Totally distributed motion control of sphere world multi-agent systems using Decentralized Navigation Functions

Dimos V. Dimarogonas, Kostas J. Kyriakopoulos and Dimitris Theodorakatos

***Abstract*— A distributed feedback control architecture that guarantees collision avoidance and destination convergence for multiple sphere world holonomic agents is presented. The well established tool of Decentralized Navigation Functions is redefined to cope with the communication restrictions of the system. Each agent plans its actions without knowing the destinations of the others and the positions of those agents lying outside its sensing neighborhood. The stability properties of the closed loop system are checked via Lyapunov stability techniques for hybrid systems. The collision avoidance and goal convergence properties are verified through simulations. The key advantage of the proposed algorithm with respect to the previous ones is the significant decrease of computational load and its applicability to large scale groups.**

I. INTRODUCTION

Navigation of multiple agents is a field that has recently gained increasing attention in the robotics community, due to the need for autonomous control of more than one mobile robotic agents in the same workspace. While most approaches in the past had focused on centralized planning, specific real-world applications have lead researchers throughout the globe to turn their attention to decentralized concepts. The basic motivation of our work comes from two application domains: (i) decentralized conflict resolution in air traffic management ([10]) and (ii) the field of micro robotics ([17],[10]), where a team of autonomous micro robots must cooperate to achieve manipulation precision in the sub micron level.

The reduced computational complexity and increased robustness with respect to agent failures makes decentralized approaches more appealing compared to the centralized ones. There have been many different approaches to the decentralized motion planning problem. Open loop approaches use game theoretic and optimal control theory to solve the problem taking the constraints of vehicle motion into account; see for example [2], [11], [23], [24]. On the other hand, closed loop approaches use tools from classical Lyapunov theory and graph theory to design control laws and achieve the convergence of the distributed system to a desired configuration both in the concept of cooperative ([8], [13]) and formation control ([1], [9], [14], [21],[22]).

Closed loop strategies are apparently preferable to open loop ones, mainly because they provide robustness with respect to modelling uncertainties and agent failures and guaranteed convergence to the desired configurations. However, a common point of most work in this area is devoted to the case of point agents. Although this allows for variable degree of decentralization, it is far from realistic in real world applications, even in the field of microrobotics, where the non-zero volume of each robot cannot be disregarded due to the fact that the surrounding objects are of comparable size. Another example is conflict resolution in Air Traffic Management, where two aircraft are not allowed to approach each other closer than a specific "alert" distance. The construction of closed loop methods for decentralized non-point multi-agent systems is both evident and appealing.

A closed loop approach for single robot navigation was proposed by Koditschek and Rimon [12] in their seminal work. This navigation functions' framework had all the sought qualities but could only handle single, point-sized, robot navigation. This framework was extended to multiple sphere world agents in a recent series of papers. In [15] this method was successfully extended to take into account the volume of each robot in a centralized multi-agent scheme, while a decentralized version of this work has been presented in [25],[7],[6] for multiple holonomic agents with global sensing capabilities. In these papers, 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. In [22], Tanner and Kumar considered the problem for point agents, taking the limited sensing capabilities of each agent into account. Limited sensing issues for sphere world agents have been considered by the authors in [4].

The level of decentralization in [4] lied in the fact that each agent had only local knowledge of the positions of the other agents at each time instant. However, the computational load required for the realization of the algorithm was restricted by the fact that each agent should have knowledge about the exact number of agents in the group. In this paper, we propose an improved algorithm with respect to the one in [4], which allows each agent to neglect any knowledge about the number of agents in the group and hence, reduces the computational load significantly with respect to the aforementioned paper. The convergence analysis is performed using the extension of LaSalle's Invariance Principle for Hybrid Systems. The key drawback of the method proposed in this paper wrt the algorithms of [4], [6], is that global convergence cannot be guaranteed in every case. This issue is discussed in detail in the stability analysis section.

The rest of the paper is organized as follows: section II presents the multi-agent system in hand and defines the problem addressed in this paper. In section III the concept of

The authors are with the Control Systems Lab, Department of Mechanical Engineering, National Technical University of Athens, 9 Heroon Polytechniou Street, Zografou 15780, Greece ddimar,kkyria@mail.ntua.gr

decentralized navigation functions, introduced in [7],[25],[6] to cope with navigation of multiple holonomic agents with global sensing capabilities, is reviewed and appropriately redefined in order to cope with the restrictions of the situation in hand. The multi-agent system is modelled as a hybrid system and convergence analysis based on LaSalle's Invariance Principle for Hybrid Systems is performed in section IV. Section V contains a nontrivial computer simulation based on the proposed algorithm while section VI summarizes the results and indicates some relevant future directions of 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: $R_i = \{q \in \mathcal{R}^2 : \| q - q_i \| \leq r_i\}$ 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 $q = [q_1, \ldots, q_N]^T$. Figure 1 shows a five-agent conflict situation. In the case of holonomic agents, the motion of each agent is 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 respectively denoted by the index d: $q_d = [q_{d1}, \ldots, q_{dN}]^T$. We make the following assumptions:

1) Each agent i has knowledge of the position of only those agents located in a cyclic neighborhood of specific radius d_C at each time instant, where $d_C > max_{i,j \in \mathcal{N}}(r_i + r_j)$, so that it is guaranteed to be larger than the maximum sum of two agents radii. The disc $T_i = \{q : \|q - q_i\| \leq d_C\}$ is called the *sensing zone* of agent i.
2) Each agent has knowledge only of its own desired destination q_{di} but not of the others $q_{dj}, j \neq i$.
3) Spherical agents are considered.
4) The workspace is bounded and spherical.

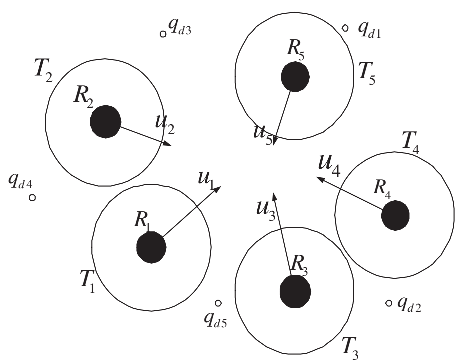

Fig. 1. A conflict scenario with five agents. Each agent i occupies a disc R_i(black discs) of radius r_i centered at q_i. Each agent's sensing zone T_i(white discs) is centered at q_i and has radius d_C.

The first two assumptions reveal the decentralized nature of this frame-work, as well as its specific limitations. Each agent needs to know the exact position only of agents found within its sensing zone at each time instant(ass.2). Furthermore, knowledge of the desired destinations of the other agents is unnecessary(ass. 1). In this paper, the *navigation functions* ([12],[25],[7],[4]) tool is redefined in order to cope with assumptions 1,2.

Assumptions 3,4 regarding the spherical shape of the agents and the workspace do not constrain the generality of this work since it has been proven that navigation properties are invariant under diffeomorphisms ([12]). Arbitrarily shaped agents diffeomorphic to spheres can be taken into account. Methods for constructing analytic diffeomorphisms are discussed in [20] for rigid body agents.

III. Decentralized Navigation Functions: a totally distributed perspective

A. Preliminaries

Navigation functions (NF's) are real valued maps realized through cost functions $\varphi(q)$, whose negated gradient field is attractive towards the goal configuration and repulsive with respect to obstacles [12]. It has been shown by Koditschek and Rimon that strict global navigation (i.e. the system $\dot{q} = u$ under a feedback control law of the form $u = -K\nabla\varphi$ admits a globally attracting equilibrium state) is not possible, and a smooth vector field on any sphere world with a unique attractor, must have at least as many saddles as obstacles [12].

A navigation function is defined as follows:

Definition 1: [12]: Let $F \subset R^{2N}$ be a compact connected analytic manifold with boundary. A map $\varphi : F \to [0, 1]$ is a navigation function if:(1) it is analytic on F, (2) it has only one minimum at $q_d \in int(F)$, (3) its Hessian at all critical points (zero gradient vector field) is full rank, and (4) $lim_{q \to \vartheta F}\varphi(q) = 1$.

In this definition, F represents the "free space" of robot movement, i.e. the subset of the workspace which is free of collisions.

B. Decentralized Navigation Functions

In previous work, the authors extended the Navigation Functions method for multiple robots. In [15], the navigation functions method has been extended to the case of multiple mobile robots with the use of centralized Multi-Robot navigation functions (MRNF's).

In the form of a centralized setup [15], where a central authority has knowledge of the current positions and desired destinations of all agents $i = 1, \ldots, N$, the sought control law $u = [u_1 \ldots u_N]$ is of the form: $u = -K\nabla\varphi(q)$ where K is a gain. The extension to the case when each agent has only partial knowledge has given birth to the concept of decentralized navigation functions (DNF's) in the series of papers [25], [7], [4],[6]. In particular in [4], decentralized navigation functions have been defined and used for navigation of multiple sphere world agents with local sensing capabilities. The level of decentralization was however limited due to the fact that each agent should have been aware of the exact number of agents in the whole workspace at each moment. On the other hand, a proof of guaranteed convergence based on nonsmooth analysis was provided.

The objective of this paper is to treat the framework of [4] in a totally distributed manner. Hence each agent does no longer need to know the exact number of agents in the workspace.

Following the procedure of [4] we consider the following class of decentralized navigation functions(*DNF's*):

$$\varphi_i(q) = \frac{\gamma_{di} + f_i}{\left((\gamma_{di} + f_i)^k + G_i\right)^{1/k}} \qquad (2)$$

where k is a positive scalar parameter, and $\gamma_{di} = \parallel q_i - q_{di} \parallel^2$, is the squared metric of the current agent's configuration q_i from its destination q_{di}. The definition of the function f_i will be given later. Function G_i has as arguments the coordinates of agent i and all the agents belonging to its sensing zone, i.e.

$$G_i = G_i(q_i, \hat{q}_i), \hat{q}_i \triangleq \{q_j | q_j \in T_i\}$$

and is used to encapsulate all possible collision schemes of agent i with the agents that are within T_i. Hence the notation $\hat{q}_i$ is used to denote the stack vector of the configurations of all agents belonging to T_i at each time instant.

C. The f_i function

Function f_i is defined in such a way to ensure that the repulsive potential vanishes when inter-agent distances are sufficiently large. This function was introduced in ([25]),[4]). In these papers, the function f_i is defined by:

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

where $X, Y = f_i(0) > 0$ are positive scalar constants. The parameters a_j are evaluated so that f_i is maximized when $G_i \to 0$ and minimized when $G_i = X$. It is also required that f_i is continuously differentiable at X. Therefore we have:

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

D. The G_i function

In this subsection, we review the construction of the G_i function from [4]. In that reference the function G_i had as arguments the coordinates of all agents. In this paper, we impose the restriction that the G_i function of each agent depends only on its own coordinates and the coordinates of the agents belonging to its sensing zone at each time instant. We note that the following paragraphs are a revision of the construction in [4], [7]. The reader is referred to those papers for a thorough analysis.

The "Proximity Function" between two agents i, j is given by

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

Consider now a situation similar to the one in figure 1 where we have five agents. For an agent "R", we proceed to define function G_R. We denote by O_1, O_2, O_3, O_4 the remaining four agents in this scenario. To encode all possible inter-agent proximity situations, the multi-agent team is associated with an (undirected) graph whose vertices are indexed by the team members. The following are discussed in more detail in [6], [25], [7] .

Definition 2: A *binary relation* with respect to an agent R is an edge between agent R and another agent.

Definition 3: A *relation* with respect to agent R is defined as a set of binary relations with respect to agent R.

Definition 4: The *relation level* is the number of binary relations in a relation with respect to agent R.

The notation $(R_j)_l$ is used to denote the jth relation of level-l with respect to agent R. The notation

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

is used 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. where we have set arbitrarily $j = 1$.

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. 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:

$$(b_{R_k})_l = \sum_{j \in (R_k)_l} \beta_{\{R,j\}} \qquad (4)$$

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

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

$$(g_{R_k})_l = (b_{R_k})_l + \frac{\lambda (b_{R_k})_l}{(b_{R_k})_l + (B_{R_k^C})_l^{1/h}} \qquad (5)$$

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.

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 $g_{R_k}(b_{R_k}, B_{R_k^C}) \equiv g_i(b_i, \tilde{b}_i)$):

1) $\lim\limits_{b_i \to 0} \lim\limits_{\tilde{b}_i \to 0} g_i\left(b_i, \tilde{b}_i\right) = \lambda$
2) $\lim\limits_{\substack{b_i \to 0 \\ \tilde{b}_i \neq 0}} g_i\left(b_i, \tilde{b}_i\right) = 0$

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

$$G_i = \prod_{l=1}^{n_L^i} \prod_{j=1}^{n_{R_l}^i} (g_{R_j})_l \tag{6}$$

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. Hence G_i is the product of the RVF's of all relations wrt i.

The construction of the G_i function is held in such a way to ensure that the gradient motion imposed on agent i under the control strategy that is defined in the sequel is repulsive with respect to the boundary of the free space. This guarantees collision avoidance. More details are found in [7].

IV. Control Design and Stability Analysis

The major difference of the framework in [4] and the current paper, is the fact that the DNF φ_i is discontinuous whenever agents enter or leave the sensing zone of other agents. Hence the stability analysis held in [4] is not valid in this case. We analyze the convergence properties of our framework using tools from hybrid systems literature.

A. Hybrid Automaton Modelling and Control Strategy

In order to capture the discontinuous behavior of the system in hand, we model the system as a deterministic hybrid system in which switches occur whenever an agent enter or leaves the sensing zone of another.

The *information pattern* $\mathcal{I}_i$ of agent i is defined as the set of agents in its sensing zone at each time instant, namely

$$\mathcal{I}_i = \{j : \|q_i - q_j\| \leq d_C\} \tag{7}$$

The whole scheme can be modelled as a (deterministic) switched system ([3],[19]). Each mode of the switched system is characterized by the *global information pattern* of the system, namely $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$. It is obvious that G_i can be rewritten as $G_i = G_i(\mathcal{I}_i)$ which shows that the G_i function for all i is discontinuous whenever I_i is updated, i.e. whenever an agent enters/leaves the sensing zone of i. Hence φ_i can be rewritten as:

$$\varphi_i(\mathcal{I}_i) = \frac{\gamma_{di} + f_i}{\left((\gamma_{di} + f_i)^k + G_i(\mathcal{I}_i)\right)^{1/k}} \tag{8}$$

The proposed feedback control strategy for agent i at each mode $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ of the switched system is defined as

$$u_i = -K_i \frac{\partial \varphi_i(\mathcal{I}_i)}{\partial q_i}, i \in N \tag{9}$$

where $K_i > 0$ a positive gain. Hence the continuous dynamics change discretely each time an agent enters/leaves the sensing zone of another.

Many stability analysis tools for switched systems have appeared in literature in the past decade; see [3],[16] to name a few. In this paper we make use of the invariance principle for hybrid systems of [16] to check the convergence of our control algorithms. A hybrid system is modelled by the following discontinuous differential equation:

$$\dot{x} = f_i(x), i \in \mathcal{M} = \{1, \ldots, M\} \tag{10}$$

where $x \in \mathcal{R}^n$ and $\mathcal{M}$ the finite set of nodes of the hybrid system. The *switching sequence* indexed by an initial state x_0 is defined as

$$S = x_0; (i_0, t_0), (i_1, t_1), \ldots, (i_j, t_j), \ldots \tag{11}$$

where the notation (i_j, t_j) means that the system in node i_j, i.e. $\dot{x} = f_{i_j}(x)$ in the time interval $t_j \leq t \leq t_{j+1}$. Denote by $\mathcal{T} = \{t_1, \ldots, t_j, \ldots\}$ the set of switching instants. Coping with the application in hand, we assume that the state of the system is continuous at the switching instants, namely $x(t_j) = x(t_j^+) \forall j$. We now present the invariance principle for hybrid systems we use in our approach:

Theorem 1: [16] Consider the hybrid system (10) and let Ω be a compact invariant set. Assume there exists a continuous function $V : \Omega \to \mathcal{R}$ such that

1) for all $x \in \Omega$, $t \in \mathcal{R}^+ \setminus \mathcal{T}$, V is continuously differentiable and $\dot{V} \leq 0$, and
2) for all $t \in \mathcal{T}$, $V(x(t^+)) \leq V(x(t))$

Define $E_1 = \{x \in \Omega \mid \exists t \in \mathcal{R}^+ \setminus \mathcal{T}$ s.t. $x(t) = x$ and $\dot{V}(x(t)) = 0\}$ and $E_2 = \{x \in \Omega \mid \exists t \in \mathcal{T}$ s.t. $x(t) = x$ and $V(x(t)) = V(x(t^+))\}$. Let L be the largest invariant subset of $E_1 \bigcup E_2$. Then every trajectory in Ω converges to L.

The inclusion of the subset E_2 involves the possibility of infinite switches in finite time,i.e. the inclusion of sliding motion in the switching surfaces. We proceed by applying Theorem 1 in the current framework.

B. Convergence Analysis

Let $\mathcal{I}_i^f = \{1, \ldots, N\} = \mathcal{N}$ denote the full information pattern for agent i, i.e. the case where all agents belong to the sensing zone of agent i. Let $\mathcal{I}^f = \left\{\mathcal{I}_1^f, \ldots, \mathcal{I}_N^f\right\}$ denote the full global information pattern. Consider the candidate Lyapunov function

$$V = \sum_i \varphi_i\left(I_i^f\right) \tag{12}$$

which is the sum of the Decentralized Navigation Functions of all agents corresponding to the full information pattern.

This function is continuous everywhere, since the full global information pattern remains constant throughout the multi-agent system evolution, and is continuously differentiable in between the switching instants since each $\varphi_i(\mathcal{I}_i)$ is continuously differentiable for any constant $\mathcal{I}_i$. A key property of the candidate Lyapunov function is that it is continuous at the switching instants, as pointed out previously, and hence the second condition of theorem 1 is fulfilled. In fact, if we denote by $\mathcal{T} = \{t_1, \ldots, t_j, \ldots\}$ the set of switching instants, i.e. the times at which an agent enters/leaves the sensing zone of another, the candidate Lyapunov function satisfies: $t \in \mathcal{T}$, $V(q(t^+)) = V(q(t))$.

The next proposition states that the candidate Lyapunov function also satisfies the first condition of theorem 1:

Proposition 2: The time-derivative of the candidate Lyapunov function is negative semi-definite in between switching instants, for all possible global information patterns,i.e. for all possible nodes of the hybrid system. In mathematical terms

$$\forall t \in \mathcal{R}^{+} \setminus \mathcal{T} : \dot{V}(q(t)) \leq 0$$

The proof of this proposition follows the exact same steps as the proof of Proposition 2 in [5] and is omitted here due to lack of space. In fact, following the procedure in [5] we can show that the largest invariant set contained in the set $E_1 = \{q \mid \exists t \in \mathcal{R}^{+} \setminus \mathcal{T} \text{ s.t. } q(t) = q \text{ and } \dot{V}(q(t)) = 0\}$ is simply $q = q_d$, i.e. the desired destination points of the agents. By virtue of Theorem 1 and Proposition 2, the following is derived directly:

Proposition 3: The trajectories of the system (1) under the control law (9) converge to the largest invariant set contained in the set

$$E = \{q_d\} \cup \{q | \exists i, j \in \mathcal{N}, i \neq j \text{s.t.} \|q_i - q_j\| = d_C\}$$

This proposition reveals the main drawback of the proposed approach. While the energy of the system remains bounded (i.e. the hybrid system is stable) agents might get "stack" in a point in the set $E_2 = \{q | \exists i, j \in \mathcal{N}, i \neq j \text{s.t.} \|q_i - q_j\| = d_C\}$. The invariance of this set cannot be checked easily due to the asymmetric properties of the control strategy. In essence, the control design cannot guarantee global stability. However, following the same line of thought as in [14],[18], we state that the events in which the system reaches blocking behavior on the set E_2 are very rare, so the algorithm can achieve convergence to the goal configuration in most cases. Guaranteed convergence is achieved whenever the switching stops. The following is an immediate corollary of Proposition 3:

Corollary 4: Suppose that the system reaches a configuration at a time instant t_0,in which the global information pattern remains constant $\forall t > t_0$. Then the trajectories of the system (1) under the control law (9) converge to the desired destination q_d.

V. Simulations

To demonstrate the navigation properties of our decentralized approach, we present a simulation of 32 holonomic agents that have to navigate from an initial to a final configuration, avoiding collisions with each other. Each agent has no knowledge of the positions of those agents lying outside its sensing zone. 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 2:

Initial Conditions:
30 agents form a 5x6 square. The 2 remaining agents are placed above and below the square.

Final Conditions:
The agents' final position form the initials of the Control Systems Laboratory: CSL

Parameters:

$$k = 110, r_1 = r_2 = r_3 = r_4 = .1, d_C = .25$$
$$\lambda = 1, h = 5, X = .0001, Y = .1$$

Pictures 1-6 of Figure 2 show the evolution of the team configuration within a horizon of 50000 time units. One can observe that the collision avoidance as well as destination convergence properties are fulfilled. This simulation is an example of the significant reduce of computational load with respect to the method of [4].

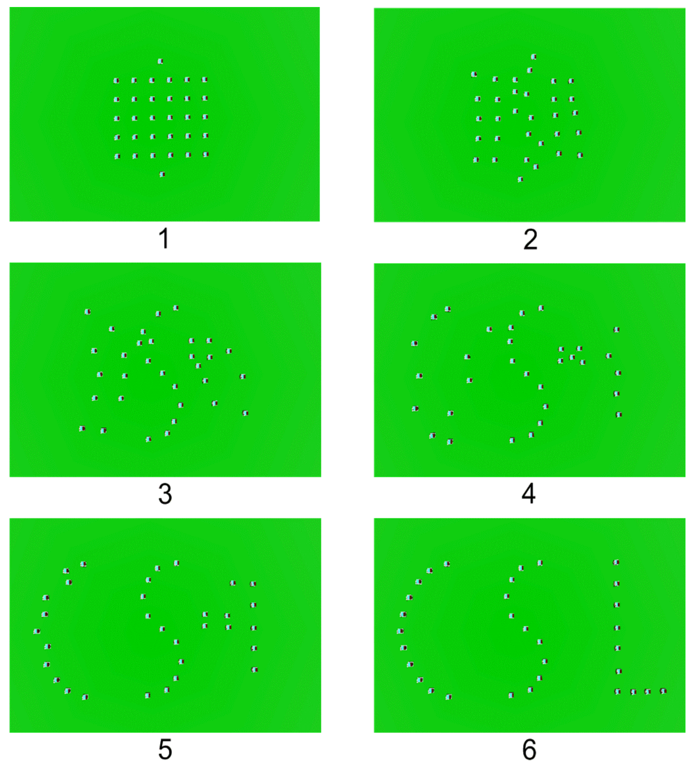

Fig. 2. Destination Convergence and Collision Avoidance for 32 Agents

A. Computational Load

The methodology's main advantage in respect to the methodology presented in ([7],[4]) is the reduction of the computational load. Therefore the methodology's real time implementation on large scale groups becomes possible. The computational load w.r.t. the agents' visibility radius is demonstrated with the following scenario; the agents are arranged on a parallelogram, where each agent's radius is R_a, and the distance between two agents' centers ,belonging in the same row or column, is $3R_a$ (Fig. 3). Using different visibility radii we calculate the time needed by a single controller for one step.

Table 1

Case	d_C	T(sec)	Visible agents
1	$3 * R_a$	0.077	4
2	$4 * R_a$	0.087	8
3	$5 * R_a$	0.782	12
4	$6 * R_a$	4.930	20

One can observe from the table above, that even though the number of visible agents increases linear w.r.t. to the visibility radius there is an exponential increase of the computational time and load. However these results are satisfactory since the time needed for the three first cases is small enough to allow application to large scale groups. Please note that the methodology in [4] forces each agent to take into account all the other agents. It is evident that our proposed algorithm is much more tractable and computationally less expensive.

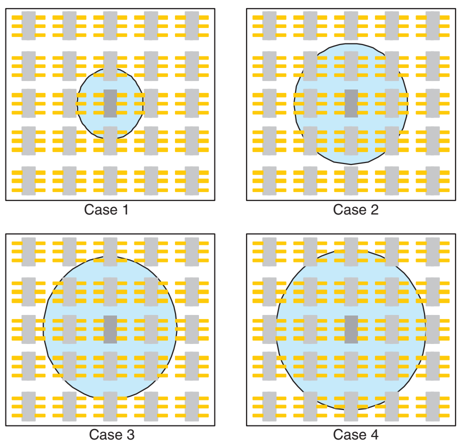

Fig. 3. Sensed Agents w.r.t. Agent Sensing Radius

VI. Conclusions

A distributed feedback control architecture that guarantees collision avoidance and destination convergence for multiple sphere world holonomic agents has been presented. The well established tool of Decentralized Navigation Functions has been redefined to cope with the communication restrictions of the system. Each agent plans its actions without knowing the destinations of the others and the positions of those agents lying outside its sensing neighborhood. The stability properties of the closed loop system have been checked via Lyapunov stability techniques for hybrid systems. The collision avoidance and goal convergence properties were verified through simulations. In most cases, the proposed scheme achieves navigation of the multi-agent team to the desired configuration. The key advantage of the proposed algorithm with respect to the previous ones is the significant decrease of computational load and its applicability to large scale groups.

Current research focuses on extending the proposed control scheme to the case of nonholonomic sphere world agents as well as comparing our algorithm wrt the method of [4] in an experimental setup. A formal analysis of the invariance properties of the set $E_2 = \{q|\exists i,j \in \mathcal{N}, i \neq j \text{s.t.} \|q_i - q_j\| = d_C\}$ is also currently pursued.

VII. Acknowledgements

The authors want to acknowledge the contribution of the European Commission through contracts MICRON(IST-2001-33567) and I-SWARM (IST-2004-507006).

References

[1] C. Belta and V. Kumar. Abstraction and control of groups of robots. *IEEE Transactions on Robotics*, 20(5):865–875, 2004.

[2] 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.

[3] M.S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):475–482, 1998.

[4] D. V. Dimarogonas and K. J. Kyriakopoulos. Decentralized stabilization and collision avoidance of multiple air vehicles with limited sensing capabilities. *2005 American Control Conference*, pages 4667–4772.

[5] 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.

[6] D. V. Dimarogonas, S. G. Loizou, K.J. Kyriakopoulos, and M. M. Zavlanos. A feedback stabilization and collision avoidance scheme for multiple independent non-point agents. *Automatica*, to appear, 2005.

[7] 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*, pages 3390–3395, 2003.

[8] J. Feddema and D. Schoenwald. Decentralized control of cooperative robotic vehicles. *IEEE Transactions on Robotics*, 18(5):852–864, 2002.

[9] V. Gazi and K.M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–696, 2003.

[10] Project HYBRIDGE. http://www.nlr.nl/public/hosted-sites/hybridge/.

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

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

[13] J.R. Lawton, R.W. Beard, and B.J. Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics and Automation*, 19(6):933–941, 2003.

[14] M. Lindhe, P. Ogren, and K. H. Johansson. Flocking with obstacle avoidance: A new distributed coordination algorithm based on voronoi partitions. *2005 IEEE International Conference on Robotics and Automation*, pages 1797–1782, 2005.

[15] 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.

[16] J. Lygeros, K.H. Johansson, S. Simic, J. Zhang, and S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, 2003.

[17] Project MICRON. http://wwwipr.ira.uka.de/ micron/.

[18] R. Olfati-Saber and R.M. Murray. Flocking with obstacle avoidance: Cooperation with limited communication in mobile networks. *42st IEEE Conf. Decision and Control*, pages 2022–2028, 2003.

[19] S. Pettersson and B. Lennartson. Stability and robustness for hybrid systems. *35th IEEE Conf. Decision and Control*, 1996.

[20] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. on Robotics and Automation*, 8(5):501–518, 1992.

[21] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents. *42st IEEE Conf. Decision and Control*, 2003.

[22] H.G. Tanner and A. Kumar. Formation stabilization of multiple agents using decentralized navigation functions. *Robotics: Science and Systems*, 2005.

[23] C. Tomlin, G.J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[24] J.P. Wangermann and R.F. Stengel. Optimization and coordination of multiagent systems using principled negotiation. *Jour.Guidance Control and Dynamics*, 22(1):43–50, 1999.

[25] M.M. Zavlanos and K.J. Kyriakopoulos. Decentralized motion control of multiple mobile agents. *11th Mediterranean Conference on Control and Automation*, 2003.