2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Near Optimal Swarm Deployment using Descriptor Functions

Marta Niccolini, Mario Innocenti, Senior Member, IEEE, and Lorenzo Pollini, Member, IEEE .

Abstract— **The article describes a novel approach for deployment of a swarm of heterogeneous autonomous vehicles. Each vehicle is treated as the agent of a network, which cooperates in order to cover a given area according to its own capabilities. A general framework is introduced, which aims at providing tools for solving a large class of coordination problems. The capabilities of each agent are modeled with Descriptor Functions; the sum of these functions constitutes the swarm's descriptor. The goal of the swarm is to match a desired descriptor, by minimizing an appropriate cost functional. A control law is proposed, which is capable of driving the agents towards the achievement of the goal. The existence of local minima and of a global minimum is discussed. Theoretical results on the existence of the global minimum are given. The Area Coverage problem is selected as a preliminary test for the algorithm. Simulation results show the effectiveness of the proposed approach.**

I. INTRODUCTION

Multi agent systems have received much attention from the scientific community due to their theoretical challenges and potential applications. Despite the growing research literature in the field, less attention has been paid to systems composed of heterogeneous agents. Agents' heterogeneity occurs primarily at two different levels: different agents may be associated to different tasks (mission level heterogeneity) and/or agents may have different capabilities in executing the same task (task level heterogeneity). Obviously the above mentioned differences could constitute a big advantage in performing more complex missions, and/or in performing the same task at different levels of quality. The other advantage is that the coordination among the subtasks of a mission can be improved if the ensemble of heterogeneous agents is treated as a single system (swarm). Furthermore, in such a situation, all the agents could be managed at a macro level (high level coordination).

In order to achieve adaptation and robustness of the system to changing environment conditions or evolution of the mission execution, agents capable of performing more than one task are preferable. Some kind of swarm self-organization can be in fact obtained by letting each agent switching between different tasks. Due to this self-organization process and, since some agents may execute more than one task at the same time, mission level heterogeneity may arise.

The scenario considered in this paper describes the swarm mission as a set of tasks, with each agent capable of performing one or more tasks, possibly at the same time, and with the agents executing the same task with different skills.

Under these assumptions, the challenge we are facing is to design a general framework such that many tasks can be described using the same mathematical tools and accomplished using the same or a minimally modified control law.

The paper is organized as follows: Section II reviews recent results of interest in this field and discusses analogies and differences with the proposed approach, Section III introduces and motivates the concept of Descriptor Functions, Section IV defines and proposes a solution to the general control problem addressed in this paper, finally Section V presents some simulation results for the selected case study.

Marta Niccolini is a Ph.D. student, with the Department of Electrical Systems and Automation, University of Pisa, Pisa, Italy 56126 (marta.niccolini@ing.unipi.it).

Mario Innocenti is a NRC Fellow with the Air Force Research Laboratory, Munitions Directorate, Eglin AFB, FL 32579, on sabbatical from the Department of Electrical Systems and Automation, University of Pisa, Pisa, Italy 56126 (minnoce@dsea.unipi.it).

Lorenzo Pollini is with the Department of Electrical Systems and Automation, University of Pisa, Pisa, Italy 56126 (lpollini@dsea.unipi.it)
The work of the second author was sponsored by the US National Academy of Sciences, under the Senior Research Associateship Program.

II. RELATED WORK

An approach similar to the one proposed in this paper can be found in [1]. The authors introduced the concept of effective coverage: "Given a sensor network and mission domain $\mathcal{D}$, how should the motion of each sensor agent be controlled such that the entire network surveys $\mathcal{D}$ by sensing each point in $\mathcal{D}$ by an amount of effective coverage equal to $\mathcal{C}^*$?" Agents were modeled by an instantaneous coverage function that describes how effective the agent senses each point of the environment. An error function was used to describe the difference between the desired and the attained effective coverage in the environment. Agents are driven towards the global minimum of a norm-like measure of the error function so that the attained coverage equals the desired one.

Improvements to this approach were presented in ([2]-[3]), where collision avoidance terms and penalization terms on the maximum acceptable inter-agent distance were added. This constraint was relaxed in [4] where a network of heterogeneous vehicles is used.

In [5] the non-stationary problem for the effective coverage was formulated. The attained coverage was updated with an information decay term, so that the agents would continuously keep gathering information. The problem of coverage over large scale domains, i.e. domains too large to be covered by a static sensor network, using instantaneous coverage function was improved in [6]. Instead of using a flocking strategy to achieve strong communication channels among sensors, another differential equation, which represents the

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

individual state of awareness, was fed back in the control law. The information sharing process would then reduce the amount of redundant coverage.

In [7] a new optimization algorithm for the coverage problem was formulated. A sensor is modeled with a function that describes the probability of event detection by the agent in the environment. The objective is encoded in a cost function that represents the joint detection probability of all the agents, weighted with a probability density function.

In [8] this algorithm was extended to teams of anisotropic sensors, i.e. sensor models with performance depending not only on the distance to the point of the environment but also on its orientation.

III. THE PROPOSED APPROACH

Three main definitions are essential to the understanding of the key points of the proposed methodology:

- *Agent:* entity that operates in the environment in order to achieve one or more objectives;
- *Task:* coordination in space of agents that are pursuing the same sub objective of the mission;
- *Mission:* coordination in space and time of the tasks.

The capability of an agent of executing a task, relative to a given point of the environment, can be quantified, in general, using some function of the relative distance from the specific point. From a qualitative point of view, agents can be defined as carriers of resources of one or more kind, and they are described by a set of functions, which we call agent Descriptor Functions (DFs). The number of DFs used to describe an agent capability equals the number of tasks that must be accomplished. If an agent is not capable of executing some of the tasks, the DFs relative to those tasks are set to zero for each point of the environment. Using this approach, mission level heterogeneity arises naturally. Task level heterogeneity can be included by assigning different DFs to the various agents. Since the tasks are defined in the operation space of the agents, a task can be specified as the request of resources of a given type for each point of the environment: the Desired Task Descriptor Function (TDF).

For each task, the sum of all the agents' DFs is a measure of the spatial distribution of resources pursuing the same objective, and can be used to describe its current state of execution: the Current TDF.

The difference between the Desired TDF and the Current TDF represents, for any point in space, the amount of resources needed, if positive, or in excess, if negative, for the accomplishment of each task: the Task Error Function (TEF).

The control objective can then be stated as the minimization, with respect to the position of the agents, of some cost function of the TEF. In this way, many different tasks can be formulated under the same framework by the use of different Desired TDF, while the architecture of the control law can be maintained for all the tasks.

It is well recognized that the awareness of the current state of execution of the tasks is a necessary condition for self-organization and adaptation. In our framework, some level of awareness is represented by the knowledge of the Current TDFs. This knowledge can then be used by the agents to accomplish the task they are executing (task level self-organization) and to switch between the tasks they are capable of executing (mission level self-organization). Furthermore, decentralized estimation (interpolation) techniques could be used to estimate (interpolate) the Current TDFs and the whole approach would then be decentralized.

This paper deals with the task level self-organization, and each agent is assumed to execute only one task. Furthermore, each agent knows exactly the Desired TDF and the Current TDF relative to the task it is executing.

A. Nomenclature and Definitions

This section presents the mathematical formulation of the proposed methodology.

The swarm is made up of N heterogeneous agents $V_i, i = 1, ...N$. The mission is composed of a set of M tasks. At each time instant, the set of agents executing the same task forms a team. The number of teams may be up to M, and each agent may be part of one or more teams simultaneously. The tasks that agent i is executing at time t are denoted by $T_i(t)$ and the teams are defined as

$$T^k(t) = \{V_i : k \in T_i(t)\}, \quad k = 1, ..., M \tag{1}$$

The space where agents operate is $Q \subset \Re^n$ and the position of agent i at time t is $p_i(t) \in Q$. Without loss of generality, Q is assumed to be closed and bounded. All agents are modeled by a single integrator kinematics with unity gain, i.e.

$$\dot{p}_i(t) = u_i(t), \quad u_i(t) \in \Re^n \tag{2}$$

Note that, even if an agent is capable of multi tasking, it optimizes its behavior for one task only.

1) Agent Descriptor Functions: M continuous functions are assigned to each agent

$$D_i^k(p_i, q) : \mathcal{P} \times Q \to \Re^+, \quad k = 1, ...M, \quad i = 1, ..., N \tag{3}$$

They describe the capability of the agent in executing the task k at location $q \in Q$. For a sensing agent, the Descriptor Function could be directly related to its sensing performance, i.e. to how much information the agent is capable of gathering at position $q \in Q$ as a function of its position. Examples can be found in the field of sensor networks, where it is common to model the sensors as functions, which are decreasing with the distance from the sensor location, [9]. For tasks, which are not sensing tasks, the DFs can be used to account for agents' presence. If the agent is not capable of executing a task then the DF relative to that task is set to zero for each $q \in Q$.

2) Current Task Descriptor Functions: The Current Descriptor Function of a task is defined as the sum of the DFs of the agents that are executing that task, i.e. $D^k(p, q) : \mathcal{P} \times Q \to \Re^+$:

$$D^k(p, q) = \sum_{V_i \in T^k(t)} D_i^k(p_i, q), \quad k = 1, ...M, \quad i = 1, ..., N \tag{4}$$

By using the terminology in [9], this is equivalent to the Sensor Intensity Field of a team of sensing agents. For agents that have no sensing capabilities it describes how they are deployed in the environment, i.e. it is a measure of their physical density.

3) Desired Task Descriptor Functions: The Desired Task Descriptor Function of a task specifies how the agents that are executing that task should be distributed. Specifically, it encodes the need of resources of a given type at each point $q \in Q$ of the environment and is formalized as:

$$D_*^k(q,t) : Q \times \Re^+ \longrightarrow \Re^+, \quad k = 1, ...M \tag{5}$$

4) Task Error Functions: The Task Error Function is defined as the difference between the Desired TDF and the Current TDF:

$$E^k(p,q,t) = D_*^k(q,t) - D^k(p,q), \quad k = 1, ...M \tag{6}$$

For each task, the error function measures the lack of resources if $E^k(p,q,t) > 0$, and the excess of resources if $E^k(p,q,t) < 0$.

IV. SWARM DEPLOYMENT AS AN OPTIMIZATION PROBLEM

The control objective here is to move the agents so that all the TEFs are minimized at every point of the environment. For each task, a cost function $J^k(p) : p \in Q \rightarrow \Re^+$, can be used as a measure of the TEF for each configuration of the agents. A possible cost function is:

$$J^k(p) = \int_Q f(E^k(p,q,t))\sigma(q)dq \tag{7}$$

where $f(\cdot)$ is a limited and continuous function of E^k that verifies $f(\tau) \geq 0, \ \forall \tau \in \Re$, and $\sigma(q) \geq 0, \ \forall q \in \Re$, is a weighting function which may drive the attention of the swarm towards specific areas of Q.

The optimization problem can then be formulated as the minimization of the cost function with respect to the position of the agents that participate actively to the task, i.e.

$$\hat{p}^k = \arg\min_{p \in T^k} J^k(p) \tag{8}$$

Note that, from Measure Theory, the cost function represents a weighted measure of the error over the whole environment.

A. Main results

Since only the self-organization at the task level is considered in this paper, the superscript relative to the task is dropped. In order to minimize the cost function consider the following gradiental control law:

$$u_i = \dot{p}_i = -\alpha \frac{\partial J(p)}{\partial p_i} \tag{9}$$

with $\alpha > 0$. For the above control law, the following holds:

Proposition 1. Under the control law

$$u_i = \dot{p}_i = -\alpha \frac{\partial J(p)}{\partial p_i} = \alpha \int_Q \frac{\partial D_i(p_i,q)}{\partial p_i} \frac{\partial f(E(p,q))}{\partial E(p,q)} \sigma(q)dq \tag{10}$$

each agent reaches a stable equilibrium position with a finite control effort and the cost function is nonincreasing in time.

Proof. Each component of the control law is the partial derivative of the cost function with respect to each component of the position vector of the agents. The gradient of the cost function is:

$$\nabla J(p) = \ [\tfrac{\partial J(p)}{\partial p_1} \quad \dots \quad \tfrac{\partial J(p)}{\partial p_N}] \tag{11}$$

The control law of the $i-th$ agent is then:

$$u_i = -\alpha \frac{\partial J(p)}{\partial p_i} = -\alpha \ [\tfrac{\partial J(p)}{\partial p_{i1}} \quad \dots \quad \tfrac{\partial J(p)}{\partial p_{in}}]^T \tag{12}$$

where p_{ij} is the $j-th$ component of the position vector of the agent i. Since $D_*(q,t)$ and $\sigma(q)$ do not depend on p, the $j-th$ component of the input vector of agent i is:

$$\begin{aligned} u_{ij} &= -\alpha \tfrac{\partial J(p)}{\partial p_{ij}} = -\alpha \int_Q \tfrac{\partial f(E(p,q,t))}{\partial p_{ij}} \sigma(q)dq \\ &= -\alpha \int_Q \tfrac{\partial E(p,q,t)}{\partial p_{ij}} \tfrac{\partial f(E(p,q,t))}{\partial E(p,q,t)} \sigma(q)dq \\ &= \alpha \int_Q \tfrac{\partial D_i(p_i,q)}{\partial p_{ij}} \tfrac{\partial f(E(p,q,t))}{\partial E(p,q,t)} \sigma(q)dq \end{aligned} \tag{13}$$

Using the chain rule the time derivative of $J(p)$ can be rewritten as:

$$\dot{J}(p) = \frac{dJ(p)}{dt} = \sum_{i=1}^{N} \sum_{j=1}^{n} \frac{\partial J(p)}{\partial p_{ij}} \dot{p}_{ij} \tag{14}$$

where N is the number of agents and n is the dimension of the space where the agents move. Since $\dot{p}_{ij} = u_{ij} = -\alpha \frac{\partial J(p)}{\partial p_{ij}}$ then

$$\dot{J}(p) = -\alpha \sum_{i=1}^{N} \sum_{j=1}^{n} \left(\frac{\partial J(p)}{\partial p_{ij}} \right)^2 \leq 0 \tag{15}$$

this proves that the cost function is nonincreasing in time. Moreover, since $f(\tau) \geq 0, \ \forall \tau \in \Re$, $J(p)$ has a lower limit, then:

$$\lim_{t \to \infty} J(p(t)) = \bar{J} \geq 0 \tag{16}$$

Since Eq.(16) holds, from Eq.(14) and Eq.(15) we have that $\lim_{t \to \infty} u_{ij} = 0$. Then each agent converges to a postion $\bar{p}$ for which $J(\bar{p}) = \bar{J}$. For each agent in $\bar{p}_i$ the following is verified:

$$u_{ij} = \alpha \int_Q \frac{\partial D_i(p_i,q)}{\partial p_{ij}} \frac{\partial f(E(p,q,t))}{\partial E(p,q,t)} \bigg|_{p=\bar{p}} \sigma(q)dq = 0 \tag{17}$$

$\forall i = 1, \dots, N, \ \forall j = 1, \dots, n$, and $\bar{p}$ is a stationary point for the system of agents under the control law in Eq.(10).

Finally, since $f(\cdot)$ is a limited and continuous function and the set Q is closed and bounded, the control law in Eq.(10) is finite.

■

Corollary 2. Under the control law

$$u_i = -\beta(t)\zeta(p,t) \tag{18}$$

with

$$sign\left(\zeta\left(p,t\right)\right) = sign\left(\left.\frac{\partial J(p)}{\partial p_i}\right|_{p=p(t)}\right) \quad (19)$$

and $\beta(t) \geq 0\ \forall t \geq 0$, each agent reaches a stable equilibrium position with a finite control law and the cost function is nonincreasing in time.

Proof. The proof is identical to that of Proposition 1, except that Eq.(15) is replaced by:

$$sign\left(\dot{J}(p)\right) = -\beta(t) sign\left(\sum_{i=1}^{N}\sum_{j=1}^{n}\frac{\partial J(p)}{\partial p_{ij}}\zeta\left(p,t\right)\right) \leq 0 \quad (20)$$

■

The class of possible functions ζ is very large and allows an additional degree of freedom control law design. Just as an example, β and ζ can be selected to produce a velocity command u_i which has a constant modulus and varies in direction only, at least when far from the goal; this approach may be particularly useful for vehicles like UAVs with speed constraints.

B. Local Minima and Selection of $f(\cdot)$

It is important at this point to make a few comments relative to the nature of the stationary points achieved by the proposed control law. Since the controller is based on a steepest descent optimization method, the system may end up in local minima. It is important to note that it is not possible to identify the global minimum and local minima only by evaluating the cost function. In fact, given a generic swarm and a generic Desired TDF, the global minimum attainable by any cost function is not known a priori. A similar issue arises with the control law derived in [1]. In that work, the minimum value of the cost function is known to be zero. Once an agent reaches a local minimum, it evaluates the cost function and, if not zero, it switches to a different control law that brings the system out from the local minimum. Clearly, this strategy cannot work if $\bar{J}$ is not known.

The behavior of the control law depends mainly on the derivative of the DF of the agents, on the Current TEF, and on the function $f(\cdot)$. Each point of the environment can be seen as generating a force on the agent with a direction that depends on the value of each component of the integrand vector at that point. The final direction of motion of the agent is the vector sum (integral) of all vectors generated by each point of the environment. When the vector sum gives the zero vector, the agent stops and Eq.(17) holds.

To better clarify how the technique works and when the system is more likely to reach a local minimum, let us consider a Gaussian agent DF in a $1D$ space. Moving in $1D$ space ($Q : q_1 \subseteq \Re$), the control is completely specified by the sign and magnitude of the velocity of the agent. The DF has a maximum value at the agent position, and decreases with the distance to the agent. The derivative of the Descriptor Function of agent i is positive for $q_1 > p_i$ and negative for $q_1 < p_i$.

In order to study the effect of selecting different functions $f(\cdot)$ consider the following two cases:

- Square of the error: $f(x) = x^2$;
- Integration of the square of positive errors only: $f(x) = \max\left(0, x\right)^2$.

In the first case, $f(x) = x^2$, the control law in Eq.(10), becomes:

$$u_i = 2\alpha \int_Q \frac{\partial D_i(p_i, q)}{\partial p_i} E(p, q) dq \quad (21)$$

From Eq.(21), with the considerations made on $\partial D_i(p_i, q)/\partial p_i$, it follows that each point q generates an attractive force if $E(p, q) > 0$, and a repulsive force if $E(p, q) < 0$. This is consistent with the fact that a positive task error function represents a lack of resources, while a negative one represents an excess of resources. The resulting forces are determined by the signed sum of the forces generated by all the points of the space. Due to the signed sum, local minima are harder to recognize and to manage than if the positive error only is considered. In the latter case, only the lack of resources is penalized. This yields better performance with respect to the presence of local minima in the cost function J.

In order to asses numerically the existence of local minima in J, a scenario was set up with a Desired TDF D_* and two agents. D_* was chosen as the sum of two Gaussians identical in shape and dimensions to those of the two agents; this guarantees that an exact solution exists and that $\bar{J} = 0$.

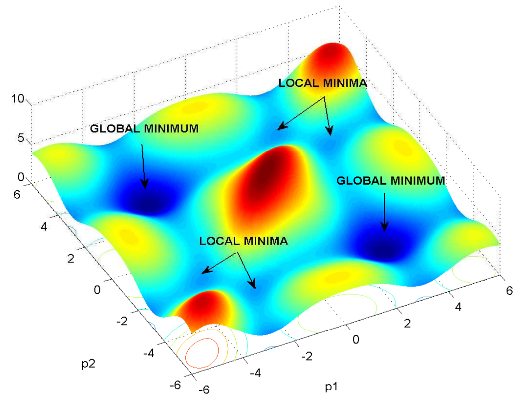
Figure 1(a) shows J for various values of p_1 and p_2, that is for various positions of the agents, for case 1. Two global minima and four local minima exist for J due to the simmetry of the problem.

Figure 1(b) shows J for various values of p_1 and p_2 for case 2. Two global minima exist, with no local minima. However, areas of the solution space exist where the slope of J is very limited and a slower convergence should be expected.

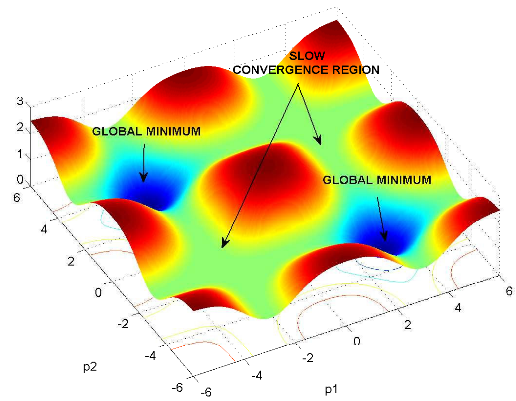
C. Algorithm Decentralization

The proposed approach assumes that all the agents know the Desired TDF $D_*(q, t)$ and can sense the presence of all the other agents through their respective $D_i(p_i, q)$.

From Eq.(10) it can be noted that if the agent DF is limited in space, i.e. $D_i(p_i, q) = 0$ if $|p_i - q| > R_D \in \Re^+$, and so is its derivative, then the control law can be computed exactly using only information in a neighborhood of the agent. The integrand of the control law is in fact zero where the derivative of the agent DF is zero. Even if this condition is appealing, it may yield local minima. An example of this occurrence is the case of a symmetric agent DF and constant TDF. In this case, if the agents DFs have no common points or do not intersect the environment boundary, integrand results to be a symmetric function and so the control signal is zero.



(a) $f(E(p,q)) = E(p,q,t)^2$



(b) $f(E(p,q)) = \max\left(0, E(p,q,t)\right)^2$

Fig. 1. Local and global minima of the cost function for different positions of two Gaussian agents that move in $1D$.

V. APPLICATION EXAMPLES

The deployment problem of a swarm of agents is considered as an example of application of the presented technique. The agents, starting from some compact initial configuration, must spread out such that the area covered by the network is maximized. A similar approach to solve this problem is found in [10] using a potential-field method: the network spreads itself throughout the environment using repulsion forces between the agents and the environment boundary. The network is considered homogeneous and the repulsion forces are computed as a function of the distance between the agents.

The control law proposed in this paper is capable of deploying the agents in the environment taking into account their sensing heterogeneity as well. The self-deployment objective is specified using the following Desired DF:

$$D_*(q) = \begin{cases} \bar{d} & if\ q \in P \\ 0 & otherwise \end{cases} \qquad (22)$$

where P is a generic polyhedron, non necessarily convex, that represents the area that the agents must cover. The weighting function $\sigma(q)$ is assumed to be constant for all $q \in Q$. The agents spread out due to the attractive forces generated by the error in the polyhedron. In fact, the initial task error function is very low in the compact set where the agents start, and larger in the rest of the polyhedron.

The final value attained by the cost function depends on the number of agents so the value of the global optimum is not necessarily zero. As the number of agents grows, the value of the cost function at the global optimum decreases, and reaches 0 only if there are enough agents to cover the request of the $D_*(q)$. Under some assumptions on the agent Descriptor Functions, and for the specific choice of $D_*(q)$ as in Eq.(22) the following proposition holds.

Proposition 3. If the Descriptor Function of the agent satisfies

- I1. $D_i(p_i, q) = D_i(|p_i - q|) = D_i(s)$
- I2. $\frac{\partial D_i(s)}{\partial s} < 0, \quad \forall s \neq 0$

i.e. it is a decreasing function of the Euclidean distance s to the agent, and if $f(E(p,q))$ in Eq. (7), satisfies

- I3. $f(E(p,q)), \frac{\partial f(E(p,q))}{\partial E(p,q)} \geq 0\ if\ E(p,q) \geq 0$
- I4. $f(E(p,q)), \frac{\partial f(E(p,q))}{\partial E(p,q)} = 0\ if\ E(p,q) < 0$

then if the agent starts in (or enters) a convex polyhedron, it will not exit from the boundary.

Proof. The control law of agent i of Eq.(10) can be rewritten as:

$$u_i = \alpha \int_Q \frac{\partial D_i(s)}{\partial s} \frac{\partial s}{\partial p_i} \frac{\partial f(E(p,q))}{\partial E(p,q)} \sigma(q) dq \qquad (23)$$

The control law in Eq. (23) is the sum of velocity vectors generated at each point $q \in Q$ for which $E(p,q) > 0$. Since s is the Euclidean distance between the agent and the point q, the quantity $\frac{\partial s}{\partial p_i}$ is a vector directed from the point to the agent position. Moreover, if $E(p,q) > 0$:

$$u_i = \alpha \int_Q \underbrace{\frac{\partial D_i}{\partial s}}_{\leq 0} \frac{\partial s}{\partial p_i} \underbrace{\frac{\partial f(E(p,q))}{\partial E(p,q)}}_{\geq 0} \underbrace{\sigma(q)}_{\geq 0} dq \qquad (24)$$

then the control law is the sum (integral) of vectors directed from the agent to points $q \in Q$. Furthermore, the interior of a convex polyhedron is a convex set, i.e. for each pair of points within the object, every point in the straight line that joints them is also within the object. This is true even for the points on the boundary.

Using this definition, even if an agent reaches the boundary of the polyhedron, the velocity vectors generated by each point in Q are directed toward the interior. Then, once inside, the agent does not exit form its boundaries.

■

Corollary 4. In the same assumptions of Proposition 2, if the agent starts in (or enters) a concave polyhedron, it will not exit from the convex hull of its vertices.

Proof. By definition, the convex hull of a set of points is the minimal convex set containing all the points. If $D_*(q)$

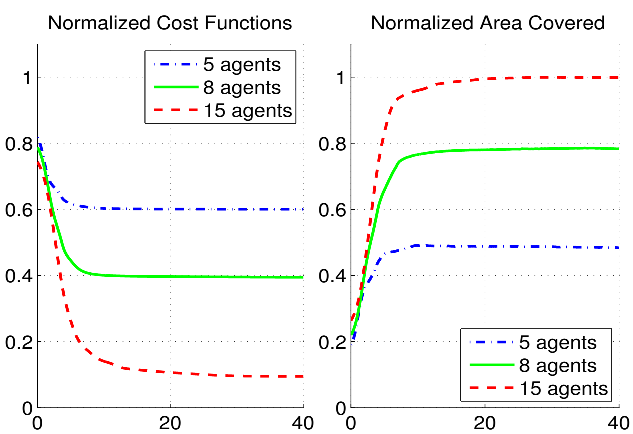
satisfies the conditions of Eq.(22) for a generic concave polyhedron, these conditions are satisfied even for the convex hull of its vertices. Then, the proof of Proposition 1 can be applied directly to the convex polyhedron obtained with the convex hull of the vertices.

■

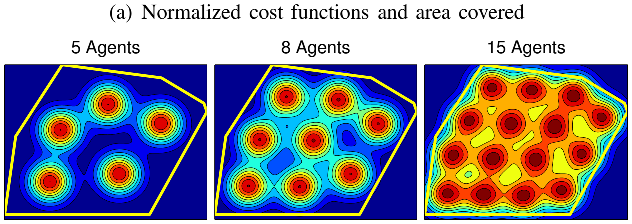
In order to asses the scalability of the algorithm, simulations were performed with 5, 8 and 15 agents to be deployed in a convex area. Fig. 2(a) shows the time histories of the normalized cost function (left), and the normalized area covered by the agents (right) for the three simulations. The agent DFs are Gaussian functions, i.e. $D_i(p_i, q) = e^{-\frac{1}{4}\left((q_1-p_{i1})^2+(q_2-p_{i2})^2\right)}$, and their initial positions are in a compact set around the origin. The area covered is computed as the area in which the Current TDF is greater than 0.2, normalized by the total area of the Desired TDF. Since the area covered increases, the deployment task can be formulated as an optimization problem within the DFs framework.

Fig. 2(b) shows the final deployment of the agents achieved for the convex polyhedron.

Finally Fig. 3 shows an example of deployment of 15 heterogeneous agents in a concave environment. Swarm heterogeneity is encoded in the parameters of the agent Descriptor Functions: i.e., each agent has a DF, which is a Gaussian function with an elliptical level set, with different parameters.



(a) Normalized cost functions and area covered



(b) Contour Plot of the Final TDF

Fig. 2. Deployment in a convex area.

VI. CONCLUSIONS

The paper presents a new framework for the deployment of a swarm of heterogeneous vehicles. The main contribution is the definitions of Descriptor Functions associated to each agent, the swarm, and the overall mission objectives. The stability analysis for the generic case and for the specific case of Area Coverage are presented. The capabilities and current limitations of the framework are discussed, together with numerical simulations for validation purposes.

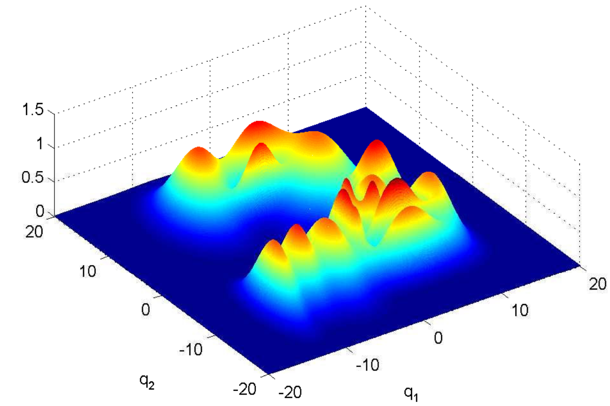


Fig. 3. Final positions of 15 heterogeneous agents in a concave environment.

ACKNOWLEDGEMENTS

The second author would like to thank Drs. David Jeffcoat and Henry Pfister of AFRL, Munitions Directorate for their suggestions in the directions of the research. The first and the second authors wish to thank Dr. John Row Rogacki for providing logistic support at the University of Florida REEF.

REFERENCES

[1] I. I. Hussein and D. M. Stipanovic, "Effective coverage control using dynamic sensor networks," in *Proc. 45th IEEE Conference on Decision and Control*, Dec. 13–15, 2006, pp. 2747–2752.

[2] ——, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 4, pp. 642–657, July 2007.

[3] ——, "Effective coverage control using dynamic sensor networks with flocking and guaranteed collision avoidance," in *Proc. American Control Conference ACC '07*, July 9–13, 2007, pp. 3420–3425.

[4] I. I. Hussein, D. M. Stipanovic, and Y. Wang, "Reliable coverage control using heterogeneous vehicles," in *Proc. 46th IEEE Conference on Decision and Control*, Dec. 12–14, 2007, pp. 6142–6147.

[5] N. Hubel, S. Hirche, A. Gusrialdi, T. Hatanaka, M. Fujita, and O. Sawodny, "Coverage control with information decay in dynamic environments," in *17th IFAC World Congress*, 2008.

[6] Y. Wang and I. I. Hussein, "Awareness coverage control over large scale domains with intermittent communications," in *Proc. American Control Conference*, June 11–13, 2008, pp. 4370–4375.

[7] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *Proc. and 2005 European Control Conference Decision and Control CDC-ECC '05. 44th IEEE Conference on*, Dec. 12–15, 2005, pp. 2542–2547.

[8] A. Gusrialdi, T. Hatanaka, and M. Fujita, "Coverage control for mobile networks with limited-range anisotropic sensors," in *Proc. 47th IEEE Conference on Decision and Control CDC 2008*, Dec. 9–11, 2008, pp. 4263–4268.

[9] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," 2001.

[10] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," 2002, pp. 299–308.