2016 IEEE Conference on Control Applications (CCA)
Part of 2016 IEEE Multi-Conference on Systems and Control
September 19-22, 2016. Buenos Aires, Argentina

Swarm Obstacle and Collision Avoidance using Descriptor Functions

Mario Innocenti[1], *Member, IEEE*, Lorenzo Pollini[1], *Member, IEEE*,
Giovanni Franzini[1], *Student Member, IEEE*, and Alessandro Salvetti[2]

***Abstract*— The descriptor function framework is used as tool for the control management of a swarm of dynamic agents. In this framework, a provision is made for obstacle and collision avoidance, thus improving the potential of the methodology from previous results. Obstacle and collision avoidance terms are added to the overall mission performance index, and the resulting control law moves the agents along obstacle and collision free trajectories. The analytical derivation is validated via numerical simulations.**

I. Introduction

The descriptor function approach was originally proposed in [1], and [2], following earlier work in [3], as an attempt to generalizing the control management of swarms, to encompass different agent characteristics, and different missions (tasks), within a common mathematical structure. The main ideas behind the framework are reviewed herein for clarity's sake.

We consider the problem of controlling a swarm of heterogeneous agents (typically autonomous vehicles), which must accomplish a cooperative mission, possibly composed by more than one tasks.

An *agent descriptor function* (ADF) is assigned to each agent that characterizes, in a general sense, the position, of the agent, the environment where the agent operates, and some of the specific properties of the payload (for instance a video camera, survival equipment, etc.). Clearly, the ADF could be a very complex even mathematically intractable function, and in the following it is considered continuous and continuously differentiable, but otherwise general. An example [4] is shown in Figure 1, where an ADF describing an ultrasound sensor (such as a sonar) in a two-dimensional environment is depicted. The third dimension, represented by level curves, could indicate the "intensity" of the particular sensor.

Mathematically, the ADF can be written as:

$$d_i(\boldsymbol{p}_i, \boldsymbol{q})\colon \mathcal{P} \times \mathcal{Q} \to \mathbb{R}^+$$

where the index i indicates the i-th agent, $\boldsymbol{p}_i \in \mathcal{P}$ is the state vector of agent i, $\mathcal{P} \subseteq \mathbb{R}^n$ is the spatial domain of each agent, $\mathcal{Q} \subseteq \mathbb{R}^m$, with $m = \{2, 3\}$, is the domain of the environment (spatial and not spatial), and $\mathbb{R}^+ = \{x \in \mathbb{R} \colon x \geq 0\}$. The motion of each agent, as common practice in this type of problems, is modeled by a single integrator:

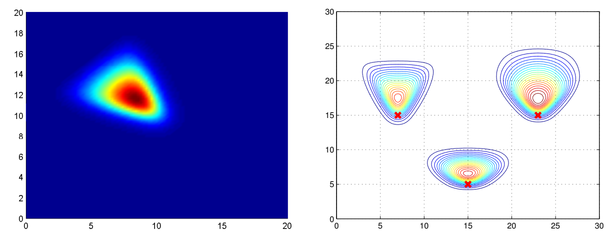


Fig. 1: Examples of ADFs (Sigmoid with Field of View).

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i$$

where $\boldsymbol{u}_i$ is the control vector of agent i. The task (or mission) to be accomplished is also described by a *task descriptor function* (TDF), which may describe the need for resources in a point $\boldsymbol{q} \in \mathcal{Q}$ of the environment, at time t:

$$d^*(t, \boldsymbol{q})\colon \mathbb{R}^+ \times \mathcal{Q} \to \mathbb{R}^+ \tag{1}$$

As an example, if we consider the uniform deployment task [5], [6], a very simple expression for Eq. (1) is a constant value.

The control law for each agent is then obtained by minimizing a *task error function* (TEF) defined as

$$e(t, \boldsymbol{p}, \boldsymbol{q}) = d^*(t, \boldsymbol{q}) - D(\boldsymbol{p}, \boldsymbol{q}), \quad \boldsymbol{p} = \left[\boldsymbol{p}_1^T, \ldots, \boldsymbol{p}_{N_a}^T\right]$$

where N_a denotes the number of agents, and $D(\boldsymbol{p}, \boldsymbol{q})$ is the *swarm descriptor function* given by

$$D(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{N_a} d_i(\boldsymbol{p}_i, \boldsymbol{q})$$

Using the procedure described in [2], and [7], we introduce the following cost function

$$J(t, \boldsymbol{p}) = \int_{\mathcal{Q}} f(e(t, \boldsymbol{p}, \boldsymbol{q}))\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q}$$

and we select a control law as:

$$\boldsymbol{u}_i = -\beta \left[\frac{\partial J(t, \boldsymbol{p})}{\partial \boldsymbol{p}_i}\right]^T, \quad \beta > 0 \tag{2}$$

The cost function J expresses the level of achievement of the task, where $\sigma(\boldsymbol{q})$ is a weight on the importance of the environment, and $f(\cdot) \geq 0$ is a weight function. Clearly the definition of an appropriate $f(\cdot)$ plays a critical role in

[1] M. Innocenti, L. Pollini and G. Franzini are with the Department of Information Engineering, University of Pisa, 56122 Pisa, Italy. {mario.innocenti,lorenzo.pollini}@unipi.it, giovanni.franzini@for.unipi.it
[2] A. Salvetti is with AgustaWestland S.p.A., 21015 Varese, Italy. alessandro.salvetti@hotmail.it

the computation of the agents' control laws, and on the task execution. Convergence and stability conditions can be found in the same references, using the cost index as a candidate Lyapunov function.

In this paper, we extend and improve the descriptor function framework by introducing obstacle and collision avoidance capabilities. The former is obtained by defining for each obstacle in the environment a descriptor function that dissuades the agents from visiting the obstacles region. Collisions are prevented introducing in the cost function J a penalization term that increases as the agents get too close. Therefore, conversely to most of the approaches proposed in literature, see for example [8], [9], [10], [11], obstacle and collision avoidance are incorporated directly into our framework, and are not introduced as augmentation terms for the existing control law. The effectiveness of the developed approach is proved by means of formal arguments, and supported by numerical simulations.

II. Obstacle Avoidance

The original format described in the previous section does not take into account the presence of obstacles in the scenario, in fact, according to most flocking methodologies, the agents' trajectories are the result of meeting mission objectives, rather than following a specific path. The problem of obstacle avoidance has been widely studied in the literature from the application of potential fields [12], to the use of gyroscopic effects [8], just to mention a few. In the area of path planning algorithms, it is common practice to modify the original control law with terms, which take into account the presence of static and/or dynamic obstacles.

In this section, we address the problem of incorporating the presence of obstacles directly in the descriptor function framework, in an attempt to generalize the agents' controller synthesis to account for scenarios with obstacles.

The introduction of an obstacle avoidance term in the framework starts from the nature of the selected optimization structure, that is the agents follow trajectories defined by a negative gradient of the cost J. The agents, in other words, move toward areas characterized by lack of resources (larger positive errors $d^* - D > 0$). If, for some areas, the TDF is too small or the swarm ADFs are limited, the agents will not direct themselves toward them. An obstacle, therefore, can be thought as an area of no interest for the swarm, and this idea will be incorporated in the TEF. With this in mind, we can define a new expression of the error as:

$$e(t, \boldsymbol{p}, \boldsymbol{q}) = d^*(t, \boldsymbol{q}) - D(\boldsymbol{p}, \boldsymbol{q}) - d_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q}) \tag{3}$$

The function $d_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q})\colon \mathcal{P} \times \mathcal{Q} \to \mathbb{R}^+$ is the *obstacle descriptor function* (ODF) associated to all obstacles present in the environment, and is defined as:

$$d_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{k=1}^{N_o} d^k_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q}) \tag{4}$$

where $d^k_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q})\colon \mathcal{P} \times \mathcal{Q} \to \mathbb{R}^+$ is the ODF associated to the k-th obstacle, and N_o is the number of obstacles in the environment. In the following, $\mathcal{Q}^k_{\text{obs}} \subset \mathcal{Q}$ will denote the area occupied by the obstacle k, and $\mathcal{Q}_{\text{obs}} = \bigcup_{k=1}^{N_o} \mathcal{Q}^k_{\text{obs}}$. Eq. (4) depends on the environment $\mathcal{Q}$, and, in general, on the position of the agents as well. As an example, we could assume a constant value, possibly time–varying, over each obstacle dimension.

A. Convergence and Stability

We now prove that with the new TEF definition, Eq. (3), the original control law still guarantees convergence to a local minimum.

Proposition 1. *For a time invariant task, the control law Eq.* (2) *guarantees converge to a local minimum.*

Proof. Considering Eq. (3), the time derivative of J is:

$$\frac{\mathrm{d}J(t,\boldsymbol{p})}{\mathrm{d}t} = \frac{\partial J(t,\boldsymbol{p})}{\partial t} + \frac{\partial J(t,\boldsymbol{p})}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} \tag{5}$$

where, having defined $f_e = \partial f(e)/\partial e$,

$$\begin{aligned} \frac{\partial J(t,\boldsymbol{p})}{\partial t} &= \int_{\mathcal{Q}} f_e \frac{\mathrm{d}d^*(t,\boldsymbol{q})}{\mathrm{d}t}\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q} \\ \frac{\partial J(t,\boldsymbol{p})}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} &= -\sum_{i=1}^{N_a}\left[\int_{\mathcal{Q}} f_e \left(\frac{\partial d_i(\boldsymbol{p}_i,\boldsymbol{q})}{\partial \boldsymbol{p}_i} \right.\right. \\ &\qquad \left.\left. + \sum_{k=1}^{N_o}\frac{\partial d^k_{\text{obs}}(\boldsymbol{p},\boldsymbol{q})}{\partial \boldsymbol{p}_i}\right)\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q}\right]\dot{\boldsymbol{p}}_i \end{aligned} \tag{6}$$

Introducing the control law, Eq. (2), in Eq. (5) we have that

$$\frac{\mathrm{d}J(t,\boldsymbol{p})}{\mathrm{d}t} = \frac{\partial J(t,\boldsymbol{p})}{\partial t} - \beta\left\|\frac{\partial J(t,\boldsymbol{p})}{\partial \boldsymbol{p}}\right\|^2$$

For a time invariant task $\partial J(t,\boldsymbol{p})/\partial t = 0$, and $\mathrm{d}J(t,\boldsymbol{p})/\mathrm{d}t \leq 0$, which guarantees convergence to a stable local minimum. □

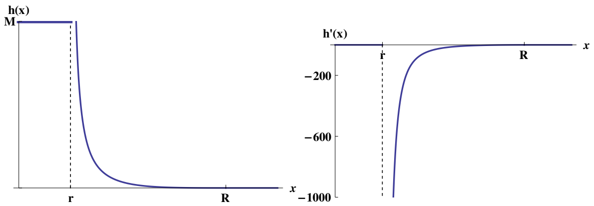
If the task is time varying, $\partial J(t,\boldsymbol{p})/\partial t \neq 0$, and the behavior of the cost function time derivative should be evaluated accordingly. In the following, we prove convergence for an effective coverage task, that could describe an exhaustive search over some area.

Proposition 2. *For an effective coverage task, whose TDF can be written as [2]*

$$d^*(t,\boldsymbol{q}) = \max\left\{0, \overline{C} - \int_0^t D(\boldsymbol{p}(\tau),\boldsymbol{q})\mathrm{d}\tau\right\},\ \overline{C} > 0 \tag{7}$$

where $\overline{C}$ is the amount of resource necessary to assume the area be sufficiently searched, and $f(e) = e^2$, the control law Eq. (2) *guarantees convergence to a local minimum.*

Proof. The derivative of the weight function, that is $f_e = 2e$, has an unknown sign. Thus, the integral sign in Eq. (6) is unknown as well. If, however, we set $\boldsymbol{u}_i = \boldsymbol{0}$, the agents tend to cancel d^* in the area covered by their ADFs. Thus, the error in Eq. (3) decreases, and so does the performance index



(a) $h(x)$ (b) $h'(x)$

Fig. 2: Graph of Eq. (9) for $r = 2$, and $R = 8$.

J. As a matter of fact, according to Eqs. (5), (6) and (7), the time derivative of J, for $\boldsymbol{u} = \boldsymbol{0}$, is equal to:

$$\left.\frac{\mathrm{d}J(t,\boldsymbol{p})}{\mathrm{d}t}\right|_{\boldsymbol{u}=\boldsymbol{0}} = \frac{\partial J(t,\boldsymbol{p})}{\partial t} = -\int_{\mathcal{Q}} f_e \sum_{i=1}^{N_a} d_i(\boldsymbol{p}_i(t),\boldsymbol{q})\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q}$$

Therefore, $\partial J(t,\boldsymbol{p})/\partial t \le 0$, and, recalling the results in Proposition 1, $\mathrm{d}J(t,\boldsymbol{p})/\mathrm{d}t \le 0$. ☐

B. The Obstacle Descriptor Function

The presence of the obstacle descriptor function d_{obs} allows the swarm to move towards the target defined by the task, but there could be cases where the attracting component of the TDF is still such as to overcome d_{obs}. The following proposition states under which condition this cannot happen.

Proposition 3. *If $d_{obs}^{\overline{k}} \to +\infty$ when the position of at least one agent tends to the border of the $\overline{k}$-th obstacle, then the agents will never collide against that obstacle.*

Proof. Suppose $d_{\text{obs}}^{\overline{k}} \to \infty$, then

$$e(t,\boldsymbol{p},\boldsymbol{q}) = d^*(t,\boldsymbol{q}) - D(\boldsymbol{q},\boldsymbol{q}) - \sum_{k=1}^{N_o} d_{\text{obs}}^k(\boldsymbol{p},\boldsymbol{q}) \to +\infty$$

Thus, $J \to +\infty$. Since we know that $\mathrm{d}J(t,\boldsymbol{p})/\mathrm{d}t \le 0$, the cost functional can never increase to infinity. Therefore it is not possible to have $d_{\text{obs}}^k \to \infty$, and the agents will not collide with the obstacle. ☐

A possible choice for the obstacle descriptor function is

$$d_{\text{obs}}^k(\boldsymbol{p},\boldsymbol{q}) = \begin{cases} \sum_{i=1}^{N_a} h(\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\|), & \boldsymbol{q} \in \mathcal{Q}_{\text{obs}}^k \\ 0, & \boldsymbol{q} \notin \mathcal{Q}_{\text{obs}}^k \end{cases} \tag{8}$$

where $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is a selection matrix that applied to the agent state $\boldsymbol{p}_i$ gives its position in $\mathcal{Q}$, $\boldsymbol{c}_i^k \in \mathcal{Q}_{\text{obs}}^k$ is the point of the k-th obstacle nearest to the i-th agent, and $h(x) \ge 0$ $\forall x$. In the light of Proposition 3, we require that $h(\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\|) \to +\infty$ as $\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\| \to 0^+$, that is as the agent i reaches the border of the obstacle. A smooth function $h(\cdot)$ that satisfies the above requirements is given by:

$$h(x) = \begin{cases} \max\left\{0, -\frac{(x-R)^3}{x-r}\right\}, & x > r \\ M, & x \le r \end{cases} \tag{9}$$

where $M > 0$, and $0 \le r < R < +\infty$. The parameter r can be tuned according to the minimum distance from the obstacle we want to maintain, whereas R defines at what distance the obstacle influences the agent motion (see Figure 2). According to Eq. (2), the new control law, in the presence of the obstacle avoidance term, becomes:

$$\begin{aligned} \boldsymbol{u}_i =& \beta\Bigg[\int_{\mathcal{Q}} f_e \left[\frac{\partial d_i(\boldsymbol{p}_i,\boldsymbol{q})}{\partial \boldsymbol{p}_i}\right]^T \sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q} \\ &+ \sum_{k=1}^{N_o} \int_{\mathcal{Q}_{\text{obs}}^k} f_e \left[\frac{\partial d_{\text{obs}}^k(\boldsymbol{p},\boldsymbol{q})}{\partial \boldsymbol{p}_i}\right]^T \sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q}\Bigg] \\ =& \beta\Bigg[\int_{\mathcal{Q}} f_e \left[\frac{\partial d_i(\boldsymbol{p}_i,\boldsymbol{q})}{\partial \boldsymbol{p}_i}\right]^T \sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q} \\ &+ \sum_{k=1}^{N_o} \boldsymbol{S}^T(\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k)\frac{h'(\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\|)}{\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\|}\int_{\mathcal{Q}_{\text{obs}}^k} f_e\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q}\Bigg] \\ =& \beta\left[\boldsymbol{u}_i^{\text{task}} + \boldsymbol{u}_i^{\text{obs}}\right] \end{aligned} \tag{10}$$

where $h'(x) = \partial h(x)/\partial x$.

The obstacle influences both terms of the control law, since the TEF e, and consequently f_e, depends on the ODF (see Eq. (3)). The contribution of the obstacle to $\boldsymbol{u}_i^{\text{task}}$ depends on the ADF, i.e. if $d_i(\boldsymbol{p}_i,\boldsymbol{q}) \gg 0$ for $\boldsymbol{q} \in \mathcal{Q}_{\text{obs}}^k$ then the k-th obstacle actually has a repulsive influence on the agents. The obstacle avoidance contribution to $\boldsymbol{u}_i^{\text{obs}}$ is independent of the ADF and the task. In particular, if $h'(\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\|) = 0$ $\forall k$, then $\boldsymbol{u}_i^{\text{obs}} = \boldsymbol{0}$. This occurs when $\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^k\| \ge R$ $\forall k$, that is when the agent is sufficiently away from all the obstacles.

Proposition 4. *If the weight function $f(\cdot)$ is chosen such that $f_e(\cdot) \le 0$ for $\boldsymbol{q} \in \mathcal{Q}_{obs}$, then the component $\boldsymbol{u}_i^{obs}$ of the control law in Eq. (10) will always move the agents away from the obstacles when the distance is less than R.*

Proof. Without loss of generality, consider the influence of a single obstacle $\overline{k}$ on the agent i. From Eq. (10) we have:

$$\boldsymbol{u}_i^{\text{obs}} = \boldsymbol{S}^T(\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}})\frac{h'(\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}}\|)}{\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}}\|}\int_{\mathcal{Q}_{\text{obs}}^{\overline{k}}} f_e\sigma(\boldsymbol{q})\mathrm{d}\boldsymbol{q} \tag{11}$$

Given the hypothesis on f_e, the integral term in Eq. (11) is negative. Recalling from Figure 2 that $h(\cdot) \ge 0$, and $h'(\cdot) \le 0$, we can then write Eq. (11) as follows,

$$\boldsymbol{u}_i^{\text{obs}} = \gamma\boldsymbol{S}^T\frac{\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}}}{\|\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}}\|}, \quad \gamma \ge 0$$

So the control law has a magnitude γ and a direction equal to $\boldsymbol{S}\boldsymbol{p}_i - \boldsymbol{c}_i^{\overline{k}}$, which is normal to the obstacle surface and away from it. ☐

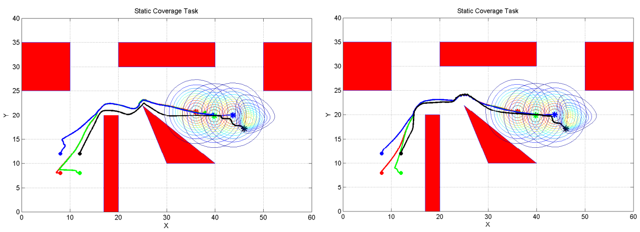
C. Weight Functions $f(\cdot)$

The actual computation of the control law depends of course on the specific form of the weight function $f(\cdot)$. One of the expressions most frequently used in the literature is the quadratic function:

$$f(e) = e^2 \tag{12}$$

This, however results in a large number of local minima and a penalty on the case of excess of resources. The latter fact can be mitigated using:

$$f(e) = \max\left\{0, e\right\}^2 \tag{13}$$



(a) No Obstacle Avoidance. (b) With Obstacle Avoidance.

Fig. 3: Obstacle Avoidance in a Static Coverage Task.

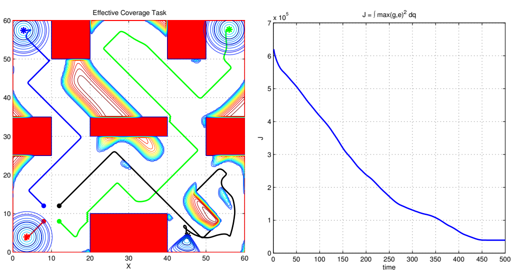


Fig. 4: Obstacle Avoidance in an Effective Coverage Task.

(see [7]). Note that both Eqs. (12) and (13) respect the condition $f_e(\cdot) \leq 0$ for $\boldsymbol{q} \in \mathcal{Q}_{\text{obs}}$ stated in Proposition 4. As a matter of fact, the derivative of Eqs. (12) and (13) are $2e$ and $2\max\{0, e\}$ respectively, and for $\boldsymbol{q} \in \mathcal{Q}_{\text{obs}}$ the TEF is such that

$$e(t, \boldsymbol{p}, \boldsymbol{q}) = -D(\boldsymbol{p}, \boldsymbol{q}) - d_{\text{obs}}(\boldsymbol{p}, \boldsymbol{q}) \leq 0$$

since $d^* = 0$ on the obstacles region.

The weight function in Eq. (13) reduces the effectiveness of the control law in Eq. (10), since the elimination of negative errors will cancel the repulsive contribution from $\boldsymbol{u}_i^{\text{task}}$. We can overcome this problem by introducing the following function:

$$g(\boldsymbol{q}) = \begin{cases} -M, & \boldsymbol{q} \in \mathcal{Q}_{\text{obs}} \\ 0, & \boldsymbol{q} \notin \mathcal{Q}_{\text{obs}} \end{cases} \tag{14}$$

with $M > 0$. Using Eq. (14) we define the new weight:

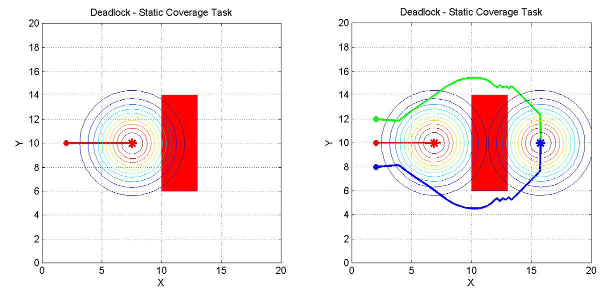
$$f(e(t, \boldsymbol{p}, \boldsymbol{q})) = \max\{g(\boldsymbol{q}), e(t, \boldsymbol{p}, \boldsymbol{q})\}^2 \tag{15}$$

Note that Eq. (15) is such that $f_e(\cdot) \leq 0$ for $\boldsymbol{q} \in \mathcal{Q}_{\text{obs}}$. The simulations results presented in the following sections were obstained using Eq. (15).

D. Examples

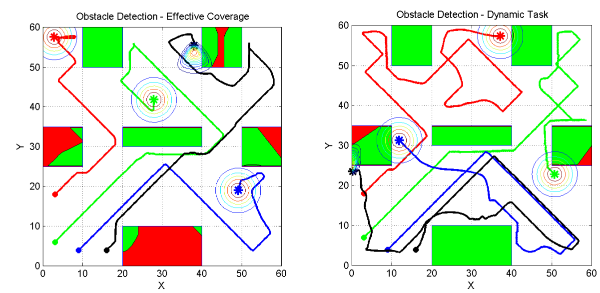
Figure 3 shows the influence of the obstacle avoidance component of the control law in a static coverage task. The swarm has four agents (three omnidirectional and one with field of view), which move from left to right of the environment. The graph on the left shows the behavior with $\boldsymbol{u}_i = \beta \boldsymbol{u}_i^{\text{task}}$, whereas the graph on the right has the complete control law. The benefit in obstacle avoidance is evident.

Figure 4 shows the example of an effective coverage task. The cost function is continuously decreasing but it cannot reach zero because of the presence of obstacles in the environment. The "uneven" behavior of the agents (some move a lot and some very little) is typical of the effective coverage as also reported in [2], [3]. This may be desirable or not depending on how tight the swarm is supposed to be moving during the task.



(a) One Agent Deadlock. (b) Three Agents Deadlock.

Fig. 5: Examples of Deadlock.



(a) Obstacle Detection During an Effective Coverage Task. (b) Obstacle Detection During a Dynamic Coverage Task.

Fig. 6: Obstacle Detection Simulations.

E. Special Cases

One of the issues that we encounter in motion planning over obstacles is the presence of degenerated cases leading to deadlock. A classic example is a geometric symmetry as shown in Figure 5. The current framework suffers from deadlock in some cases, but we remind the reader that, in practice, the uncertainties present in the modeling, first of all the use of single integrator as dynamics, make perfect symmetry very unlikely in most instances. In addition, during the motion of the swarm, complete symmetry of the agent descriptor function is hardly achievable because of the nature of problem itself.

The ODF in Eq. (8) can also be used for obstacle detection, depending on the agent capability to do so. This is described in Figure 6, where two examples of effective coverage and dynamic coverage are shown. In the figure, the color red indicates the obstacle as undetected, while the color green shows detection by the agents as the task unfolds. The starting point of the swarm is at the bottom left corner of the scenario. We note that the ODF is not contributing to the overall control law if the obstacle is set at some distance greater than R. The detection occurs as soon as the obstacle is in the field of view of the agent, that is a threshold present in the agent descriptor function ADF. Based on this intersection, a portion of the obstacle is considered known, and the obstacle avoidance controller takes over. It is clear from Figure 6 that this process is more accurate in a dynamic coverage task (most obstacles are green), since the agents perform an accurate search of the environment, as compared to an effective coverage scenario. Finally, the inclusion of an obstacle avoidance term in the framework could also be used to establish areas of the environment that are of less use for the swarm, so that resources are not wasted.

III. Collision Avoidance

The collision avoidance problem deals with the requirements necessary to maintain a minimum inter-agent distance, in order to avoid collisions. This problem was not addressed in the original work [1], [2], and is widely studied in the literature in areas dealing with connectivity maintenance (structure of the underlying graph) or flocking [12], [13]. An ad hoc approach to the problem can also be found in [4].

We define a new cost function J as follows,

$$J(t,\boldsymbol{p}) = J^{\mathrm{DF}}(t,\boldsymbol{p}) + \omega J^{\mathrm{CA}}(\boldsymbol{p}), \quad \omega > 0 \tag{16}$$

where J^{DF} denotes the cost function used in the previous section, and J^{CA} is a term that takes in account the possibility of collision among the agents. Following the approach in [13], [9], which is based on potential theory considerations, we can use Eq. (16) as a candidate Lyapunov function, i.e. $V(t,\boldsymbol{p}) = J(t,\boldsymbol{p})$, and use the sign of its time derivative to compute a new control law defined as

$$\boldsymbol{u} = \boldsymbol{u}^{\mathrm{DF}} + \boldsymbol{u}^{\mathrm{CA}}$$

where $\boldsymbol{u}^{\mathrm{DF}}$ denotes the control law stated in the previous section (see Eqs. (2) and (10)), and $\boldsymbol{u}^{\mathrm{CA}}$ is the collision avoidance term. The time derivative of V is given by:

$$\frac{\mathrm{d}V(t,\boldsymbol{p})}{\mathrm{d}t} = \frac{\partial J^{\mathrm{DF}}(t,\boldsymbol{p})}{\partial t} + \frac{\partial J^{\mathrm{DF}}(t,\boldsymbol{p})}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} + \omega \frac{\partial J^{\mathrm{CA}}(\boldsymbol{p})}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} \tag{17}$$

Considering Eq. (2) and choosing

$$\boldsymbol{u}_i^{\mathrm{CA}} = -\alpha \left[\frac{\partial J^{\mathrm{CA}}(\boldsymbol{p})}{\partial \boldsymbol{p}_i}\right]^T, \quad \alpha > 0 \tag{18}$$

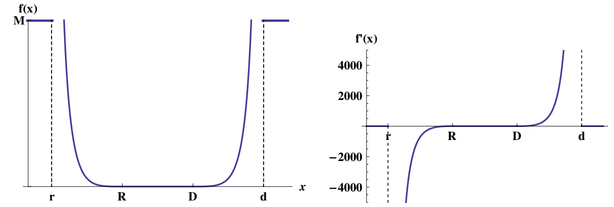
Eq. (17) can be written as:

$$\frac{\mathrm{d}V(t,\boldsymbol{p})}{\mathrm{d}t} = \frac{\partial J^{\mathrm{DF}}(t,\boldsymbol{p})}{\partial t} - \left[\frac{\partial J^{\mathrm{DF}}(t,\boldsymbol{p})}{\partial \boldsymbol{p}}, \frac{\partial J^{\mathrm{CA}}(\boldsymbol{p})}{\partial \boldsymbol{p}}\right] \cdot \\ \cdot \boldsymbol{W} \left[\left[\frac{\partial J^{\mathrm{DF}}(t,\boldsymbol{p})}{\partial \boldsymbol{p}}\right]^T, \left[\frac{\partial J^{\mathrm{CA}}(\boldsymbol{p})}{\partial \boldsymbol{p}}\right]^T\right]^T$$

where:

$$\boldsymbol{W} = \overline{\boldsymbol{W}} \otimes \boldsymbol{I}_{2\times 2} = \begin{bmatrix} \beta & \frac{1}{2}(\alpha + \omega\beta) \\ \frac{1}{2}(\alpha+\omega\beta) & \omega\alpha \end{bmatrix} \otimes \boldsymbol{I}_{2\times 2}$$

As discussed in the previous section, the time varying component $\partial J^{\mathrm{DF}}(t,\boldsymbol{p})/\partial t$ is equal to zero for time invariant tasks. Otherwise, its sign must be studied according to the task definition. However, in Proposition 2, we proved that for the effective coverage task we have that $\partial J^{\mathrm{DF}}(t,\boldsymbol{p})/\partial t \leq 0$.



(a) $f(x)$ (b) $f'(x)$

Fig. 7: Selected Expression for $f(x)$ in Eq. (20).

Therefore, we need the symmetric weighting matrix $\boldsymbol{W}$ to be positive definite, i.e. the coefficient α, β and ω must be selected such that the eigenvalues of $\overline{\boldsymbol{W}}$ are strictly positive. The above constraints allow the system to evolve towards a local minimum (local maxima and saddle points are unstable and can be avoided [1], [2]).

In order to extend the work in [13], [9] without the use of potential flow fields, we propose a cost function that increases indefinitely as the distance between two agents becomes lower or greater than some limiting values selected by design. Consider the following expression:

$$\begin{aligned} J^{\mathrm{CA}}(\boldsymbol{p}) &= \sum_{i=1}^{N_a}\sum_{\substack{j=1\\ j\neq i}}^{N_a} f(\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|) \\ &= 2\sum_{\substack{i=1\\ j=i+1}}^{N_a} f(\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|) \end{aligned} \tag{19}$$

with:

$$f(x) = \begin{cases} \max\left\{0, \frac{(R-x)^3(x-D)^3}{(x-r)(x-d)}\right\}, & r \leq x \leq d \\ M, & \text{otherwise} \end{cases} \tag{20}$$

with $M > 0$, and $0 \leq r < R \leq D < d$. As shown in Figure 7, the agents are constrained in relative distance by r and d respectively. If the relative distance is between R and D, the contribution of J^{CA} is zero, increasing the flexibility of the proposed solution.

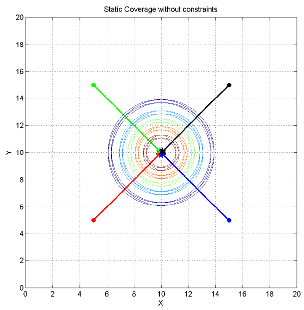
Given Eqs. (18), (19), and (20), we have that,

$$\boldsymbol{u}_i^{\mathrm{CA}} = -2\alpha \sum_{\substack{j=1\\ j\neq i}}^{N_a} \frac{f'(\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|)}{\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|}\boldsymbol{S}^T\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j) \tag{21}$$

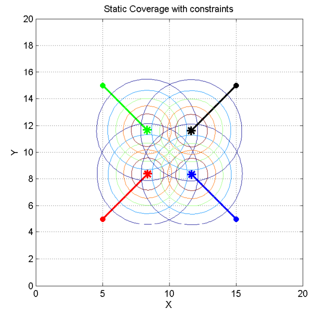
where $f'(x) = \partial f(x)/\partial x$. From Eq. (21) we see that the i-th agent is subjected to an action by the other $N_a - 1$ agents, with amplitude and direction specified by $-\alpha f'(\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|)$. The above comments confirm the behavior described in Figure 7, from which the i-th agent moves away from the j-th one when their distance is less than R, since $-\alpha f'(\|\boldsymbol{S}(\boldsymbol{p}_i - \boldsymbol{p}_j)\|) > 0$. The opposite occurs when their relative distance is greater than D.

A. Examples

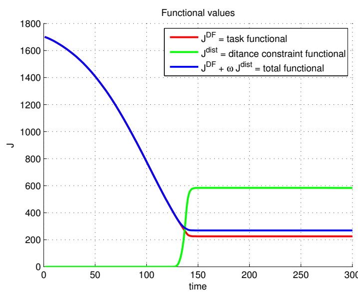
Consider a static coverage task with four identical agents in a symmetric configuration. Figure 8 shows the case of absence and presence of the collision avoidance term in



(a) No Collision Avoidance.

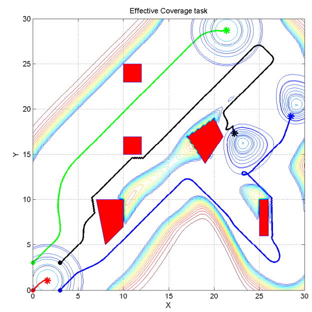


(b) With Collision Avoidance

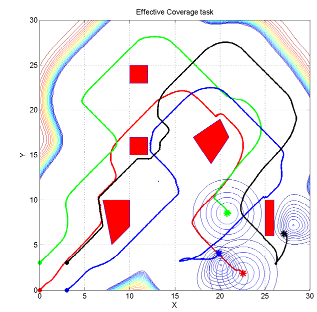


(c) Cost Functions Evolutuions.

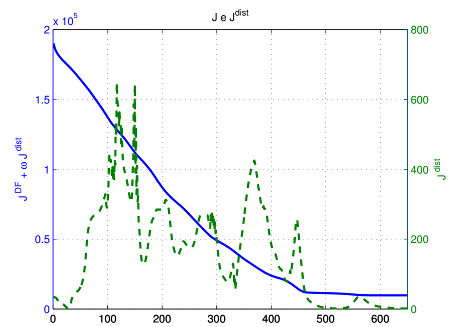
Fig. 8: Collision Avoidance for a Static Coverage Task.



(a) No Collision Avoidance.



(b) With Collision Avoidance



(c) Cost Functions Evolutions.

Fig. 9: Collision Avoidance for an Effective Coverage Task.

Eq. (21) (Figure 8a and 8b, respectively). The behavior of the cost function is shown in Figure 8c. From the figure, the contribution of the collision avoidance term is evident, when the relative distance becomes lower than a prescribed value.

Next we consider an effective coverage scenario. Figure 9a shows the behavior of the swarm without the collision avoidance term. In this case the red agent does not contribute as much to the task, due to the selected scenario cost function. The influence of the collision avoidance term is clear in Figure 9b. In particular, this contribution increases connectivity as well since there is a constraint on the maximum relative distance, and the red agent now contributes fully to the task. Figure 9c describes the values of the cost function. The green dashed curve indicates the contribution of the collision avoidance term, whose slope depends of course on maintaining the agents within the relative distance constraints.

IV. Conclusions

The paper presents an analytical structure for obstacle and collision avoidance of a swarm of agents. The descriptor function framework for the control management of the swarm is extended, and its flexibility in handling different tasks is increased. The resulting control law maintains local stability and convergence properties, and the results are validated via simulation for static, dynamic, and effective coverage tasks. Current work is directed towards experimental validation.

References

[1] M. Niccolini, M. Innocenti, and L. Pollini, "Near optimal swarm deployment using descriptor functions," in *Proc. 2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010, pp. 4952–4957.

[2] M. Niccolini, L. Pollini, and M. Innocenti, "Cooperative control for multiple autonomous vehicles using descriptor functions," *J. Sens. Actuator Netw.*, vol. 3, no. 1, pp. 26–43, 2014.

[3] I. I. Hussein and D. M. Stipanović, "Effective coverage using dynamic sensor networks," in *Proc. 45th IEEE Conference on Decision and Control*, San Diego, California, 2006, pp. 2747–2752.

[4] A. Ferrari Braga, M. Innocenti, and L. Pollini, "Multi-agent coordination with arbitrarily shaped descriptor function," in *Proc. 2013 AIAA Guidance, Navigation, and Control Conference*, Boston, Massachusetts, 2013.

[5] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009.

[6] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer Japan, 2002, pp. 299–308.

[7] M. Niccolini, "Swarm abstractions for distributed estimation and control," Ph.D. dissertation, Univ. of Pisa, Pisa, Jul. 2011.

[8] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," in *Proc. 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, 2003, pp. 539–543.

[9] I. I. Hussein and D. M. Stipanović, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 4, pp. 642–657, 2007.

[10] G. M. Atinç, D. M. Stipanović, P. G. Voulgaris, and M. Karkoub, "Swarm-based dynamic coverage control," in *Proc. 53rd IEEE Conference on Decision and Control*, Los Angeles, California, 2014, pp. 6963–6968.

[11] V. G. Santos and L. Chaimowicz, "Cohesion and segregation in swarm navigation," *Robotica*, vol. 32, pp. 209–223, 2014.

[12] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[13] I. I. Hussein and D. M. Stipanović, "Effective coverage control using dynamic sensor networks with flocking and guaranteed collision avoidance," in *Proc. 2007 American Control Conference*, New York City, USA, 2007, pp. 3420–3425.