Completely Decentralised Navigation of Multiple Unicycle Agents with Prioritisation and Fault Tolerance

Giannis Roussos and Kostas J. Kyriakopoulos

***Abstract*— We propose an algorithm for decentralised navigation of multiple independent agents, applicable to Robotics and Air Traffic Control (ATC). We present completely decentralised Navigation Functions that are used to build potential fields and consequently feedback control laws. Our approach employs local sensing, limited by a maximum sensing range and integrates priorities in the Navigation Function (NF) construction. Static and moving obstacles are taken into account, as well as agents that are unable to maneuver. A decentralised feedback control law is used, based on the gradient of the potential field, ensuring convergence and collision avoidance for all agents while respecting a lower velocity bound. An upper limit for the convergence time is given and simulation results are presented to demonstrate the efficacy of the proposed algorithm.**

I. Introduction

Multi-agent systems have gained a lot of attention in the last decade and are becoming increasingly popular in a number of different applications. A large part of the literature in this area focuses on achieving cooperative tasks, like formation control and flocking, see for example [1], [2]. A somewhat different class of problems is that of decentralised navigation, where each agent pursues an independent task but shares a common workspace. Two major applications for this class of problems are mobile robot path planning and automated aircraft navigation and collision avoidance. In both of these problems an increased level of decentralisation is desired to allow for greater performance, computational efficiency and robustness with respect to agent failures.

A wide variety of methods for robot navigation has emerged, employing various techniques. One such approach handles the problem in two steps [3]: the workspace is initially divided into cells, which are then used to formulate the navigation problem as a graph search problem. Artificial potential or vector fields are used to steer the agents between cells, following the sequence provided by the graph search. An extension of this scheme to multi-agent navigation is presented in [4]. Although this class of solutions provides an intuitive line of thought, it requires considerable pre-calculations and thus a-priori knowledge. Moreover, performing the cell decomposition in the combined state space of all agents and solving the graph search problem can become computationally challenging for large groups of agents.

A different class of methods uses artificial potential fields [5] to directly derive feedback controllers steering the agents over the entire workspace. A common weakness of artificial potential fields is the existence of local minima away from the goal that can prevent convergence. A special class of potential fields, Navigation Functions (NFs), have been introduced in [6], featuring a single, global minimum. The main advantages of this class of methods are the formal performance guarantees they can provide, computational efficiency and their real-time feedback nature, that can compensate for measuring and modeling errors. Navigation Functions have been so far applied of multi-agent problems ranging from robotic navigation [7] to ATC applications [8].

In this paper we further develop limited sensing in the NF methodology, which combined with a feedback control law designed on the principles of [8] yields a completely decentralised solution for multi-agent navigation and collision avoidance in a workspace with obstacles. Our approach requires no a-priori computation or knowledge and does not rely at all on centralised controllers. Each agent requires only its position within the workspace and knowledge about other agents and obstacles within a sensing range around it. Thus, our algorithm is completely distributed and its computational cost does not depend on the total number of agents.

Furthermore, we introduce priorities in the construction of the potential fields, as an additional design parameter: high priority agents are allowed to maintain right of way wrt lower priority ones. The priority scheme provides our algorithm with some fault tolerance, by assigning agents with limited or no maneuvering capability the highest priority. Prioritisation in ATC has been presented in [9] for a completely discrete solution. In this paper we introduce discrete priorities in the continuous NF framework, enabling the integration of moving obstacles in the algorithm. Moving obstacles have been also considered in [10], but their motion is assumed to be known a-priori, as the algorithm pre-calculates the complete trajectories of the agents.

The rest of this paper is organised as follows: Section II defines the problem considered, followed by Section III where the construction of the proposed potential field is described. In Section IV the feedback control scheme is presented. Finally, simulation results are given in V and the conclusions of the paper are summarised in Section VI.

II. Problem Statement

We assume a scenario involving N spherical agents of radius r_i described by the unicycle kinematic model:

$$\dot{\boldsymbol{q}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \mathbf{J}_i \cdot u_i,$$
$$\dot{\phi}_i = \omega_i,$$

Giannis Roussos and Kostas J. Kyriakopoulos are with the Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Greece. email: {jrous, kkyria}@mail.ntua.gr

where $\boldsymbol{q}_i = [\, x_i \quad y_i \,]^\top$ is agent's i position wrt a global frame $\mathcal{E}$, ϕ_i its heading angle between its longitudinal axis and the global x axis, and $\boldsymbol{J}_i = [\, \cos(\phi_i) \quad \sin(\phi_i) \,]^\top$. The control inputs are the linear velocity u_i and angular velocity ω_i. All agents are operating in a spherical workspace centered at the origin of $\mathcal{E}$ with radius R_w and can sense each other and obstacles within a sensing range R_s around them.

The objective is to drive each agent i to its destination $\boldsymbol{q}_{di}$ while avoiding collisions with other agents or obstacles. We want to enforce some form of prioritisation between agents, so that those with high priority can maintain right of way versus lower priority ones. Our goal is a completely decentralised solution, handling both static and moving obstacles.

III. Completely Decentralised Navigation Functions

Decentralisation in the NF methodology has been introduced by allowing each agent to ignore the targets of other agents and navigate using its own NF-generated potential field. Limited sensing is a key factor for decentralisation: it allows the use of onboard sensors with finite range and greatly limits the information that each agent needs to acquire and process, significantly improving the applicability and scalability of the algorithm in large scenarios. Limited sensing so far has been introduced in a number of ways in NFs. In [7] the authors use a C^0 sensing scheme, but assume a priori knowledge of the total number of agents. This requirement is removed in [11], where a switching sensing graph is used, resulting in a hybrid system. This approach does not ensure global stability, as blocking situations may be reached. Thus, convergence occurs only if the switching of the sensing graph eventually stops. A completely locally computable NF has been presented in [12], but only for single-agent problems and with the assumption that at each time instant there is at most one visible obstacle. This effectively means that the algorithm solves the collision with one obstacle at a time, which is not practical in a multi-agent scenario.

Our work here improves upon the above approaches, offering completely decentralised navigation for multiple independent agents with limited sensing. Moreover, prioritisation as well as static and moving obstacles have been also incorporated, to enable the application to a wider class of real problems, especially from the fields of Robotics and ATC. We propose an absolutely locally computable potential field that takes into account multiple agents according to their priorities, as well as static and moving obstacles. This potential field used in a control scheme such as the one presented in [8] can offer decentralised, non-cooperative navigation for multiple agents. In fact, any controller that ensures a decreasing rate for the potential's value over time is applicable. Thus, the use of the potential field presented here is not limited to unicycle agents but can also be applied to other types of agents (holonomic or non-holonomic), when combined with an appropriate control scheme.

Decentralised NFs have been of the form:

$$\Phi_i = \frac{\gamma_i + f_i}{\left((\gamma_i + f_i)^k + G_i \cdot \beta_i\right)^{1/k}}, \tag{1}$$

where the target function γ_i, cooperation function f_i, obstacle function G_i and workspace boundary function β_i depend on various euclidean distances, and have length units in some positive power. Especially $G_i = \prod_{j=1}^{N} g_{ij}$, based on $g_{ij} = g_{ij}\left(||\boldsymbol{q}_i - \boldsymbol{q}_j||^2\right)$, can vary in a very wide range within a single scenario. This introduces a number of difficulties:

- Tuning the NF parameters (eg. exponent k used to eliminate local minima) is difficult, depends on the scale of each problem and often requires extreme values.
- The overall behavior of the potential field becomes unpredictable, counter intuitive and impractical.
- High G_i values, combined with high k values that are required (see above), cause numerical problems.

We propose here to scale all distances using some reference lengths that are native to each problem setting. Thus we nondimensionalise the NF construction and derive a single potential field for a class of similar real problems.

Using dimensionless functions for the metrics γ_i, G_i and β_i to construct the potential (1) results in a more elegant and predictable behavior of the potential field, enabling easier parameter tuning and limiting numerical problems in simulations and experiments. Furthermore, the results of parameter tuning are valid for all similar problems. The benefits of the improvement we propose are not only practical; limited sensing by considering a finite sensing radius can now be implemented in a more natural way. Finally, prioritisation can be integrated in the potential field, enabling some agents to maintain right of way wrt lower priority agents.

A. Priority classes

We assume that each agent i, $i \in \{1, \ldots, N\}$ has an associated priority class $c_i \in \mathbb{N}$. Lower values of c_i represent higher priority, with $c_i = 0$ denoting either uncontrolled or faulty agents, or obstacles, that can be stationary or moving. We define the *threat set* T_i of agent i as the set of all agents (or obstacles) of the same or higher priority class, i.e. with the same or lower c_i: $T_i \triangleq \{j \in \{1, ., N\} \setminus \{i\} \,|c_j \leq c_i\}$. Priorities are used here in the following sense: each agent i takes into account all other agents and obstacles that belong to its threat set T_i, while ignoring agents of lower priority, i.e. agents with $c_j > c_i$. Thus, agents with high priority have right of way, while lower priority agents steer around them.

The higher priority class, $c_i = 0$ is reserved for obstacles (stationary or moving) and uncontrolled or faulty agents. Thus if an agent i is known to experience a degradation of its navigation and collision avoidance capabilities it is assigned the priority class $c_i = 0$, in order to have maximum priority and force all other normally operating agents to avoid it. Using priorities in this way means that two agents i and j have mutual sensing between them, i.e. they both take each other into account to navigate, $i \in T_j$ and $j \in T_i$, if and only if $c_i = c_j \neq 0$, i.e. they belong to the same priority class, other than the highest one. Otherwise, if one of the agents, say i, belongs to a higher priority class (even the highest one), $0 \leq c_i < c_j$, then $i \in T_j$ but $j \notin T_i$. Thus, at all combinations of c_i, c_j where at least one of them is nonzero, i.e. $max(c_i, c_j) > 0$, there is at least one-way sensing between agents i and j. As will be shown in the following, this ensures that all collisions will be avoided, at least by one of the two involved agents. Finally, when $c_i = c_j = 0$, both agents i and j are uncontrolled and any collisions between them can not be avoided, as they are both unable to maneuver.

This priority scheme is intuitive and simple to implement, yet can be usefull in a wide range of applications. One such example is ATC, where the use of priorities has shown beneficial results [13]. Other applications can include heterogeneous mobile robots executing tasks of different priorities.

B. Limited sensing

We use the dimensional obstacle function $\hat{g}_{ij}$ as defined in previous NF approaches:

$$\hat{g}_{ij} = \hat{g}_{ji} = ||\boldsymbol{q}_i - \boldsymbol{q}_j||^2 - r_{ij}^2 \tag{2}$$

where $r_{ij} \triangleq r_i + r_j$. By the above definition, $\hat{g}_{ij}$ is zero when agents i, j touch, i.e. when $||\boldsymbol{q}_j - \boldsymbol{q}_i|| = r_{ij}$, and increases as they move away from each other.

Since each agent can sense or communicate with other agents that are within a maximum sensing range R_s away, i.e. when $||\boldsymbol{q}_j - \boldsymbol{q}_i|| \leq R_s$, we nondimensionalise the obstacle function $\hat{g}_{ij}$ between agents i, j into g_{ij}:

$$g_{ij} = \begin{cases} \frac{L(\hat{g}_{ij})}{R_s^2 - r_{ij}^2}, & ||\boldsymbol{q}_i - \boldsymbol{q}_j|| \leq R_s \\ 1, & ||\boldsymbol{q}_i - \boldsymbol{q}_j|| > R_s \end{cases} \tag{3}$$

where the shaping function $L(x)$ is:

$$L(x) = x^3 - 3x^2 + 3x \tag{4}$$

The following properties hold for $L(x)$:

$$L(0) = 0 \tag{5a}$$
$$L(1) = 1 \tag{5b}$$
$$L'(x) > 0 \ \forall x \in [0, 1) \tag{5c}$$
$$L'(1) = L''(1) = 0 \tag{5d}$$

The dimensionless obstacle function g_{ij} is zero when i,j are in a collision, i.e. $||\boldsymbol{q}_i - \boldsymbol{q}_j|| = r_{ij}$ and increases up to 1 at the boundary of the sensing area, i.e. when $||\boldsymbol{q}_i - \boldsymbol{q}_j|| = R_s$. Outside the sensing range of agent i, it is constant and equal to 1. Using the above properties of $L(x)$ it can be verified that g_{ij} is $\mathcal{C}^2$ in the interior of the free space, i.e. away from collisions, where $\hat{g}_{ij} \in (0, +\infty)$. Thus, the potential Φ_i is also $\mathcal{C}^2$, as required for it to be a Navigation Function [6]. Function $g_{ij} = g_{ij}\left(||\boldsymbol{q}_i - \boldsymbol{q}_j||\right)$ is plotted in Figure 1. Since g_{ij} is constantly 1 when $||\boldsymbol{q}_i - \boldsymbol{q}_j|| \geq R_s$, each agent i is only affected by other agents $j \in T_i$ that are up to R_s away.

We propose the use of the following form of G_i:

$$G_i = \prod_{j \in T_i} g_{ij} \tag{6}$$

The priority classes defined in III-A are used here to allow an agent i to ignore agent j when $c_i < c_j$, while agent j has to manoeuvre around i. Essentially, this construction of G_i requires only the knowledge about those agents in T_i that are within the sensing range:

$$G_i = \prod_{j \in \tilde{T}_i} g_{ij} \tag{7}$$

where $\tilde{T}_i = \{j \in T_i \,|\, ||\boldsymbol{q}_i - \boldsymbol{q}_j|| < R_s\}$ is the "close threat" set, i.e. the subset of T_i within the sensing range R_s.

Similarly to g_{ij}, we modify β_i to limit the effect of the workspace boundary in a zone of width R_s near it. The dimensional workspace boundary function $\hat{\beta}_i$ is:

$$\hat{\beta}_i = (R_w - r_i)^2 - ||\boldsymbol{q}_i||^2$$

The dimensionless function β_i is derived similarly to g_{ij}:

$$\beta_i = \begin{cases} \frac{L(\hat{\beta}_i)}{(R_w - r_i)^2 - (R_w - R_s)^2}, & ||\boldsymbol{q}_i|| \geq R_w - R_s \\ 1, & ||\boldsymbol{q}_i|| < R_w - R_s \end{cases} \tag{8}$$

Thus, as Figure 2 shows, β_i is zero when agent i touches the workspace, i.e. $||\boldsymbol{q}_i|| = R_w - r_i$, and varies in a $\mathcal{C}^2$ fashion to exactly 1 when agent i is at a distance R_s or more away from the workspace boundary, i.e. $||\boldsymbol{q}_i|| \leq R_w - R_s$.

C. Potential Construction

For the target function γ_i we use the following form:

$$\gamma_i = \frac{||\boldsymbol{q}_i - \boldsymbol{q}_{di}||^2}{R_w^2} \tag{9}$$

Since the largest distance between any points in the spherical workspace of radius R_w is $2R_w$, γ_i is equal to or lower than 4 for any combination of $\boldsymbol{q}_i$, $\boldsymbol{q}_{di}$.

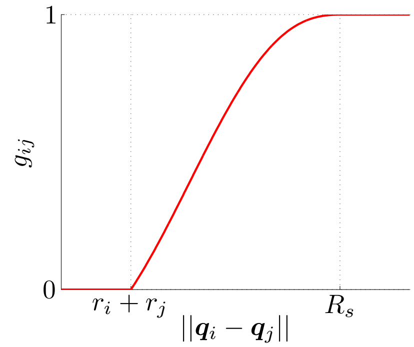

Fig. 1. Obstacle function g_{ij} wrt distance $||\boldsymbol{q}_i - \boldsymbol{q}_j||$ between agents i, j

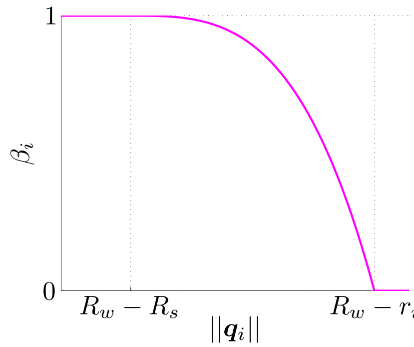

Fig. 2. Workspace boundary function β_i wrt $||\boldsymbol{q}_i||$

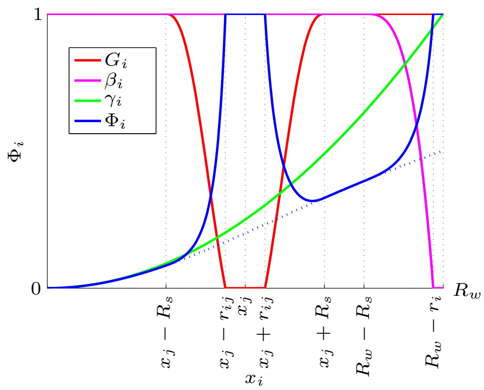

Fig. 3. Obstacle function G_i, workspace boundary function β_i, target function γ_i and the resulting potential Φ_i for $x_i \in [0, Rw]$, $y_i = 0$.

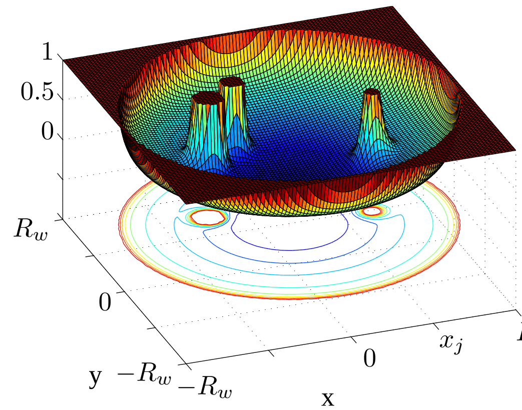

Fig. 4. NF field in a workspace with 3 obstacles and local sensing.

The cooperation function f_i is used here as in [7]:

$$f_i(G_i) = \begin{cases} a_0 + \sum_{l=1}^{3} a_l G_i^l, & G_i \leq X \\ 0, & G_i > X \end{cases} \tag{10}$$

where $a_0 = Y$, $a_1 = 0$, $a_2 = \frac{-3Y}{X^2}$, $a_3 = \frac{2Y}{X^3}$ and X, Y are positive parameters. The aim of f_i is to become non-zero in proximity situations, forcing an agent that has already reached its destination to temporarily move away from it to facilitate the convergence of near-by agents. X sets a threshold for G_i, such that values of G_i lower than X activate the cooperation function f_i. Parameter Y defines the maximum value of f_i, which is attained when $G_i = 0$.

The final result of using the above defined G_i, β_i and γ_i in (1) for a setup with 3 obstacles is shown in Figures 3 and 4. The target $\boldsymbol{q}_{di}$ is set in the center of the workspace and 3 obstacles are included. Figure 4 presents the potential field in the workspace, while Figure 3 shows the values of G_i, β_i, γ_i and Φ_i along the positive x axis, that crosses through the center of one of the obstacles that is placed between the target and the workspace boundary. In this example the cooperation function f_i is not activated, i.e. $f_i = 0$ everywhere. As Figure 3 shows, G_i and β_i become less than 1 only within the sensing range R_s of the obstacle and workspace boundary, respectively. The dotted blue line represents the value of Φ_i for $G_i = \beta_i = 1$ everywhere, i.e. without the effect of any obstacles or the workspace boundary. As expected, this coincides with the actual Φ_i outside the sensing range of the obstacle and the workspace boundary.

D. Proof of correctness

It has been shown in [6] that NF properties are invariant under diffeomorphisms. We will exploit this property here to ensure that the potential (1) using the definitions of γ_i, f_i, G_i and β_i given above maintains the navigation properties and can provide almost global convergence to the destination. The shaping function $L(x)$ is smooth and strictly increasing in the set $[0, 1)$ (see (5c)). Thus, $g_{ij} = g_{ij}(\hat{g}_{ij}) : [0, R_s^2 - r_{ij}^2) \to [0, 1)$ is a diffeomorphism when agent j is inside the sensing area of agent i. Generalising, $g_{ij}(\hat{g}_{ij})$ is a diffeomorphism whenever $j \in \hat{T}_i$. Thus, the critical points of Φ_i inside the sensing range of agent i are the same with those of the potential in [11], which does not uses the non-dimensionalisation and shaping function. Moreover, agents outside $\tilde{T}_i$ do not affect Φ_i, and $G_i = \prod_{\tilde{T}_i} g_{ij}$ becomes equal to $\prod_{\tilde{T}_i \setminus j} g_{ij}$ in a $\mathcal{C}^2$ way as $||\boldsymbol{q}_i - \boldsymbol{q}_{di}||^2$ reaches R_s. Thus, agents outside the sensing range can be ignored and do not affect the navigation properties of Φ_i.

Consequently, the potential presented here is a NF and as such it provides almost global navigation and collision avoidance for all k higher than a finite lower bound. Moreover, since only nearby agents and obstacles affect the potential, the number of g_{ij} that contribute to Φ_i at any given time is significantly reduced, especially in scenarios with many agents. Simulation experience with NFs indicates that the minimum value of the exponent k required to eliminate local minima increases with the number of contributing obstacles. Thus, the exponent k needed for the potential presented here is in most cases lower than the one required in [11].

IV. Completely Decentralised Navigation

We can exploit the navigation properties of the above potential field Φ_i to avoid collisions and guide all agents to their destinations. In fact, any controller that can maintain a decreasing rate for each potential Φ_i, i.e. $\dot{\Phi}_i < 0$ can be employed in combination with the potential field presented previously to stabilise the agents to their targets, while avoiding collisions. Such a control law has been presented in [8] for unicycle aircraft-like agents in 3D space.

Deriving a controller for planar unicycles using the same principles can be achieved by neglecting the vertical velocity in [8]. The resulting control scheme employs the projection of the gradient $\nabla_i \Phi_i = \begin{bmatrix} \Phi_{ix} & \Phi_{iy} \end{bmatrix}^\top$ on the agent's i longitudinal (heading) direction:

$$P_i = \boldsymbol{J}_i^\top \cdot \nabla_i \Phi_i \tag{11}$$

where $\boldsymbol{J}_i = \begin{bmatrix} \cos(\phi_i) & \sin(\phi_i) \end{bmatrix}^\top$ The sign of P_i, $s_i = \mathrm{sgn}(P_i)$, determines the direction of motion, where:

$$\mathrm{sgn}(x) \triangleq \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0. \end{cases}$$

Moreover, we use the partial derivative $\frac{\partial \Phi_i}{\partial t}$, which sums the effect of all but the i^{th} agents' motion on Φ_i:

$$\frac{\partial \Phi_i}{\partial t} = \sum_{j \neq i} \nabla_j \Phi_i^\top \cdot \boldsymbol{J}_j u_j$$

where $\nabla_j \Phi_i = \frac{\partial \Phi_i}{\partial \boldsymbol{q}_j}$ is the gradient of Φ_i with respect to $\boldsymbol{q}_j$.

The proposed control law for the linear velocity u_i is:

$$u_i = \begin{cases} -s_i U_i, & \frac{\partial \Phi_i}{\partial t} \leq U_i \left(|P_i| - \varepsilon\right) \\ -s_i \frac{U_i \varepsilon + \frac{\partial \Phi_i}{\partial t}}{|P_i|}, & \frac{\partial \Phi_i}{\partial t} > U_i \left(|P_i| - \varepsilon\right) \end{cases} \quad (12)$$

where $\varepsilon > 0$ is a small constant and U_i the nominal velocity:

$$U_i = \begin{cases} u_{di}, & ||\boldsymbol{q}_i - \boldsymbol{q}_{di}|| > d_i \\ \frac{||\boldsymbol{q}_i - \boldsymbol{q}_{di}||}{d_i} \cdot u_{di}, & ||\boldsymbol{q}_i - \boldsymbol{q}_{di}|| \leq d_i. \end{cases} \quad (13)$$

which follows identically a reference signal u_{di} away from the target $\boldsymbol{q}_{di}$ and is continuously reduced to 0 inside a ball of radius d_i around $\boldsymbol{q}_{di}$. Velocity u_i is designed to follow the nominal velocity U_i when stability and conflict avoidance is ensured and only diverge from it temporarily when absolutely required.The angular velocity ω_i is used to align the agent with the integral lines of the potential field:

$$\omega_i = \begin{cases} 0, & M_i \geq \varepsilon_\phi \\ \Omega_i \cdot \left(1 - \frac{M_i}{\varepsilon_\phi}\right), & 0 < M_i < \varepsilon_\phi \\ \Omega_i, & M_i \leq 0, \end{cases} \quad (14)$$

where:

$$M_i \triangleq \dot{\phi}_{\boldsymbol{nh}_i} \left(\phi_i - \phi_{\boldsymbol{nh}_i}\right),$$
$$\Omega_i \triangleq -k_\phi \left(\phi_i - \phi_{\boldsymbol{nh}_i}\right) + \dot{\phi}_{\boldsymbol{nh}_i}.$$

The *nonholonomic* heading angle $\phi_{\mathbf{nh}i}$ represents the heading of $\text{sgn}(p_i)\nabla_i \Phi_i$:

$$\phi_{\mathbf{nh}i} \triangleq atan2\left(\text{sgn}\left(p_i\right)\Phi_{iy}, s_i \Phi_{ix}\right), \quad (15)$$

where the function atan2 is:

$$\text{atan2}(y, x) \triangleq \arg\left(x, y\right), \quad (x, y) \in \mathbb{C},$$

and $p_i = \mathbf{J}_{di}^\top \cdot (\boldsymbol{n}_{i1} - \boldsymbol{n}_{i1d})$ is the position vector with respect to the destination, projected on the longitudinal axis of the desired orientation. Consequently, $\text{sgn}(p_i)$ is equal to 1 in front of the target configuration and -1 behind it. Finally, ε_ϕ is a small positive constant and k_ϕ a positive gain.

A. Stability and convergence analysis

Since the stability analysis in [8] does not rely on the specific Navigation Function used, one can follow the same line of thought here to prove that the above control scheme ensures a decreasing rate for all Φ_i over time:

$$\dot{\Phi}_i \leq -u_{di}\varepsilon \quad (16)$$

Thus, convergence and collision avoidance are guaranteed. It should be noted though that here we have not included in (1) a nonholonomic obstacle H_{nh} to render it Dipolar [14]. Thus, the integral lines of the resulting potential field approach the destination with arbitrary orientation, allowing only its position to be stabilised.

The use of the priority scheme described in III-A means that collisions between any two agents i, j are avoided when at least one of them has non-zero priority, $max(c_i, c_j) > 0$, i.e. one of them is able to maneuver. This holds because by construction a NF is transverse on the boundary of collisions with other agents or obstacles. One can easily show similarly to [8] that the above control scheme ensures that $\nabla_i \Phi_i u_i \leq 0$ holds always, i.e. all agents move towards the direction that decreases their potential. Thus, when there is at least one-way sensing between any two neighboring agents, at least one of the agents moves away from the other and collisions between them are avoided. Off course, when both agents are uncontrolled, $c_i = c_j = 0$, no collision avoidance can be performed between them. Thus, the proposed control scheme combined with the priority rules in section III-A ensures that all collisions between two controlled agents or a controlled and an uncontrolled one or an obstacle are avoided.

The time required for each agent i to reach an area of radius d_i around its target $\boldsymbol{q}_{di}$ can be bounded by considering that the control law presented above ensures that $\dot{\Phi}_i \leq -U_i\varepsilon$. We will assume here that each agent starts further away from its target than d_i, i.e. that each agent has to move a significant amount to reach its destination. Since $U_i = u_{di}$ whenever $||\boldsymbol{q}_i - \boldsymbol{q}_{di}|| \geq d_i$, we deduce that $\dot{\Phi}_i \leq -u_{di}\varepsilon$, i.e. the decreasing rate of Φ_i is smaller than a finite negative quantity $-u_{di}\varepsilon$ outside a circle of radius d_i around $\boldsymbol{q}_{di}$. We denote as $\Phi_{i0} = \Phi_i(t = 0)$ the initial value of Φ_i and Φ_{id} the value of Φ_i when agent i reaches for the first time at a distance d_i away from $\boldsymbol{q}_{di}$. Thus, the total change in the potential value from the initial position of each agent i up to when it reaches the circle of radius d_i around its target $\boldsymbol{q}_{di}$ is $\Delta\Phi_i = \Phi_{id} - \Phi_{i0}$. Since in a real scenario there should be no collision at the initial conditions, $\Phi_{i0} < 1 \forall i \in \{1, \ldots, N\}$. Moreover, as $d_i > 0$, $\Phi_{id} > 0$. Consequently:

$$\Delta\Phi_i = \Phi_{id} - \Phi_{i0} > -1 \quad (17)$$

Denoting as t_{id} the first time instant that agent i reaches a distance d_i from $\boldsymbol{q}_{di}$ and assuming a constant u_{di} over time, by (16) we have:

$$\Delta\Phi_i = \int_0^{t_{id}} \dot{\Phi}_i dt \leq -u_{di}\varepsilon t_{id} \quad (18)$$

Combining (17) and (18) we derive:

$$-1 < \Delta\Phi_i \leq -u_{di}\varepsilon t_{id} \Longrightarrow \quad (19)$$
$$t_{id} < \frac{1}{u_{di}\varepsilon} \quad (20)$$

Thus, the time t_{id} required for agent i to reach a distance d_i from $\boldsymbol{q}_{di}$ is always less that $t_{imax} = \frac{1}{u_{di}\varepsilon}$. It should be noted though that agent i may enter the circle of radius d_i around $\boldsymbol{q}_{di}$ but exit again if forced to do so by other agents or obstacles. However, t_{imax} gives a reasonable limit for the time required to reach the vicinity of the target $\boldsymbol{q}_{di}$. Finally, the maximum time needed for all agents to converge within distance d_i from their targets is:

$$t_{MAX} = \max_i \left(t_{imax}\right) \quad (21)$$

V. Simulation Results

In order to demonstrate the effectiveness of the proposed control approach we present simulation results for two possible application scenarios. The first one can be described as a *stream crossing* situation: a stream of agents move in parallel, while another agent starting from one side of the stream is assigned a destination in the other side, thus being forced to cross it. Agents 1-4 are given higher priority, i.e. lower c_i, than agent 5, allowing them to move straight while the crossing agent 5 maneuvers around them.

The result of this scenario can be seen in Figure 5. Agents 1-4 do not maneuver at all, as their potential fields do not consider the intruding agent 5. On the other hand, agent 5 maneuvers around all other agents and finally reaches its destination without any collisions.

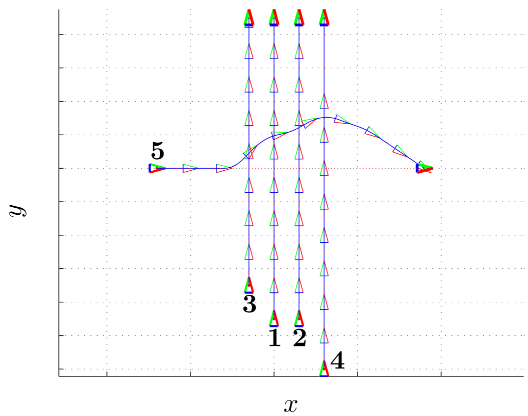

Fig. 5. High priority agents 1-4 move straight in a stream, while low priority agent 5 maneuvers around them to reach its destination.

For the second test case we used the same initial and final positions with inverted priorities. Thus, agents 1-4 now have low priority, while the high priority agent 5 crosses their paths. This scenario could resemble a situation in ATC where an aircraft in emergency condition assumes higher priority to facilitate its motion. The results are shown in Figure 6. Agents 1-4 are forced into large deviations from their straight line paths in order to avoid collisions with agent 5 and each other. Finally, all agents converge to their destinations.

VI. Conclusions

We have presented an algorithm for multi-agent navigation and collision avoidance employing a feedback control scheme. Our work here combines contributions from previous work in the NF methodology to derive a completely decentralised algorithm. Furthermore, we have extended the capabilities of our algorithm by integrating prioritisation and allowing for static and moving obstacles, as well as disabled agents. Guarantees for convergence and collision avoidance are given and the maximum convergence time is discussed. Future work in this area is directed towards the use of a scheme for assigning and updating agents' priorities according to the needs of applications from the fields of Robotics and Air Traffic Control (ATC), and studying its interaction with the algorithm presented here.

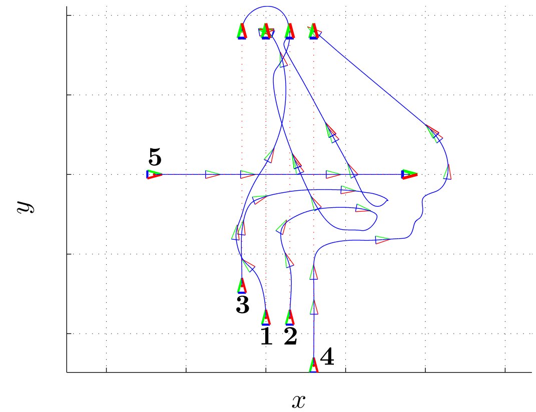

Fig. 6. High priority agent 5 crosses the path of lower priority agents 1-4, forcing them to maneuver around it and each other to reach their destinations.

VII. Acknowledgements

The authors of this paper want to acknowledge the contribution of the European Commission through project iFLY.

References

[1] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–126, 2005.

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," 2002.

[3] S. Lindemann and S. L. Valle, "Smoothly blending vector fields for global robot navigation," in *44th IEEE Conference on Decision and Control and European Control Conference*, Omnipress, 2005.

[4] N. Ayanian and V. Kumar, "Decentralized feedback controllers for multi-agent teams in environments with obstacles," in *IEEE Int. Conf. on Robotics and Automation*, pp. 1936–1941, May 2008.

[5] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[6] D. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Advances in Applied Mathematics*, vol. 11, no. 4, pp. 412–442, 1990.

[7] D. V. Dimarogonas and K. J. Kyriakopoulos, "Decentralized navigation functions for multiple robotic agents with limited sensing capabilities," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 3, pp. 411–433, 2007.

[8] G. Roussos and K. J. Kyriakopoulos, "Decentralised navigation and collision avoidance for aircraft in 3d space," *2010 American Control Conference, Baltimore, USA*, 2010. to appear.

[9] S. Resmerita, M. Heymann, and G. Meyer, "A framework for conflict resolution in air traffic management," in *42^{nd} Conference on Decision and Control*, (USA), pp. 2035–2040, 2003.

[10] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, "Multi-vehicle path planning in dynamically changing environments," in *2009 IEEE International Conference on Robotics and Automation*, (Piscataway, NJ, USA), pp. 2148–2153, IEEE Press, 2009.

[11] D. V. Dimarogonas, K. J. Kyriakopoulos, and D. Theodorakatos, "Totally distributed motion control of sphere world multi-agent systems using decentralized navigation functions," *2006 IEEE International Conference on Robotics and Automation*, pp. 2430–2435, 2006.

[12] G. Lionis, X. Papageorgiou, and K. J. Kyriakopoulos, "Locally computable navigation functions for sphere worlds," *2007 IEEE International Conference on Robotics and Automation*, pp. 1998–2003, 2007.

[13] R. Irvine, "Comparison of pairwise priority-based resolution schemes through fast-time simulation," *8^{th} Innovative Research Workshop & Exhibition, INO 2009, Eurocontrol Experimental Center*, 2009.

[14] H. G. Tanner, S. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, 2003.