomic vehicles, as they do not account for the constraints that apply on such vehicles. In [15], dipolar NFs were proposed, that have the following advantageous property: the flow curves associated with the resulting vector field are all tangent to the desired orientation at the destination, thus eliminating in-place rotation.

In a multi-agent setting, a dipolar Decentralized Navigation Function (DNF) is defined for each agent  $i \in \mathcal{N}$ :

$$\Phi_i = \frac{\gamma_{di} + f_i}{((\gamma_{di} + f_i)^k + H_{nhi} \cdot G_i \cdot \beta_{0i})^{1/k}} \quad (3)$$

Details regarding the construction of (3) can be found in [16]. Briefly, the function  $G_i$  measures the  $i$ -th agent's proximity to conflicts, i.e.,  $G_i$  is zero when minimum separation is lost, whereas  $G_i > 0$  away from conflicts. The function  $\gamma_{di} = \|\mathbf{p}_i - \mathbf{p}_{id}\|^2$  is a measure of the distance of agent  $i$  from its target destination. The function  $f_i = f_i(G_i)$  is activated temporarily to facilitate the navigation of neighboring agents. The term  $\beta_{0i}$  refers to the workspace bounding obstacle while  $H_{nhi}$  corresponds to the artificial obstacle used to render the potential field dipolar. Finally,  $k > 0$  is a tuning parameter.

Navigation for nonholonomic agents [2] is based on the projection of the DNF's gradient  $\nabla_i \Phi_i = [\Phi_{ix} \ \Phi_{iy}]^\top$  on the agent's heading direction:  $P_i = [\cos \phi_i \ \sin \phi_i] \cdot \nabla_i \Phi_i$ . The sign of  $P_i$ ,  $s_i = \text{sgn}(P_i)$ , determines the direction of motion.

Collision avoidance and convergence depend on the decrease of potential  $\Phi_i$  whose time derivative has two terms:  $\dot{\Phi}_i = \frac{\partial \Phi_i}{\partial t} + \nabla_i \Phi_i \cdot \dot{\mathbf{p}}_i = \sum_{j \neq i} \nabla_j \Phi_i^\top \cdot \mathbf{I}_j v_j + P_i v_i$ . The second term is due to the motion of agent  $i$ , while the first due to the motion of the other agents  $j \in \mathcal{N}$ ,  $j \neq i$ .

A multi-agent navigation problem can be stated as follows:

*Problem 1:* Design a control law  $\forall i \in \mathcal{N}$ , described by (1), that will steer each agent to its destination  $\mathbf{p}_{id}$  with the desired orientation  $\phi_{id}$ , while avoiding conflicts and  $\partial W$ .

A DNF-based control law (similar to the one in [17]), that solves Problem 1 and respects the lower bound on  $v_i$ , Eq. (2), is given below. The linear and angular velocities,  $v_i$  and  $\omega_i$  respectively, are given by:

$$v_i = \begin{cases} -s_i \cdot U_i, & \frac{\partial \Phi_i}{\partial t} \leq U_i(|P_i| - \varepsilon) \\ -s_i \cdot \frac{\frac{\partial \Phi_i}{\partial t} + \varepsilon U_i}{|P_i|}, & \frac{\partial \Phi_i}{\partial t} > U_i(|P_i| - \varepsilon) \end{cases} \quad (4a)$$

$$\omega_i = -k_\phi(\phi_i - \phi_{nhi}) + \dot{\phi}_{nhi}, \quad (4b)$$

where  $U_i$  is given by (2). In Eq. (4a), the nominal speed is applied as long as it can guarantee the decrease of  $\Phi_i$ , and the transition is continuous by construction. The angle  $\phi_{nhi} \triangleq \text{atan2}(\text{sgn}(d_i)\Phi_{iy}, \text{sgn}(d_i)\Phi_{ix})$  represents the heading of  $\text{sgn}(d_i)\nabla_i \Phi_i$ ,  $d_i = [\cos \phi_{id} \ \sin \phi_{id}] \cdot (\mathbf{p}_i - \mathbf{p}_{id})$  is the projection of the current position vector wrt to the destination on the direction of the desired orientation, and  $\text{atan2}(y, x) \triangleq \arg(x, y)$ ,  $(x, y) \in \mathbb{C}$ . Therefore  $\text{sgn}(d_i)$  is equal to 1 in front and  $-1$  behind the target configuration. Finally,  $k_\phi$  is a positive gain. The control law (4b) tracks the direction of the dipolar DNF's gradient. Note that each agent is assumed to measure the position, orientation and linear velocity of all other agents, but has no knowledge of their destinations.

### C. Model Predictive Control

Model Predictive Control (MPC) is based on iterative, finite horizon optimization. At a calculation time  $t$ , the current system state  $\mathbf{x}(t)$  is sampled and a cost minimizing control law is computed for a relatively short time horizon  $[t, t+T)$ . That is, one aims at minimizing a cost functional

$$J(t, \mathbf{x}, \mathbf{u}, T) = \int_t^{t+T} \Lambda(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau + M(\mathbf{x}(t+T)),$$

which consists of an incremental cost (also called running cost) and a terminal cost. The functional  $J(\cdot)$  quantifies the cost of flowing along a system trajectory  $\mathbf{x}[t, t+T)$ , with  $\mathbf{x}(t)$  the initial condition, under the control law  $\mathbf{u}[t, t+T)$ . In the above,  $t$  denotes (current) time,  $T$  is the (fixed) prediction horizon and  $\Lambda(\mathbf{x}, \mathbf{u})$  is a positive definite function of  $\mathbf{x}$  and  $\mathbf{u}$  (running cost function). The function  $M(\cdot)$  is an approximation of the cost-to-go from  $t+T \rightarrow \infty$ . The FHOCP is to determine  $\mathbf{u}^*[t, t+T) \triangleq \arg \min_{\mathbf{u}} J(t, \mathbf{x}, \mathbf{u}, T)$ . This control law is then applied over the interval  $[t, t+T_c)$ , the control phase, where  $T_c \leq T$  is the (typically fixed) control horizon and the FHOCP is solved again for  $\mathbf{x}(t+T_c)$ .

The notation  $\mathbf{u}[t, t+T)$  will refer to the sequence of control inputs over the time interval  $[t, t+T)$ , whereas the value of said input at a specific time instant  $t_s \in [t, t+T)$  will be denoted as either  $\mathbf{u}_{[t, t+T)}(t_s)$  or simply  $\mathbf{u}(t_s)$ .

## III. PROPOSED SCHEME

### A. Multi-agent Predictive Navigation

In a non-centralized multi-agent navigation setting, the FHOCP, for each agent  $i \in \mathcal{N}$ , is stated as follows:

$$J_i(t, \mathbf{q}_i, \mathbf{u}_i, T) = \int_t^{t+T} \Lambda_i(\mathbf{q}_i(\tau), \mathbf{u}_i(\tau)) d\tau + \Phi_i(t+T), \quad (5a)$$

$$J_i^*(t, \mathbf{q}_i, T) = \min_{\mathbf{u}_i[t, t+T)} J_i(t, \mathbf{q}_i, \mathbf{u}_i, T) \quad (5b)$$

subject to the system kinematics (1) and conflict avoidance constraints. The positive definite function  $\Lambda(\mathbf{q}, \mathbf{u})$  encodes the desired performance criteria and  $\Phi_i$  is a dipolar DNF (3). Navigation Functions will be used as terminal costs since they can approximate the cost-to-go from  $(t+T) \rightarrow \infty$ . We state a multi-agent navigation problem, that takes into account performance issues, as follows:

*Problem 2:* For each agent  $i \in \mathcal{N}$ , given a DNF (3), a running cost function  $\Lambda_i(\mathbf{q}_i, \mathbf{u}_i)$  and a prediction horizon

$T$ , derive, for each prediction interval  $[t, t+T)$ , the control strategies  $\mathbf{u}_i^*[t, t+T)$  that solve (5), in such a way that their concatenation over  $t \in [0, \infty)$  is also a solution to Problem 1.

The trajectories of agents tracking the negated gradient of a NF only depend on initial-final conditions and the NF's parameters (set a priori). Thus, we introduce a deviation  $\theta_i$  from the direction of the DNF's gradient  $\phi_{nhi}$ , which is translated into a control input as follows:

$$\bar{\mathbf{u}}_i(\tau) \triangleq [0 \ \bar{\omega}_i]^\top = [0 \ k_\phi \theta_i + \dot{\theta}_i]^\top, \quad (6)$$

where  $k_\phi$  the same positive gain as in (4b). Thus the FHOCP of calculating the open-loop input  $\bar{\mathbf{u}}_i^*$  comes down to calculating the optimum deviation  $\theta_i^*$  at each iteration:

$$J_i(t, \mathbf{q}_i, \mathbf{u}_i, T) = \int_t^{t+T} \Lambda_i(\mathbf{q}_i(\tau), \mathbf{u}_i(\tau)) d\tau + \Phi_i(t+T), \quad (7a)$$

$$J_i^*(t, \mathbf{q}_i, T) = \min_{\theta_i[t, t+T]} J_i(t, \mathbf{q}_i, \mathbf{u}_i, T), \quad (7b)$$

$$\theta_i^*[t, t+T) = \arg J_i^*(t, \mathbf{q}_i, T) \quad (7c)$$

Thus, performance criteria are introduced through  $J_i(\cdot)$ . The new term  $\bar{\mathbf{u}}_i$ , is added to an existing DNF-based feedback control law of each agent, which will be denoted as “ $-k(\mathbf{q}_i, \nabla \Phi_i)$ ” to simplify the notation, as follows:

$$\mathbf{u}_i \triangleq -k(\mathbf{q}_i, \nabla \Phi_i) + \bar{\mathbf{u}}_i \quad (8)$$

During the prediction phase, the  $i$ -th agent's control input has the form (8), with  $\bar{\mathbf{u}}_i$  being a function of  $\theta_i[t, t+T)$ . Once the FHOCP (7) is solved,  $\mathbf{u}_i^* \triangleq -k(\mathbf{q}_i, \nabla \Phi_i) + \bar{\mathbf{u}}_i^*$ , where  $\bar{\mathbf{u}}_i^* = [0 \ k_\phi \theta_i^* + \dot{\theta}_i^*]^\top$ , is applied over the control horizon.

### B. Randomized Finite Horizon Optimization

We denote by  $\theta_i^*(t)$  the concatenation of deviations:

$$\theta_i^*(t) \triangleq \theta_i^*[0, t_1^i] | \theta_i^*[t_1^i, t_2^i] | \dots | \theta_i^*[t_k^i, t_{k+1}^i] | \dots, \quad (9)$$

obtained iteratively by solving the FHOCP (7) and applying each  $\theta_i^*[t_k^i, t_{k+1}^i + T)$  over the time interval  $[t_k^i, t_{k+1}^i)$ ,  $k \in \mathbb{N}$ . Time instant  $t_k^i$  (recalculation time) denotes the  $k$ -th time agent  $i$  recalculates  $\theta_i^*$  by solving (7). Typically in MPC  $t_{k+1}^i - t_k^i = T_c$  (fixed), but in general the recalculation times satisfy:  $0 < t_{k+1}^i - t_k^i < T$ . Also,  $t_0^i = 0$  and  $\theta_i^*(0) \triangleq 0, \forall i \in \mathcal{N}$ .

By relaxing the need for optimality, a class of randomized algorithms used in distribution-free statistical learning methods can be used to solve the FHOCP (7). The basic idea is to generate a sufficient number of candidate solutions by sampling a set (according to some probability distribution  $\mathcal{P}$ ), simulate the system's dynamics using each candidate and select the one that performs best [9].

*Lemma 1 ([9], [10]):* The number of samples  $N_s$  that guarantees  $J_i^*$  is a “probable near minimum” of  $J_i$  to level  $\alpha$  and confidence  $1 - \delta$  satisfies:  $N_s \geq \ln(1/\delta) / \ln(1/1 - \alpha)$ .

In the following, the superscript “ $(\cdot)^*$ ” will be used to denote a probable near minimum (optimum) instead of the minimum (optimum). At each recalculation time  $t_k^i$ , agent  $i \in \mathcal{N}$  calculates a near optimum deviation  $\theta_i^*[t_k^i, t_{k+1}^i + T)$  by executing Algorithm 1.

---

### Algorithm 1

---

Parameters:  $\alpha, \delta, \Theta, \mathcal{P}(\theta), T, J(\cdot)$

- 1: Calculate a sufficient # of samples  $N_s$  (Lemma 1).
  - 2: Generate  $N_s$  random samples  $\theta_i^m$ ,  $m = 1, \dots, N_s$  from a set  $\Theta \subset (-\frac{\pi}{2}, +\frac{\pi}{2})$  according to  $\mathcal{P}(\theta)$ .
  - 3: Generate each candidate deviation  $\theta_i^m[t_k^i, t_k^i + T)$  as:  $\theta_i^m[t_k^i, t_k^i + T) = (1 - \frac{\tau}{T})\theta_i^*(t_k^i) + (\frac{\tau}{T})\theta_i^m$ , where  $\tau \in [0, T)$  (dummy time variable).
  - 4: Simulate the (multi-agent) system's dynamics over the horizon  $[t_k^i, t_k^i + T)$  using each candidate deviation.
  - 5: Calculate the cost  $J_i(t_k^i, \mathbf{q}_i, \mathbf{u}_i, T)$  of each candidate.
  - 6: Pick  $\theta_i^*[t_k^i, t_k^i + T) = \arg \min_{\theta_i^m[t_k^i, t_k^i + T)} J_i(t_k^i, \mathbf{q}_i, \mathbf{u}_i, T)$ .
- 

Each candidate deviation (Step 3) is a line segment (1st degree polynomial) connecting the point  $(t_k^i, \theta_i^*(t_k^i))$  with one of the  $N_s$  sampled points  $(t_k^i + T, \theta_i^m(t_k^i + T))$  of Step 2. Higher degree polynomials would result in “smoother” deviations but the condition  $|\theta_i^*(t)| < \frac{\pi}{2}$  (proven in Lemma 2) would not hold for *any* value of the involved parameters.

Finally denote by  $t_f^i = \inf\{t : \|\mathbf{p}_i - \mathbf{p}_{id}\| \leq r_0\}$  the time instant when agent  $i$  enters a neighbourhood  $r_0$  of its destination  $\mathbf{p}_{id}$ . For  $t \geq t_f^i$ ,  $\theta_i$  will be given as a function of the distance  $S_i = \|\mathbf{p}_i - \mathbf{p}_{id}\|$ :  $\theta_i(t \geq t_f^i) = \theta_i(S_i) = S_i^2 \cdot \theta_i^*(t_f^i) / r_0^2$ . We could say that the predictive controller is “turned-off”. This deviation term has the following properties:  $\theta_i(S_i = r_0) = \theta_i^*(t_f^i)$ ,  $\theta_i(S_i = 0) = 0$ , and  $\dot{\theta}_i(S_i = 0) = d\theta_i/dS_i \cdot dS_i/d\mathbf{p}_i \cdot \dot{\mathbf{p}}_i|_{S_i=0} = 0$ . Therefore  $\theta_i^*(t)$  is restated as follows:

$$\theta_i^*(t) \triangleq \theta_i^*[0, t_1^i] | \dots | \theta_i^*[t_k^i, t_{k+1}^i] | \dots | \theta_i(t \geq t_f^i). \quad (10)$$

*Lemma 2:* The deviation (10), where each  $\theta_i^*[t_k^i, t_{k+1}^i + T)$  is calculated by Algorithm 1 and applied over the time intervals  $[t_k^i, t_{k+1}^i)$ ,  $0 < t_{k+1}^i - t_k^i < T$ , is a continuous function of time (class  $C^0$ ) that satisfies  $|\theta_i^*(t)| < \frac{\pi}{2}, \forall i \in \mathcal{N}$ .

*Proof:* The continuity part of Lemma 2 is a direct result of Algorithm 1, Step 3 and the definition of  $\theta_i(t \geq t_f^i)$ . Thus  $\theta_i^*(t)$  given by (10), is a continuous function of time.

We have  $-\frac{\pi}{2} < \theta_i^*[0, t_1^i] | \theta_i^*[t_1^i, t_2^i] | \dots | \theta_i^*[t_k^i, t_{k+1}^i] | \dots < \frac{\pi}{2}$  since  $\theta_i^*(0) \triangleq 0$ , each  $\theta_i^*(t_k^i + T) \in \Theta \subset (-\frac{\pi}{2}, +\frac{\pi}{2})$  and each pair  $\theta_i^*(t_k^i)$ ,  $\theta_i^*(t_{k+1}^i)$  is connected by a line segment,  $\forall k \in \mathbb{N}$ . Finally  $|\theta_i(t \geq t_f^i)| < |\theta_i^*(t_f^i)| < \frac{\pi}{2}$  since  $\theta_i(t_f^i) = \theta_i^*(t_f^i)$ ,  $\theta_i(S_i = 0) = 0$  and the only critical point of  $\theta_i(t \geq t_f^i)$  in  $S_i \in [0, r_0]$  is at  $S_i = 0$  by construction,  $\dot{\theta}_i(S_i = 0) = d\theta_i/dS_i \cdot dS_i/d\mathbf{p}_i \cdot \dot{\mathbf{p}}_i|_{S_i=0} = 0$ . Thus (10) also satisfies  $|\theta_i^*(t)| < \frac{\pi}{2}$ . ■

### C. Control Laws

Choose “ $-k(\mathbf{q}_i, \nabla \Phi_i)$ ” in Eq. (8) to be the control law (4). From (6), the new control law  $\mathbf{u}_i = -k(\mathbf{q}_i, \nabla \Phi_i) + \bar{\mathbf{u}}_i^*$ ,  $\forall i \in \mathcal{N}$ , is stated below (11). The term  $\bar{\mathbf{u}}_i^*$  is calculated by solving the FHOCP (7) at time instants  $t = t_k^i$  and substituting  $\theta_i^*[t_k^i, t_k^i + T)$ , and its slope  $\dot{\theta}_i^*$  over  $[t_k^i, t_k^i + T)$ , in (6).

$$\mathbf{v}_i = \begin{cases} -s_i \cdot U_i, & \frac{\partial \Phi_i}{\partial t} \leq U_i(|P_i| - \varepsilon) \\ -s_i \cdot \frac{\partial \Phi_i + \varepsilon U_i}{|P_i|}, & \frac{\partial \Phi_i}{\partial t} > U_i(|P_i| - \varepsilon) \end{cases} \quad (11a)$$

$$\omega_i = -k_\phi(\phi_i - \phi_{nhi} - \theta_i^*) + \dot{\phi}_{nhi} + \dot{\theta}_i^* \quad (11b)$$

#### D. Sampling Set

The nonholonomic nature of the unicycle and the control law (11b) make it impossible to accurately track a desired heading angle. A tracking error,  $(\phi_i - \phi_{nhi} - \theta_i^*)$ , will be present at all times. Thus,  $|\theta_i^*(t)| < \frac{\pi}{2}$  (Lemma 2), does not also guarantee that  $|\phi_i - \phi_{nhi}| < \frac{\pi}{2}$ . Satisfaction of this condition is necessary to preserve the navigation properties of the original control law (4). At each recalculation time, the sampling set  $\Theta$  in Algorithm 1 will be adjusted accordingly in order to take into account the tracking error.

*Theorem 1:* Consider an agent (1) under (11b) and denote by  $\psi_i$  the angle between the field's gradient and the agent's longitudinal axis. At each recalculation time  $t_k^i$ , let  $\Theta \leftarrow \Theta_k^i = (-\frac{\pi}{2} + |\psi_i(t_k^i) - \theta_i^*(t_k^i)|, +\frac{\pi}{2} - |\psi_i(t_k^i) - \theta_i^*(t_k^i)|)$  in Algorithm 1. Then, if  $|\psi_i| = |\phi_i - \phi_{nhi}|$  is initially less than  $\frac{\pi}{2}$ , it will always remain in  $[0, \frac{\pi}{2})$ .

*Proof:* Consider the dynamics of the term  $(\psi_i - \theta_i^*)$  (which is continuous because of Lemma 2) over  $\tau \in [0, T)$ :  $d(\psi_i - \theta_i^*)/dt = \dot{\psi}_i - \dot{\theta}_i^* = \omega_i - \dot{\phi}_{nhi} - \dot{\theta}_i^* = -k_\phi(\psi_i - \theta_i^*) \Rightarrow \dot{\psi}_i = -k_\phi(\psi_i - \theta_i^*) + \dot{\theta}_i^* \Rightarrow \psi_i(\tau) = [\psi_i(0) - \theta_i^*(0)]e^{-k_\phi\tau} + \theta_i^*(\tau)$ . Therefore:  $|\psi_i(\tau)| \leq |\psi_i(0) - \theta_i^*(0)|e^{-k_\phi\tau} + |\theta_i^*(\tau)| \leq |\psi_i(0) - \theta_i^*(0)| + |\theta_i^*(\tau)|$ . From Algorithm 1,  $\theta_i^*(T) \in \Theta_0^i \Rightarrow |\theta_i^*(\tau)| \in [0, \frac{\pi}{2} - |\psi_i(0) - \theta_i^*(0)|)$ . Thus,  $|\psi_i(\tau)| < |\psi_i(0) - \theta_i^*(0)| + \frac{\pi}{2} - |\psi_i(0) - \theta_i^*(0)| \Rightarrow |\psi_i(\tau)| < \frac{\pi}{2}$ ,  $\tau \in [0, T)$ . This results in  $|\psi_i(t_k^i)| < \frac{\pi}{2}$  since  $t_k^i \in [0, T)$  and thus  $|\psi_i(t)| < \frac{\pi}{2}$ ,  $\forall t > 0$ , since  $|\theta_i^*(t_k^i)| < \frac{\pi}{2}$  from Lemma 2. ■ Finally, note that  $\Theta_k^i \neq \emptyset$  and  $\Theta_k^i \subset (-\frac{\pi}{2}, +\frac{\pi}{2})$ ,  $\forall k \in \mathbb{N}$ .

### IV. DECENTRALIZATION

#### A. Nominal Multi-Agent System

In a decentralized setting, limited information is a source of uncertainty wrt each agent's predictions. We assume the following are available for executing Algorithm 1: (I) Each agent can measure the configuration (position & orientation,  $\mathbf{q}_i$ ) of all other agents at each time instant, (II) Each agent can measure the speed (linear velocity  $v_i$ ) of all other agents at each time instant, (III) Agents broadcast their target configuration  $\mathbf{q}_{id}$  to all other agents at  $t = t_0^i = 0$ , (IV) Agents' predictive controllers do not exchange any information regarding their decisions (future state or input trajectories and deviation trajectory). Assumptions I, II are present in similar settings, i.e., when DNFs are used for the navigation of nonholonomic agents [2]. The information exchange in Assumption III is minimal compared to the broadcasting of future state trajectories. Assumption IV is what differentiates our decentralized scheme from a distributed [5] or cooperative approach [7].

We define as *nominal* multi-agent system, wrt  $i$ , the one whose agents  $j \in \mathcal{N}$  are described by (1), under (11), with:

$$\theta_j[t_k^i, t_k^i + T) = \begin{cases} \theta_j^m[t_k^i, t_k^i + T), & j = i \\ 0, & j \neq i \end{cases}, \quad (12a)$$

$$\hat{U}_{jd} = \begin{cases} U_{id}, & j = i \\ v_j(t_k^i), & j \neq i \end{cases}, i, j \in \mathcal{N}. \quad (12b)$$

Deviations  $\theta_j^m[t_k^i, t_k^i + T)$  correspond to the samples generated by Algorithm 1. Regarding the prediction phase of agent  $i$ , it starts at the initial conditions:  $\hat{\mathbf{q}}_j(t_k^i) = \mathbf{q}_j(t_k^i)$  and  $\hat{v}_j(t_k^i) = v_j(t_k^i)$ ,  $\forall j \in \mathcal{N}$ , i.e., the predicted trajectories at  $\tau = t_k^i$  are equal to the measurements made by agent  $i$  at  $t = t_k^i$ . The rest of the predicted trajectories are derived by calculating the direction of  $\nabla_j \Phi_j$  and applying control laws (11a), (11b), as stated above, over  $[t_k^i, t_k^i + T)$ ,  $\forall j \in \mathcal{N}$ . Therefore uncertainty enters the system since, at each  $t_k^i$ , agent  $i$  solves the FHOC (7) by simulating the dynamics of the *nominal* multi-agent system, i.e., without taking into account the future decisions of other agents (deviation from  $\nabla_j \Phi_j$ ,  $j \neq i$ ) and their actual desired absolute speed,  $U_{jd}$ . In other words, agent  $i$ 's prediction is based on the nominal system's assumptions (12), not the actual evolution of the multi-agent system.

#### B. Event-Triggered Execution

We introduce event-triggered execution of the predictive navigation scheme (independently on each agent), in order to tackle the deterioration in performance, caused by the uncertainty described in §IV-A. For each agent  $i$ , denote by

$$\hat{C}_i(\tau) = \int_{t_k^i}^\tau \Lambda_i(\hat{\mathbf{q}}_i(\tau), \hat{\mathbf{u}}_i^*(\tau)) dt, \quad \tau \in [t_k^i, t_k^i + T), \quad (13a)$$

$$C_i(\tau) = \int_{t_k^i}^\tau \Lambda_i(\mathbf{q}_i(\tau), \mathbf{u}_i^*(\tau)) dt, \quad \tau \in [t_k^i, t_k^i + T) \quad (13b)$$

the predicted and real running costs respectively, where  $\hat{\mathbf{q}}_i, \hat{\mathbf{u}}_i^*, \mathbf{q}_i, \mathbf{u}_i^*$  correspond to the predicted and real state and input trajectories of agent  $i$  while applying the optimum (wrt the nominal system) deviation  $\theta_i^*[t_k^i, t_k^i + T) = \arg J_i^*(t_k^i, \mathbf{q}_i, T)$ . Due to the uncertainty caused by decentralization, there will be a discrepancy between the predicted and actual running costs. An execution rule will be introduced to derive the recalculation times (event times)  $\{t_1^i, t_2^i, \dots, t_k^i, \dots\}$ . Its goal will be to maintain the discrepancy below some bound. Consider the inequality:  $C_i(\tau) \geq \hat{C}_i(\tau) + c_\epsilon$ , where  $c_\epsilon > 0$  is the cost discrepancy bound.

Then an appropriate execution rule for the event times is:

$$t_{k+1}^i = \begin{cases} \tau_c, & \tau_c := \inf\{\tau : C_i(\tau) \geq \hat{C}_i(\tau) + c_\epsilon\} \\ t_k^i + T_c, & \text{if } C_i(t_k^i + T_c) < \hat{C}_i(t_k^i + T_c) + c_\epsilon \end{cases} \quad (14)$$

where  $T_c < T$  is the (maximum) control horizon. Execution rule (14) requires the storing of the predicted cost  $\hat{C}_i(\tau)$ , corresponding to  $\theta_i^*[t_k^i, t_k^i + T)$ , and monitoring of the above inequality over the time interval  $[t_k^i, t_k^i + T)$ .

Let the running cost function  $\Lambda_i(\cdot)$  be quadratic:  $\Lambda_i(\cdot) = (\mathbf{q}_i - \mathbf{q}_{id})^\top \mathbf{Q}(\mathbf{q}_i - \mathbf{q}_{id}) + R_1(|v_i| - U_i)^2 + R_2\omega_i^2$ , where  $\mathbf{Q}, R_1, R_2$  are constant weighting parameters.

*Theorem 2:* For a quadratic  $\Lambda_i(\cdot)$  and a bound  $c_\epsilon$ , the recalculation times  $\{t_1^i, \dots, t_k^i, \dots\}$ , implicitly defined by the execution rule (14) satisfy  $0 < c_\epsilon/\Lambda_{Zi,k}^{max} \leq t_{k+1}^i - t_k^i < T$ , where  $\Lambda_{Zi,k}^{max} \geq \Lambda_i(\cdot) - \Lambda_i(\cdot)$ ,  $\tau \in [t_k^i, t_k^i + T)$ ,  $\forall k \in \mathbb{N}$  and  $i \in \mathcal{N}$ .

*Proof:* Denote the cost discrepancy by  $Z_i(\tau) = C_i(\tau) - \hat{C}_i(\tau) = \int_{t_k^i}^\tau [\Lambda_i(\cdot) - \Lambda_i(\cdot)] dt \triangleq \int_{t_k^i}^\tau \Lambda_{Zi,k}(\tau)$ , where  $\Lambda_{Zi,k}(\tau) \triangleq \Lambda_i(\mathbf{q}_i(\tau), \mathbf{u}_i^*(\tau)) - \Lambda_i(\hat{\mathbf{q}}_i(\tau), \hat{\mathbf{u}}_i^*(\tau))$ .

The following conditions hold:  $\Lambda_{Zi,k}$  is continuous in  $\mathbf{q}_i$ ,  $\mathbf{u}_i$ ,  $\mathbf{q}_i$  is continuous in time,  $|v_i|$  is continuous at all switching instants and  $\omega_i$  is continuous in  $[t_k^i, t_k^i + T)$ . Therefore,  $Z_i(\tau)$  is differentiable in  $[t_k^i, t_k^i + T)$ , with  $\dot{Z}_i(\tau) = \Lambda_{Zi,k}(\tau)$ . Also note that  $Z_i(\tau = t_k^i) = 0$ , i.e. the discrepancy is zero at the beginning of the control phase. In addition, let  $v_{max} = \max_{\tau} \{|v_i(\tau)|\}$ . Thus,  $|v_i|$  is upper bounded by  $v_{max}$ . It is proven in Corollary 1 that  $|v_i| \rightarrow \infty \Rightarrow v_{max} \rightarrow \infty$ . The angular velocity  $\omega_i$  is bounded,  $\|\mathbf{p}_i - \mathbf{p}_{id}\| < 2R_w$  and  $\phi_i \in (-\pi, \pi)$ . Thus  $\|\mathbf{q}_i - \mathbf{q}_{id}\|$  is also bounded. Therefore  $\exists \Lambda_{Zi,k}^{max} \geq \Lambda_{Zi,k}(\tau)$ , s.t.  $\dot{Z}_i^{max} = \Lambda_{Zi,k}^{max}$ . Then the inter-event times satisfy  $0 < c_e / \Lambda_{Zi,k}^{max} \leq t_{k+1}^i - t_k^i$ . Finally, the  $t_{k+1}^i - t_k^i < T$  side is trivially satisfied by the execution rule (14) since  $T_c < T$ . ■

## V. ANALYSIS

*Corollary 1:* Since by Theorem 1  $|\phi_i - \phi_{nhl}| \in [0, \frac{\pi}{2})$ , the projection of the field's gradient on the agent's longitudinal axis,  $P_i$ , is never zero for  $\mathbf{p}_i \neq \mathbf{p}_{id}$ . Therefore, the linear velocity  $v_i$  in (11a) does not go to infinity.

*Proof:*  $|P_i| = |\nabla_i \Phi_i^\top \cdot \mathbf{I}_i| = \|\nabla_i \Phi_i\| \cdot |\cos(\phi_i - \phi_{nhl})|$ . For  $\|\nabla_i \Phi_i\| = 0$  to hold, agent  $i$  must have arrived at its destination  $\mathbf{p}_i = \mathbf{p}_{id}$  [16]. For  $|\cos(\phi_i - \phi_{nhl})| = 0$  to hold, the agent's longitudinal axis must be normal to the field's gradient  $\nabla_i \Phi_i$ , i.e.,  $|\phi_i - \phi_{nhl}| = \frac{\pi}{2}$ , which contradicts  $|\phi_i - \phi_{nhl}| \in [0, \frac{\pi}{2})$ . Therefore  $|P_i| \neq 0 \Rightarrow |v_i| \rightarrow \infty$  in (11a). ■

*Corollary 2:* Theorem 1 implies that if an agent starts in the subspace behind its target ( $d_i < 0$ ) with the initial negated gradient vector driving it forward ( $P_i < 0$ ), only forward motion will be used for navigation and conflict resolution (a necessary condition for application to ATC).

*Proof:* The condition  $d_i < 0$  substituted in the definition of  $\phi_{nhl}$  results in:  $\phi_{nhl} = \text{atan2}(-\Phi_{iy}, -\Phi_{ix})$ , meaning that  $\phi_{nhl}$  defines exactly the angle of  $-\nabla_i \Phi_i$ .

Initially  $P_i < 0$  therefore the agent's heading,  $\phi_i$ , is initially in the direction of  $-\nabla_i \Phi_i$ , i.e.,  $\phi_i \in (\phi_{nhl} - \frac{\pi}{2}, \phi_{nhl} + \frac{\pi}{2})$ . From Corollary 1 we have  $P_i \rightarrow 0$ . Thus  $P_i$  remains negative and  $-s_i = -\text{sgn}(P_i) > 0$  always holds. As a result the linear velocity (11a) is always positive (forward motion). ■

The requirements of Corollary 2 are mild and represent reasonable physical conditions.

*Theorem 3:* A team of agents described by (1) under the control law (11a) remains always *safe*, i.e., no conflicts (loss of separation) occur and no agent collides with  $\partial W$ .

*Proof:* (Sketch, see [17]) Since agents are circular, conflicts can only occur by translation. By construction  $\Phi_i = 1$  on  $\partial W$  or when separation is lost. But  $\Phi_i|_{t=0} < 1$ ,  $\forall i \in \mathcal{N}$ , and  $v_i$  in Eq. (11a) guarantees  $\dot{\Phi}_i < 0$ ,  $\forall i \in \mathcal{N}$ . ■

*Theorem 4:* The multi-agent system described by (1), under the control laws (11a), (11b), admits a continuous Lyapunov function,  $V$ . In addition, each agent  $i$  converges to its target destination  $\mathbf{p}_{id}$  with the desired orientation  $\phi_{id}$ .

*Proof:* Consider the finite, strictly increasing (because of Lemma 2) sequence of recalculation time instants:  $\pi := \{t_0, t_1, \dots, t_k, \dots, t_f\}$ ,  $k \in \mathbb{N}$ , where  $t_k$  denotes the time instant *at least one* agent triggered an execution of Algorithm 1 (recalculation time) or entered the neighbourhood  $r_0$  of its destination and  $t_0 \triangleq 0$ . The last element in  $\pi$ ,  $t_f$ , denotes the

time instant at which the *last* agent entered the neighbourhood  $r_0$  of its destination. On each time interval  $[t_k, t_{k+1})$ ,  $t_k \neq t_f$ , we employ the following Lyapunov function candidate:  $V_k = \sum_{i=1}^N V_{ik}$ ,  $V_{ik} = \Phi_i + \frac{1}{2}(\phi_i - \phi_{nhl} - \theta_i^*[t_k, t_{k+1}])^2$ , and consider the extended multi-agent system  $\dot{\mathbf{x}} = f(\mathbf{x})$ :

$\mathbf{x} = [\mathbf{p}_1^\top \dots \mathbf{p}_N^\top \phi_1 \dots \phi_N \phi_{nh1} \dots \phi_{nhN} \theta_1^* \dots \theta_N^*]^\top$ ,  $f(\mathbf{x}) = [v_1 \cdot \mathbf{I}_1^\top \dots v_N \cdot \mathbf{I}_N^\top \omega_1 \dots \omega_N \dot{\phi}_{nh1} \dots \dot{\phi}_{nhN} \dot{\theta}_1^* \dots \dot{\theta}_N^*]^\top$ . In order to apply the chain rule in [18], we employ the Filippov set [19]  $K[f(\mathbf{x})]$  and the generalized derivative [20] of  $V_k(\mathbf{x})$ :

$$K[f] = \begin{bmatrix} K[v_1] \mathbf{I}_1 \\ \vdots \\ K[v_N] \mathbf{I}_N \\ \omega_1 \\ \vdots \\ \omega_N \\ \dot{\phi}_{nh1} \\ \vdots \\ \dot{\phi}_{nhN} \\ \dot{\theta}_1^* \\ \vdots \\ \dot{\theta}_N^* \end{bmatrix}, \quad \partial V_k = \begin{bmatrix} \sum_i \nabla_i \Phi_i \\ \vdots \\ \sum_i \nabla_i \Phi_i \\ (\phi_1 - \phi_{nh1} - \theta_1^*) \\ \vdots \\ (\phi_N - \phi_{nhN} - \theta_N^*) \\ -(\phi_1 - \phi_{nh1} - \theta_1^*) \\ \vdots \\ -(\phi_N - \phi_{nhN} - \theta_N^*) \\ -(\phi_1 - \phi_{nh1} - \theta_1^*) \\ \vdots \\ -(\phi_N - \phi_{nhN} - \theta_N^*) \end{bmatrix}$$

We calculate the generalized time derivative of  $V_k(\mathbf{x})$ :  $\dot{V}_k = \bigcap_{\xi \in \partial V} \xi^\top K[f] = \sum_i^N \sum_j^N K[v_i] \nabla_i \Phi_j^\top \mathbf{I}_i + \sum_i^N \omega_i \cdot (\phi_i - \phi_{nhl} - \theta_i^*) - \sum_i^N \dot{\phi}_{nhl} \cdot (\phi_i - \phi_{nhl} - \theta_i^*) - \sum_i^N \dot{\theta}_i^* \cdot (\phi_i - \phi_{nhl} - \theta_i^*) = \sum_i K[v_i] P_i + \sum_i \sum_{j \neq i} K[v_j] \nabla_j \Phi_i^\top \mathbf{I}_j - \sum_i k_\phi (\phi_i - \phi_{nhl} - \theta_i^*)^2$ .

We discriminate between the following two sets of agents:  $\mathcal{N}_1 \triangleq \{i \in \mathcal{N} \mid \frac{\partial \Phi_i}{\partial t} \leq U_i(|P_i| - \varepsilon)\}$  and  $\mathcal{N}_2 \triangleq \{i \in \mathcal{N} \mid \frac{\partial \Phi_i}{\partial t} > U_i(|P_i| - \varepsilon)\}$ ,  $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$ . From (11a):

$$K[v_i] = \begin{cases} -K[s_i] \cdot U_i, & i \in \mathcal{N}_1 \\ -K[s_i] \cdot \frac{\partial \Phi_i + \varepsilon U_i}{|P_i|}, & i \in \mathcal{N}_2 \end{cases}$$

We can now proceed with the derivation:  $\dot{V}_k = \sum_{\mathcal{N}_1} \left\{ -K[s_i] P_i U_i + \frac{\partial \Phi_i}{\partial t} \right\} + \sum_{\mathcal{N}_2} \left\{ -K[s_i] P_i \frac{U_i \varepsilon + \frac{\partial \Phi_i}{\partial t}}{|P_i|} + \frac{\partial \Phi_i}{\partial t} \right\} - \sum_{\mathcal{N}} k_\phi \cdot (\phi_i - \phi_{nhl} - \theta_i^*)^2 = -\sum_{\mathcal{N}_1} \left\{ |P_i| U_i - \frac{\partial \Phi_i}{\partial t} \right\} - \sum_{\mathcal{N}_2} U_i \varepsilon - \sum_{\mathcal{N}} k_\phi (\phi_i - \phi_{nhl} - \theta_i^*)^2 < 0$ . We deduced that  $0 \notin \dot{V}_k$  because in  $\mathcal{N}_1$ ,  $|P_i| U_i - \frac{\partial \Phi_i}{\partial t} \geq U_i \varepsilon$ , and for both  $\mathcal{N}_1, \mathcal{N}_2$ , we have that  $\sum_{\mathcal{N}} U_i > 0$  because  $\mathbf{p}_i \neq \mathbf{p}_{id}$  holds for at least one agent, since  $t < t_f$ . Now consider a recalculation time  $t_k \in \pi$ . For  $j \in \mathcal{N}$  that triggered at  $t = t_k = t_k^j$ , we have

$$\theta_j^*[t_{k-1}^j, t_k^j] = \theta_j^*[t_k^j, t_{k+1}^j] \Rightarrow \theta_j^*[t_{k-1}^j, t_k] = \theta_j^*[t_k, t_{k+1}^j],$$

The same holds if  $t_k = t_f^j$ , as proven in Lemma 2.

Using this result in  $V_k$  we get:  $V_{k-1}(t_k) = V_k(t_k)$ , i.e., the multiple Lyapunov functions [21]  $V_k$  are equal at  $t = t_k$ . It is easy to show that the same also holds for  $t = t_f$ . Finally, let  $V_f = \sum_{i=1}^N \Phi_i + \frac{1}{2}(\phi_i - \phi_{nhl} - \theta_i(t \geq t_f))^2$ . Using the same analysis, we deduce that  $0 \in \dot{V}_f$ . This is possible since for  $t > t_f$ , all agents have entered the neighbourhoods  $r_0$  of  $\mathbf{p}_{id}$ , making  $U_i = 0$  possible  $\forall i \in \mathcal{N}$ . Therefore the concatenation

$V = V_0|V_1| \dots |V_k| \dots |V_f|$  is a continuous, strictly decreasing Lyapunov function for the multi-agent system [21, p. 53].

Since each  $V_{ik}$ , and consequently  $V$ , is regular and the level sets of  $V$  are compact, we apply the nonsmooth version of LaSalle's invariance principle. Therefore, the multi-agent system converges to the largest invariant subset  $S$ :

$S \triangleq \left\{ [\mathbf{p}^\top, \phi]^\top \mid 0 \in \tilde{V} \right\}$ . For  $\tilde{V} = 0$  to hold, we get:

$S = \left\{ \mathbf{q} : (|P_i|U_i - \frac{\partial \Phi_i}{\partial t} = 0, \forall i \in \mathcal{N}_1) \wedge (U_i \varepsilon = 0, \forall i \in \mathcal{N}_2) \wedge (\phi_i - \phi_{nhi} - \theta_i^* = 0, \forall i \in \mathcal{N}) \right\}$ .

Since  $|P_i|U_i - \frac{\partial \Phi_i}{\partial t} \geq U_i \varepsilon \geq 0$  (condition of  $\mathcal{N}_1$ ), the equality  $U_i = 0$  must hold inside  $S$ , requiring  $\mathbf{p}_i = \mathbf{p}_{id}$  so that  $\phi_{nhi} = \phi_{id}$  (property of dipolar DNFs), and  $\theta_i^* = 0, \forall i \in \mathcal{N}$ . Thus,  $\phi_i = \phi_{id}, \forall i$ . Therefore,  $S$  reduces to the singleton:  $\left\{ \mathbf{q} : (\mathbf{p}_i = \mathbf{p}_{id}, \forall i \in \mathcal{N}) \wedge (\phi_i = \phi_{id}, \forall i \in \mathcal{N}) \right\}$ , i.e., all agents converge to their destinations with the desired orientation. ■

## VI. SIMULATION

We present comparative results for a navigation scenario involving 4 aircrafts. For all agents,  $U_{id} = 454$  knots,  $r_i = 2.5$  nm, and we use,  $\forall i \in \mathcal{N}$ , the cost functional  $J_i(t, \mathbf{q}_i, \mathbf{u}_i, T) =$

$$\int_t^{t+T} [Q \|\mathbf{p}_i - \mathbf{p}_{id}\|^2 + R_1 (|v_i| - U_i)^2] d\tau + \Phi_i(t+T), \quad (15)$$

where  $T = 10$  min,  $T_c = 1$  min,  $c_\varepsilon = \frac{1}{T_c} \cdot \hat{C}_i(t_k^i + T_c)$ ,  $N_s = 22$  ( $\alpha = 0.10$ ,  $\delta = 0.10$ ) and  $\mathcal{P}$  is the uniform distribution.

## VII. CONCLUSIONS AND FUTURE WORK

A decentralized multi-agent navigation and conflict resolution, scheme that considers the issue of performance, was presented. It employs MPC to calculate deviations from the direction of a dipolar DNF's gradient. Event-triggered recalculation of the deviation trajectories is used to tackle the discrepancy between the predicted and actual performance, caused by the limited exchange of information between agents. It is proven that the proposed scheme preserves the navigation properties of the original DNF-based approach.

Future research is towards further decentralization and improved scalability by employing a limited sensing range.

## REFERENCES

- [1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. Journal of Robotics Research*, vol. 5, pp. 90–98, 1986.
- [2] S. G. Loizou, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Decentralized feedback stabilization of multiple nonholonomic agents," in *IEEE Intl. Conf. on Robotics and Automation*, 2003, pp. 3012–3017.
- [3] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–508, 1992.
- [4] W. B. Dunbar and R. M. Murray, "Distributed receding-horizon control with application to multi-vehicle formation stabilization," *Automatica*, vol. 42, pp. 549–558, 2006.
- [5] G. Roussos, G. Chaloulos, K. Kyriakopoulos, and J. Lygeros, "Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation functions," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, dec. 2008, pp. 1225–1230.
- [6] Y. Kuwata, A. Richards, T. Schouwenaars, and J. How, "Decentralized robust receding horizon control for multi-vehicle guidance," in *American Control Conference, 2006*, June 2006, p. 6 pp.
- [7] E. Franco, T. Parisini, and M. M. Polycarpou, "Design and stability analysis of cooperative receding-horizon control of linear discrete-time agents," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10–11, pp. 982–1001, 2007.

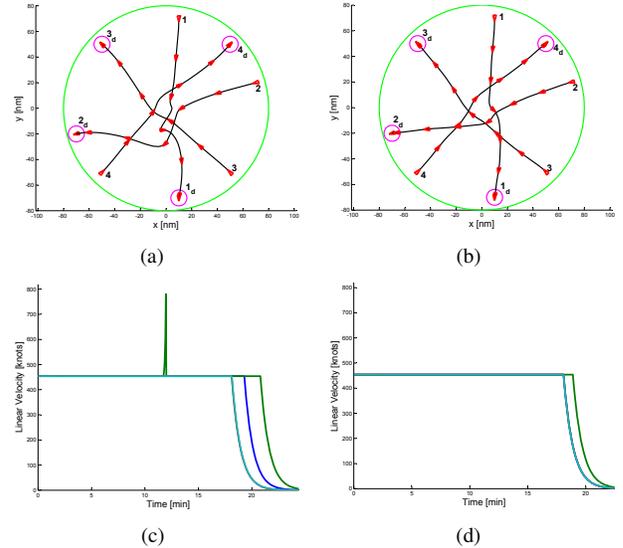


Fig. 1. A 4-agent scenario. The trajectories resulting from the DNF-based (left) and the proposed (right) schemes are depicted in 1(a)–1(b) respectively. The red triangles, representing agent position and orientation, are not to scale. In accordance with the cost functional (15), the linear velocity was maintained equal to the nominal speed  $U_i$ , as depicted in Fig. 1(d).

- [8] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. J. Balas, "Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 19–33, 2008.
- [9] J. L. Piovesan and H. G. Tanner, "Randomized model predictive control for robot navigation," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1817–1822.
- [10] M. Vidyasagar, "Randomized algorithms for robust controller synthesis using statistical learning theory," *Automatica*, vol. 37, pp. 1515–1528, 2001.
- [11] H. Tanner and J. Piovesan, "Randomized receding horizon navigation," *Automatic Control, IEEE Transactions on*, vol. 55, no. 11, pp. 2640–2644, Nov. 2010.
- [12] A. Eqtami, D. Dimarogonas, and K. Kyriakopoulos, "Event-triggered strategies for decentralized model predictive controllers," *IFAC World Congress*, 2011.
- [13] P. Varutti, T. Faulwasser, B. Kern, M. Kogel, and R. Findeisen, "Event-based reduced-attention predictive control for nonlinear uncertain systems," in *Proc. IEEE Int Computer-Aided Control System Design Symposium*, 2010, pp. 1085–1090.
- [14] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, pp. 723–731, 2009.
- [15] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 53–64, 2002.
- [16] S. G. Loizou, D. V. Dimarogonas, K. J. Kyriakopoulos, and M. M. Zavlanos, "A feedback stabilization and collision avoidance scheme for multiple independent non-point agents," *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.
- [17] G. Roussos and K. J. Kyriakopoulos, "Decentralised navigation and collision avoidance for aircraft in 3D space," *2010 American Control Conference, Baltimore, USA*, 2010.
- [18] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 9, pp. 1910–1914, 1994.
- [19] A. Filippov, *Differential equation with discontinuous right-hand sides*. Kluwer Academic Publishers, 1998.
- [20] F. H. Clarke, Y. S. Ledyaev, R. J. Stern, and P. R. Wolenski, *Nonsmooth analysis and control theory*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1998.
- [21] D. Liberzon, *Switching in Systems and Control*. Birkhauser, 2003.