2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Decentralized Stabilization and Collision Avoidance of Multiple Air Vehicles with Limited Sensing Capabilities

Dimos V. Dimarogonas and Kostas J. Kyriakopoulos
Control Systems Laboratory,Mechanical Eng. Dept.
ddimar,kkyria@mail.ntua.gr

Abstract— **Motivated by the needs of distributed Air Traffic Management, we extend the decentralized navigation function methodology, established in previous work for navigation of multiple holonomic agents with global sensing capabilities to the case of local sensing capabilities. Each agent plans its actions without knowing (i) the destinations of the others and (ii) the positions of agents outside its sensing neighborhood. A nonsmooth decentralized navigation function is used for each agent. The collision avoidance and global convergence properties are verified through simulations.**

I. INTRODUCTION

Navigation of mobile agents has been an area of significant interest in robotics and control communities. Most efforts have focused on the case of a single agent navigating in an environment with obstacles [10]. Recently, decentralized navigation for multiple agents has gained increasing attention. The motivation comes from many application domains, among which decentralized conflict resolution in air traffic management(ATM) has gained increasing attention in the past few years.

Today's air traffic systems remain to a large extent widely centralized [15]. A central authority, namely the Air Traffic Controllers (ATC), is responsible for issuing instructions to conflict-bound aircraft . To resolve conflicts they ask aircraft to climb/descend or vector them away from the path in the flight plan and then back on to it. Flight plans are completely pre-defined and aircraft fly along fixed corridors and at specified altitude. Only minor deviations from the original flight plan are permitted on line. Autonomous decision-making by aircraft is allowed under the Traffic Alert Collision Avoidance System (TCAS)[11], which issues advisories in order to avoid potential collisions, yet is used only in extreme situations.

On the other hand, the demand for air transportation is constantly increasing and threatens to exceed the capacity of the current centralized ATM structure. The number of passengers using air traffic is predicted to increase up to 120 % in the next ten years [14], and studies in [15] indicate that, with the current ATM structure, a major accident could occur every 7 to 10 days by the year 2015. Moreover, recent technological advances in avionics such as satellite positioning systems (the Global Positioning System (GPS)), inter-communication systems (the Automatic Dependent Surveillance-Broadcast (ADSB)-although its current use in air traffic is rather limited), and powerful on-board computers, are used in terms of the current centralized ATM system and provide an improvement on it, but not a radical change in the air traffic community.

These facts have resulted in the growing will of the air traffic world for new architectures, which employ these new technological innovations towards a more user-centered system. The purpose is to supply pilots with more decisional freedom and to reduce the authority and influence of the ATCs. The ultimate purpose of these efforts is *free-flight*, a concept in which aircraft will be allowed to plan their en-route trajectories and resolve any conflicts with other aircraft in a distributed and cooperative manner. In this case, the ATC will play the role of a passive observer.

The need for development of decentralized conflict resolution algorithms is therefore evident. The level of decentralization depends on the perception an agent has on the other agents' actions and the knowledge of their objectives. In our previous work ([5],[13]) the decentralization factor lied in the fact that each agent had knowledge only of its own desired destination, but not of the desired destinations of the others. Clearly, this is a suitable model for a futuristic distributed ATM system, where each aircraft will have knowledge of the actions and positions of the other aircraft at each time instant, but not of their destinations.

Nevertheless, in practice, the sensing capabilities of each agent are limited. Consequently, each agent can not have knowledge of the positions and/or velocities of every agent in the workspace but only of the agents within its sensing zone at each time instant. The interpretation of the sensing zone of an agent that we use in this paper is a circle of constant radius around its center of mass. The same problem has been dealt in [1],[8] under a game theoretic perspective, while the concept of a sensing zone has also been used in [6]. In this paper we use the navigation function method established in [9], [12],[5] and solve the problem in a closed loop fashion.

The rest of the paper is organized as follows: in section 2 the system definition, the corresponding assumptions and the problem statement are presented. In section 3 we redefine the decentralized navigation functions introduced in [5] to cope with the limited sensing capabilities of the agents. The stability analysis of the system is contained in section 4. Simulation results are presented in section 5 while section 6 summarizes the conclusions and indicates our current research.

II. System and Problem Definition

Consider a system of N agents operating in the same workspace $W \subset \mathcal{R}^2$. Each agent i occupies a disc: $R = \{q \in \mathcal{R}^2 : \| q - q_i \| \leq r_i\}$ in the workspace where $q_i \in \mathcal{R}^2$ is the center of the disc and r_i is the radius of the agent. The configuration space is spanned by $q = [q_1, \ldots, q_N]^T$. The motion of each agent are described by the single integrator:

$$\dot{q}_i = u_i, i \in \mathcal{N} = [1, \ldots, N] \tag{1}$$

The desired destinations of the agents are denoted by the index d: $q_d = [q_{d1}, \ldots, q_{dN}]^T$. We make the following assumptions:

- Each agent has only knowledge of the position agents located in a cyclic neighborhood of specific radius d_C at each time instant, where $d_C > max_{i,j \in \mathcal{N}}(r_i + r_j)$.
- Each agent has knowledge only of its own desired destination but not of the others.
- Each agent knows the exact number of agents in the workspace.
- We consider spherical agents.
- The workspace is bounded and spherical.

Figure 1 shows a three-agent conflict situation. The multi

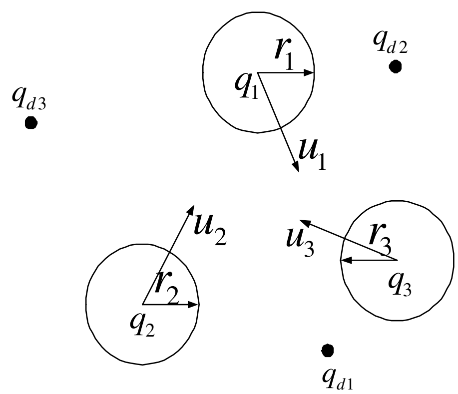

Fig. 1. A conflict scenario with three agents.

agent navigation problem treated in this paper can be stated as follows: "*derive a set of control laws (one for each agent) that drives a team of agents from any initial configuration to a desired goal configuration avoiding, at the same time, collisions. Each agent has no knowledge of the others desired destinations and has only local knowledge of their positions at each instant*".

III. Decentralized Navigation Functions for Agents with Limited Sensing Capabilities

In this section, we review the decentralized navigation function method used in [5] for the case of multiple holonomic agents and modify it in order to cope with the limited sensing capabilities specification. Consider a system of *n* agents operating in the same workspace $W \subset R^2$. Each agent i occupies a disk: $R = \{q \in R^2 : \|q - q_i\| \leq r_i\}$ in the workspace where $q_i \in R^2$ is the center of the disk and r_i is the radius of the agent. The dynamics of each agent are given by $\dot{q}_i = u_i$ and the configuration space is spanned by $q = [q_1, \ldots, q_n]^T$. The proposed control law for each agent is given by

$$u_i = -K_i \cdot \frac{\partial \varphi_i}{\partial q_i} \tag{2}$$

where K_i is a positive gain and the *decentralized navigation function* φ_i is defined as

$$\varphi_i = \frac{\gamma_{di} + f_i}{((\gamma_{di} + f_i)^k + G_i)^{1/k}} \tag{3}$$

The term $\gamma_{di} = \|q_i - q_{di}\|^2$ in the potential function is the squared metric of the agent's i configuration from its desired destination q_{di}. The exponent k is a scalar positive parameter. The function G_i expresses all possible collisions of agent i with the others, while f_i guarantees that φ_i attains positive values whenever collisions with respect to i tend to occur even when i has already reached its destination.

A. Construction of the G_i function

We review now the construction of the "collision" function G_i for each agent i. In [5], the decentralization feature of the whole scheme lied in the fact that each agent didn't have knowledge of the desired destinations of the rest of the team. On the other hand, each one had global knowledge of the positions of the others at each time instant. This is far from realistic in real world applications. The "Proximity Function" between agents i and j in [5] was given by

$$\beta_{ij} = \|q_i - q_j\|^2 - (r_i + r_j)^2$$

In this work we take the limited sensing capabilities of each agent into account. Each agent has only local knowledge of the positions of the others at each time instant. Specifically, it only knows the position of agents which are in a cyclic neighborhood of specific radius d_C around its center. Therefore the Proximity Function between two agents has to be redefined in this case. We propose the following nonsmooth function:

$$\beta_{ij} = \begin{cases} \|q_i - q_j\|^2 - (r_i + r_j)^2, \text{for } \|q_i - q_j\| \leq d_C \\ d_C^2 - (r_i + r_j)^2, \text{for } \|q_i - q_j\| > d_C \end{cases} \tag{4}$$

Figure 2 represents a nonsmooth proximity function. Consider now the situation in figure 3. There are 5 agents and we proceed to define the function G_R for agent R.

Definition 3.1: A *relation* with respect to agent R is every possible collision scheme that can occur in a multiple agents scene with respect R.

Definition 3.2: A *binary relation* with respect to agent R is a relation between agent R and another.

Definition 3.3: The *relation level* in the number of binary relations in a relation.

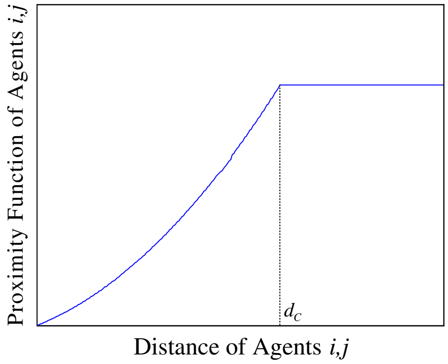

Fig. 2. The function β_{ij} for $r_i + r_j = 1, d_C = 4$.

We denote by $(R_j)_l$ the jth relation of level-l with respect to agent R. With this terminology in hand, the collision scheme of figure (3a) is a level-1 relation (one binary relation) and that of figure (3b) is a level-3 relation (three binary relations), always with respect to the specific agent R. We use the notation

$$(R_j)_l = \{\{R, A\}, \{R, B\}, \{R, C\}, \ldots\}$$

to denote the set of binary relations in a relation with respect to agent R, where $\{A, B, C, ...\}$ the set of agents that participate in the specific relation. For example, in figure 3b:

$$(R_1)_3 = \{\{R, O_1\}, \{R, O_2\}, \{R, O_3\}\}$$

where we have set arbitrarily $j = 1$.

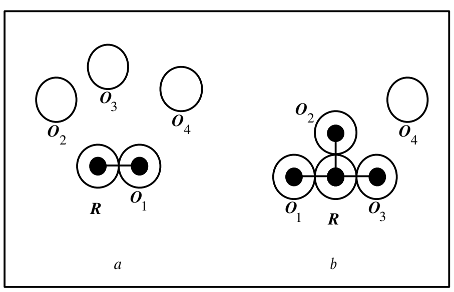

Fig. 3. Part *a* represents a level-1 relation and part *b* a level-3 relation wrt agent R.

The complementary set $(R_j^C)_l$ of relation j is the set that contains all the relations of the same level apart from the specific relation j. For example in figure 3b:

$$(R_1^C)_3 = \{(R_2)_3, (R_3)_3, (R_4)_3\}$$

where

$$(R_2)_3 = \{\{R, O_1\}, \{R, O_2\}, \{R, O_4\}\}$$

$$(R_3)_3 = \{\{R, O_1\}, \{R, O_3\}, \{R, O_4\}\}$$

$$(R_4)_3 = \{\{R, O_2\}, \{R, O_3\}, \{R, O_4\}\}$$

A "Relation Proximity Function" (RPF) provides a measure of the distance between agent i and the other agents involved in the relation. Each relation has its own RPF. Let R_k denote the k^{th} relation of level l. The RPF of this relation is given by:

$$(b_{R_k})_l = \sum_{j \in (R_k)_l} \beta_{\{R,j\}} \tag{5}$$

where the notation $j \in (R_k)_l$ is used to denote the agents that participate in the specific relation of agent R.

For example, in the relation of figure (2b) we have

$$(b_{R_1})_3 = \sum_{m \in (R_1)_3} \beta_{\{R,m\}} = \beta_{\{R,O_1\}} + \beta_{\{R,O_2\}} + \beta_{\{R,O_3\}}$$

A "Relation Verification Function" (RVF) is defined by:

$$(g_{R_k})_l = (b_{R_k})_l + \frac{\lambda (b_{R_k})_l}{(b_{R_k})_l + (B_{R_k^C})_l^{1/h}} \tag{6}$$

where λ, h are positive scalars and

$$(B_{R_k^C})_l = \prod_{m \in (R_k^C)_l} (b_m)_l$$

where as previously defined, $(R_k^C)_l$ is the complementary set of relations of level-l, i.e. all the other relations with respect to agent i that have the same number of binary relations with the relation R_k. Continuing with the previous example we could compute, for instance,

$$\left(B_{R_1^C}\right)_3 = (b_{R_2})_3 \cdot (b_{R_3})_3 \cdot (b_{R_4})_3$$

which refers to level-3 relations of agent R.

Using the simplified notation $(b_{R_k})_l = b_i, (B_{R_k^C})_l = \tilde{b}_i$, the RVF can be written as $g_i = b_i + \frac{\lambda b_i}{b_i + \tilde{b}_i^{1/h}}$ It is obvious that for the highest level $l = n-1$ only one relation is possible so that $(R_k^C)_{n-1} = \emptyset$ and $(g_{R_k})_l = (b_{R_k})_l$ for $l = n-1$. The basic property that we demand from RVF is that it assumes the value of zero if a relation holds, while no other relations of the same or other levels hold. In other words it should indicate which of all possible relations holds. We have he following limits of RVF (using the simplified notation): (a) $\lim\limits_{b_i \to 0} \lim\limits_{\tilde{b}_i \to 0} g_i\left(b_i, \tilde{b}_i\right) = \lambda$ (b) $\lim\limits_{\substack{b_i \to 0 \\ \tilde{b}_i \neq 0}} g_i\left(b_i, \tilde{b}_i\right) = 0$. These limits guarantee that RVF will behave in the way we want it to, as an indicator of a specific collision.

The function G_i is now defined as

$$G_i = \prod_{l=1}^{n_L^i} \prod_{j=1}^{n_{R_l}^i} (g_{R_j})_l \tag{7}$$

where n_L^i the number of levels and $n_{R_l}^i$ the number of relations in level-l with respect to agent i.

B. Construction of the f_i function

The key difference of the decentralized method with respect to the centralized case is that the control law of each agent ignores the destinations of the others. By using $\varphi_i = \frac{\gamma_{di}}{\left((\gamma_{di})^k + G_i\right)^{1/k}}$ as a navigation function for agent i, there is no potential for i to cooperate in a possible collision scheme when its initial condition coincides with its final destination. In order to overcome this limitation,we add a function f_i to γ_i so that the cost function φ_i attains positive values in proximity situations even when i has already reached its destination. We define the function f_i by:

$$f_i(G_i) = \begin{cases} a_0 + \sum\limits_{j=1}^{3} a_j G_i^j, \ G_i \leq X \\ 0, \ G_i > X \end{cases}$$

where $X, Y = f_i(0) > 0$ are positive parameters the role of which will be made clear in the following. The parameters a_j are evaluated so that f_i is maximized when $G_i \to 0$ and minimized when $G_i = X$. We also require that f_i is continuously differentiable at X. Therefore we have:

$$a_0 = Y, a_1 = 0, a_2 = \frac{-3Y}{X^2}, a_3 = \frac{2Y}{X^3}$$

The parameter X serves as a sensing parameter that activates the f_i function whenever possible collisions are bound to occur. The only requirement we have for X is that it must be small enough whenever the system has reached its equilibrium, i.e. when everyone has reached its destination. In mathematical terms:

$$X < G_i\left(q_{d1}, \ldots, q_{dN}\right) \ \forall i$$

That's the minimum requirement we have regarding knowledge of the destinations of the team. Intuitively, the destinations should be far enough from one another.

A key feature of navigation functions and in particular, Decentralized Navigation Functions, is that their gradient motion is repulsive with respect to the boundary of the free space. The free space for each agent is defined as the subset of W which is free of collisions with the other agents. Hence collision avoidance is reassured. For further information regarding terminology the reader is referred to [5], [3].

IV. Stability Analysis

Following [5], we use the sum of the separate decentralized navigation functions $\varphi = \sum \varphi_i$ as a candidate Lyapunov function for the whole system. Specifically, the following holds:

Theorem 4.1: The system is asymptotically stabilized to $q_d = [q_{d1}, \ldots, q_{dN}]^T$ up to a set of initial conditions of measure zero if the parameters k,h assume values bigger than a finite lower bound.

We immediately note that the result of this theorem is existential rather than computational. We show that finite k, h that renders the system almost everywhere asymptotically stable *exist*, but we do not provide an analytical expression for this lower bound. However, practical values of k, h will be provided in the simulation section.

Proof Sketch of Theorem 4.1: We only provide a sketch and guidlines of the proof. The complete proof can be downloaded from [3]. The fact that the nonsmooth modification of the Proximity Function (eq.(4)) does not affect the stability result can be justified using tools from nonsmmoth analysis [2],[16]. The proof is omitted here due to lack s space. The interested reader is referred to [4] for more details. The Proximity function between agents i and j is given by:

$$\beta_{ij}(q) = \|q_i - q_j\|^2 - (r_i + r_j)^2 = q^T D_{ij} q - (r_i + r_j)^2$$

where the $2N \times 2N$ matrix D_{ij} is defined in [12]. We can also write $b_r^i = q^T P_r^i q - \sum\limits_{j \in P_r} (r_i + r_j)^2$,where $P_r^i = \sum\limits_{j \in P_r} D_{ij}$, and P_r denotes the set of binary relations in relation r. It can easily be seen that $\nabla b_r^i = 2P_r^i q, \nabla^2 b_r^i = 2P_r^i$. We also use the following notation for the r-th relation wrt agent i:

$$g_r^i = b_r^i + \frac{\lambda b_r^i}{b_r^i + \left(\tilde{b}_r^i\right)^{1/h}}, \tilde{b}_r^i = \prod_{\substack{s \in S_r \\ s \neq r}} b_s^i,$$

$$\nabla \tilde{b}_r^i = \sum_{\substack{s \in S_r \\ s \neq r}} \underbrace{\prod_{\substack{t \in S_r \\ t \neq s, r}} b_i^t}_{\tilde{b}_{s,r}^i} \cdot 2P_s^i q$$

where S_r denotes the set of relations in the same level with relation r. An easy calculation shows that

$$\nabla g_r^i = \ldots = 2\left[d_r^i P_r^i - w_r^i \tilde{P}_r^i\right] q \triangleq Q_r^i q, \tilde{P}_r^i \triangleq \sum_{\substack{s \in S_r \\ s \neq r}} \tilde{b}_{s,r}^i P_s^i$$

where $d_r^i = 1 + (1 - \frac{b_r^i}{b_r^i + (\tilde{b}_r^i)^{1/h}}) \frac{\lambda}{b_r^i + (\tilde{b}_r^i)^{1/h}}$, $w_r^i = \frac{\lambda b_r^i (\tilde{b}_r^i)^{\frac{1}{h} - 1}}{h(b_r^i + (\tilde{b}_r^i)^{1/h})^2}$. The gradient of the G_i function is given by:

$$G_i = \prod_{r=1}^{N_i} g_r^i \Rightarrow \nabla G_i = \sum_{r=1}^{N_i} \underbrace{\prod_{\substack{l=1 \\ l \neq r}}^{N_i} g_l^i}_{\tilde{g}_r^i} \nabla g_r^i = \sum_{r=1}^{N_i} \tilde{g}_r^i Q_r^i q \triangleq Q_i q$$

where N_i all the relations with respect to agent i. We define

$$\nabla G \triangleq \begin{bmatrix} \nabla G_1 \\ \vdots \\ \nabla G_N \end{bmatrix} = \begin{bmatrix} Q_1 \\ \vdots \\ Q_N \end{bmatrix} q \triangleq Qq$$

Remembering that $u_i = -K_i \frac{\partial \varphi_i}{\partial q_i}$ and that $\varphi_i = \frac{\gamma_{di} + f_i}{\left((\gamma_{di} + f_i)^k + G_i\right)^{1/k}}, f_i = \sum\limits_{j=0}^{3} a_i G_i^j$ the closed loop dynamics of the system are given by:

$$\dot{q} = \begin{bmatrix} -K_1 A_1^{-(1+1/k)} \left\{ G_1 \frac{\partial \gamma_{d1}}{\partial q_1} + \sigma_1 \frac{\partial G_1}{\partial q_1} \right\} \\ \vdots \\ -K_N A_N^{-(1+1/k)} \left\{ G_N \frac{\partial \gamma_{dN}}{\partial q_N} + \sigma_N \frac{\partial G_N}{\partial q_N} \right\} \end{bmatrix} = \dots$$
$$= -A_K G\left(\partial\gamma_d\right) - A_K \Sigma Q q$$

where $\sigma_i = G_i \sigma(G_i) - \frac{\gamma_{di}+f_i}{k}, \sigma(G_i) = \sum_{j=1}^{3} j a_j G_i^{j-1}, A_i = (\gamma_{di} + f_i)^k + G_i$ and the matrices

$$A_K \triangleq \underbrace{diag \begin{pmatrix} K_1 A_1^{-(1+1/k)}, K_1 A_1^{-(1+1/k)}, \dots \\ , K_N A_N^{-(1+1/k)}, K_N A_N^{-(1+1/k)} \end{pmatrix}}_{2N\times 2N}$$

$$G \triangleq \underbrace{diag\left(G_1, G_1, \dots, G_N, G_N\right)}_{2N\times 2N}, \left(\partial\gamma_d\right) = \begin{bmatrix} \frac{\partial \gamma_{d1}}{\partial q_1} \dots \frac{\partial \gamma_{dN}}{\partial q_N} \end{bmatrix}$$

$$\Sigma \triangleq \underbrace{\begin{bmatrix} \underbrace{\Sigma_1}_{2N\times 2N}, \dots, \underbrace{\Sigma_N}_{2N\times 2N} \end{bmatrix}}_{2N\times 2N^2}, \Sigma_i = diag \begin{pmatrix} 0, 0, \dots, \underbrace{\sigma_i, \sigma_i}_{2i-1, 2i} \\ , \dots, 0, 0 \end{pmatrix}$$

By using $\varphi = \sum_i \varphi_i$ as a candidate Lyapunov function we have $\varphi = \sum_i \varphi_i \Rightarrow \dot{\varphi} = \left(\sum_i \left(\nabla\varphi_i\right)^T\right) \dot{q}, \nabla\varphi_i = A_i^{-(1+1/k)} \left\{ G_i \nabla\gamma_{di} + \sigma_i \nabla G_i \right\}$ and after some trivial calculation

$$\sum_i \left(\nabla\varphi_i\right)^T = \dots = \left(\partial\gamma_d\right)^T A_G + q^T Q^T A_\Sigma$$

where $A_G = \underbrace{diag \begin{pmatrix} G_1 A_1^{-(1+1/k)}, G_1 A_1^{-(1+1/k)}, \dots, \\ G_N A_N^{-(1+1/k)}, G_N A_N^{-(1+1/k)} \end{pmatrix}}_{2N\times 2N}$

and

$$A_\Sigma = \underbrace{\begin{bmatrix} \underbrace{A_{\Sigma_1}}_{2N\times 2N} \\ \vdots \\ \underbrace{A_{\Sigma_N}}_{2N\times 2N} \end{bmatrix}}_{2N^2\times 2N}, A_{\Sigma_i} = \underbrace{diag \begin{pmatrix} A_i^{-(1+1/k)} \sigma_i, \dots, \\ A_i^{-(1+1/k)} \sigma_i \end{pmatrix}}_{2N\times 2N}$$

The derivative of the candidate Lyapunov function is calculated as

$$\dot{\varphi} = \left(\sum_i \left(\nabla\varphi_i\right)^T\right) \cdot \dot{q} = \dots$$
$$= -\begin{bmatrix} \left(\partial\gamma_d\right)^T & q^T \end{bmatrix} \underbrace{\begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}}_{M} \begin{bmatrix} \partial\gamma_d \\ q \end{bmatrix}$$

where $M_1 = A_G A_K G, M_2 = A_G A_K \Sigma Q, M_3 = Q^T A_\Sigma A_K G, M_4 = Q^T A_\Sigma A_K \Sigma Q$.

In [3], we make use of matrix analysis, and in particular, of Gersgorin's theorem ([7]) to prove that the matrix M is positive definite up to a measure zero set of initial conditions that lead to saddle points. This establishes the asymptotic stability of the proposed scheme.

V. SIMULATIONS

To demonstrate the navigation properties of our decentralized approach, we present a simulation of four holonomic agents that have to navigate from an initial to a final configuration, avoiding collision with each other. Each agent has no knowledge of the positions of agents outside its sensing zone, which is the big circle around its center of mass in Fig.3, Pic.1. In this picture A-i,T-i denote the initial condition and desired destination of agent i respectively. The chosen configurations constitute non-trivial setups since the straight-line paths connecting initial and final positions of each agent are obstructed by other agents. The following have been chosen for the simulation of figure 3:

Initial Conditions:

$$q_1(0) = \begin{bmatrix} -.1732 & -.1 \end{bmatrix}^T, q_2(0) = \begin{bmatrix} .1732 & -.1 \end{bmatrix}^T,$$
$$q_3(0) = \begin{bmatrix} 0 & .2 \end{bmatrix}^T, q_4(0) = \begin{bmatrix} 0 & -.2 \end{bmatrix}^T$$

Final Conditions:

$$q_{d1} = \begin{bmatrix} .1732 & .1 \end{bmatrix}^T, q_{d2} = \begin{bmatrix} -.1732 & .1 \end{bmatrix}^T,$$
$$q_{d3} = \begin{bmatrix} 0 & -.1 \end{bmatrix}^T, q_{d4} = \begin{bmatrix} 0 & .25 \end{bmatrix}^T$$

Parameters:

$$k = 110, r_1 = r_2 = r_3 = r_4 = .05, d_C = .11$$
$$\lambda = 1, h = 5, X = .001, Y = .01$$

Pictures 1-6 of Figure 3 show the evolution of the team configuration within a horizon of 6000 time units. One can observe that the collision avoidance as well as destination convergence properties are fulfilled.

In the next simulation (Fig.4) the sensing zone of the red agent A2 is shown in all the screenshots. The following have been chosen for the simulation:

Initial Conditions:

$$q_1(0) = \begin{bmatrix} -.1732 & -.1 \end{bmatrix}^T, q_2(0) = \begin{bmatrix} .1732 & -.1 \end{bmatrix}^T,$$
$$q_3(0) = \begin{bmatrix} 0 & .2 \end{bmatrix}^T, q_4(0) = \begin{bmatrix} 0 & -.2 \end{bmatrix}^T$$

Final Conditions:

$$q_{d1} = \begin{bmatrix} .15 & .05 \end{bmatrix}^T, q_{d2} = \begin{bmatrix} -.1732 & .2 \end{bmatrix}^T,$$
$$q_{d3} = \begin{bmatrix} 0 & -.1 \end{bmatrix}^T, q_{d4} = \begin{bmatrix} 0 & .25 \end{bmatrix}^T$$

Parameters:

$$k = 100, r_1 = r_2 = r_3 = r_4 = .03, d_C = .08$$
$$\lambda = 1, h = 5, X = .001, Y = .01$$

The collision avoidance and destination requirements are met in this case as well. We point out that since the sensing zone of the red agent is always empty,i.e. it does not participate in a conflict situation, its trajectory is the straight line between its initial and final destination. This is due to the fact that the sensing parameter d_C is small in this case.

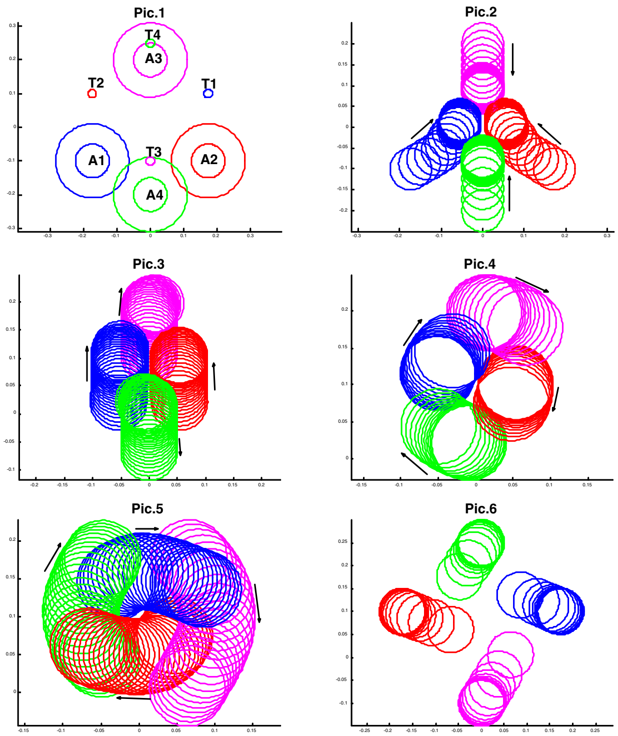

Fig. 4. Simulation A

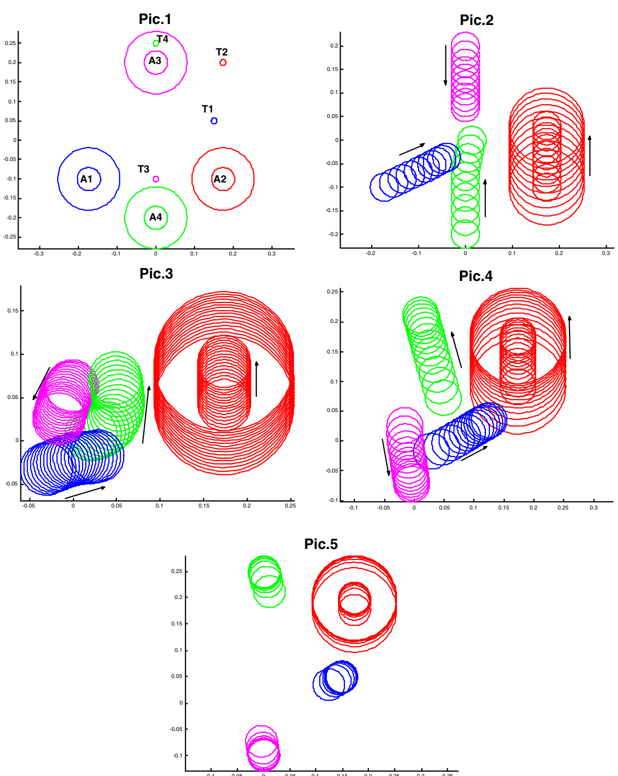

Fig. 5. Simulation B

VI. Conclusions

In this paper we extended the decentralized navigation method to the case of multiple holonomic agents with limited sensing capabilities. We proposed a nonsmooth extension of the navigation function of [5] and proved system convergence using tools from nonsmooth stability analysis. The effectiveness of the methodology is verified through computer simulations.

Current research includes applying this method to the case of distributed nonholonomic agents [13] as well as introducing new definitions of the sensing zone of an agent. Extensions of this method to 3-dimensional dynamics are also under investigation.

VII. Acknowledgements

The authors want to acknowledge the contribution of the European Commission through contracts HYBRIDGE(IST-2001- 32460) and I-SWARM (IST- 2004-507006).

References

[1] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):221–232, 2000.

[2] F. Clarke. *Optimization and Nonsmooth Analysis*. Addison - Wesley, 1983.

[3] D. V. Dimarogonas, S. G. Loizou, K.J. Kyriakopoulos, and M. M. Zavlanos. Decentralized feedback stabilization and collision avoidance of multiple agents. Tech. report, NTUA, http://users.ntua.gr/ddimar/TechRep0401.pdf, 2004.

[4] D.V. Dimarogonas and K.J. Kyriakopoulos. Decentralized navigation functions for multiple agents with limited sensing capabilities. *submitted for journal publication*, 2005.

[5] D.V. Dimarogonas, M.M. Zavlanos, S.G. Loizou, and K.J. Kyriakopoulos. Decentralized motion control of multiple holonomic agents under input constraints. *42nd IEEE Conference on Decision and Control*, 2003, submitted for Journal Publication.

[6] M. Egerstedt and X. Hu. A hybrid control approach to action coordination for mobile robots. *Automatica*, 38:125–130, 2002.

[7] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1996.

[8] G. Inalhan, D.M. Stipanovic, and C.J. Tomlin. Decentralized optimization, with application to multiple aircraft coordination. *41st IEEE Conf. Decision and Control*, 2002.

[9] D. E. Koditschek and E. Rimon. Robot navigation functions on manifolds with boundary. *Advances Appl. Math.*, 11:412–442, 1990.

[10] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[11] C. Livadas, J. Lygeros, and N.A. Lynch. High-level modeling and analysis of the traffic alert and collision avoidance system (tcas). *Proceedings of the IEEE*, 88(7):949–970, 2000.

[12] S. G. Loizou and K. J. Kyriakopoulos. Closed loop navigation for multiple holonomic vehicles. *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2861–2866, 2002.

[13] S.G. Loizou, D.V. Dimarogonas, and K.J. Kyriakopoulos. Decentralized feedback stabilization of multiple nonholonomic agents. *2004 IEEE International Conference on Robotics and Automation*, 2004.

[14] Department of Transport UK Government. Air traffic forecasts for the united kingdom 1997. *Technical Repor*, 1997.

[15] T.S. Perry. In search of the future of air traffic control. *IEEE Spectrum*, 34(8):18–34, 1997.

[16] D. Shevitz and B. Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Trans. on Automatic Control*, 49(9):1910–1914, 1994.