Obstacle Avoidance in Formation
using Navigation-like Functions and Constraint Based Programming

Michele Colledanchise[†], Dimos V. Dimarogonas[‡] and Petter Ögren[†]

Abstract— In this paper, we combine navigation function-like potential fields and constraint based programming to achieve obstacle avoidance in formation. Constraint based programming was developed in robotic manipulation as a technique to take several constraints into account when controlling redundant manipulators. The approach has also been generalized, and applied to other control systems such as dual arm manipulators and unmanned aerial vehicles. Navigation functions are an elegant way to design controllers with provable properties for navigation problems. By combining these tools, we take advantage of the redundancy inherent in a multi-agent control problem and are able to concurrently address features such as formation maintenance and goal convergence, even in the presence of moving obstacles. We show how the user can decide a priority ordering of the objectives, as well as a clear way of seeing what objectives are currently addressed and what are postponed. We also analyze the theoretical properties of the proposed controller. Finally, we use a set of simulations to illustrate the approach.

I. INTRODUCTION

Multiagent coordination has received a lot of attention in recent years [1]–[8]. The reason is that advantages in terms of robustness, performance and flexibility can be realized when tasks are performed by groups of smaller agents, instead of highly complex single agents. Examples of such tasks are exploration, search and rescue, surveillance, cooperative lawn mowing and harvesting. An important and frequently needed capability when performing tasks such as the above, is that of navigation from one place to another in a structured fashion.

Navigation can either be done in a hierarchical way with planning followed by open loop execution, or using continuous feedback control. In this paper we will use navigation functions [9] to provide feedback control, which gives increased robustness in the presence of moving obstacles.

While performing this navigation, we want to achieve formation keeping, obstacle avoidance, and goal convergence. However, due to the unknown motion of the obstacles, we might have to temporarily sacrifice some of these objectives, in order to achieve the others. For example, the desired formation might have to be broken to pass through a narrow passage to reach the goal, or the progress towards goal might have to be postponed, if a moving obstacle is blocking the

[†]The author is with the Centre for Autonomous Systems, Computer Vision and Active Perception Lab, School of Computer Science and Communication, The Royal Institute of Technology - KTH, Stockholm, Sweden.
[‡]The author is with the Centre for Autonomous Systems, Automatic Control Lab, School of Electrical Engineering, The Royal Institute of Technology - KTH, Stockholm, Sweden.

destination. In order to handle multiple possibly contradictory objectives, we apply a constraint based programming approach [10].

Constraint based programming is a way to exploit the redundancy of a robotic system to simultaneously achieve a number of tasks. The approach is founded upon the early work of [11]–[13], while recent contributions include the iTasc framework (Instantaneous Task Specification using Constraints) [14], [15], and generalizations including a margin based handling of inequalities and homogenous handling of both equalities and inequalities in the context of both dual arm manipulation and unmanned aerial vehicle control [10], [16], [17].

Artificial Potential Fields are a class of methods introduced by Khatib [18] to solve motion planning problems. By defining a scalar potential over the workspace, and then following the negative gradient of this potential, a path to the goal can be found. However, some problem instances can exhibit local minima of the potential function, where the robot can get stuck. Rimon and Koditschek [9] extended the potential field approach by defining Navigation Functions, potential fields that were designed to be free of local minima. Their work was later extended to different robots, including limited sensing range non-holonomic systems [19], multiagent systems [20]–[22] and different obstacle settings [23].

The contribution of this work is the combination the recent advances on navigation functions with the new generalization of constraint based programming. Doing this, we are able to add flexible formation keeping capabilities to the robust navigation performance provided by appropriately designed and tuned navigation-like functions.

The outline of the paper is as follows. In Section II we present the problem we address. Section III then describes the proposed solution. The theoretical properties of the solution are analyzed in Section IV. Finally, simulations are presented to illustrate the approach in Section V and conclusions are drawn in Section VI.

II. PROBLEM STATEMENT

Consider a system of $M \in \mathbb{N}$ spherical agents among $N \in \mathbb{N}$ moving obstacles operating in the same sphere world $\mathcal{W} \subset \mathbb{E}^n$. Let $q_{a_i} \in \mathbb{E}^n$ denote the center position of the agent i and $r_i \in \mathbb{R}_+$ its radius, $q_{o_j} \in \mathbb{E}^n$ the center position of the obstacle j and $\rho_j \in \mathbb{R}_+$ its radius. The motion of each agent is described by a single integrator

$$\dot{q}_{a_i} = u_i, \qquad (1)$$

where the control input u_i denotes the velocity of agent i. The goal position for each agent is given by a point $d_i \in \mathcal{W}$. The objective of the controller is thus to reach the goal position, without colliding with obstacles or other agents, and keeping a desired formation while doing this.

Let the target formation be given by the desired spacing $\lambda_{ih} > r_i + r_h$, between agents i and h. To measure formation maintenance we define the *formation function* (FF) as follows

$$\psi \triangleq \sum_{i=1}^{M} \sum_{\substack{h=1 \\ h \neq i}}^{M} \frac{1}{2}(\|q_{a_i} - q_{a_h}\|^2 - \lambda_{ih}^2)^2 \tag{2}$$

Note that this function reaches its minimum value $\psi = 0$, when all agents are in the desired formation. It is assumed that this formation is defined such that it is achieved when the agents are at their desired destinations.

The problem considered in the paper is provided below.

Problem 1: Given a system of agents evolving according to (1) find controls u_i such that

1) The destinations are reached, i.e., q_{a_i} converges to d_i from almost all initial conditions.
2) The desired formation is reached and then maintained, $\psi \to 0$.
3) Collision avoidance is guaranteed.

There are trivial examples for which the three objectives described above cannot be fulfilled. In order to tackle this issue we propose a prioritized framework that puts higher priority to the most critical task at each time instant. This is enforced by the use of the constrained programming methodology described in the sequel.

III. PROPOSED SOLUTION

We will use a semi-decentralized approach, where the control signals of the agents are computed in a so-called *active redundant* way [24]. In detail, all agents compute their own control signal, as well as the control signal of their fellow agents. If all agents have the same information, as is assumed in this paper, they will all reach the same results u_i regarding any given agent i, and the group will act in a coordinated fashion.

To reach the three goals of Problem 1 above, we propose the following switched control structure

$$u_i(t) \triangleq \begin{cases} u_{i_{OPT}}(t) & \text{if} \quad \varphi_i(q_{a_i}(t)) \leq \varphi_i(q_{a_i}(t_0)) - t\epsilon, \\ u_{i_{CF}}(t) & \text{otherwise} \end{cases} \quad \forall i \tag{3}$$

where φ_i is the potential function proposed for the agent i as further described, $\epsilon \geq 0$ is a non negative parameter whose inverse is proportional to an initial estimate the user has on the transient response time of the robotic team while being driven to the goal configuration. If the goal configuration has not been reached at a given time based on the first controller $u_{i_{OPT}}$, then the second controller becomes active from that time onwards. Note that $\epsilon = 0$ thus corresponds to an infinite estimate of the response time, which make sense in the case of Solution 2 below. Depending on the user priorities, we then have two different parameter settings.
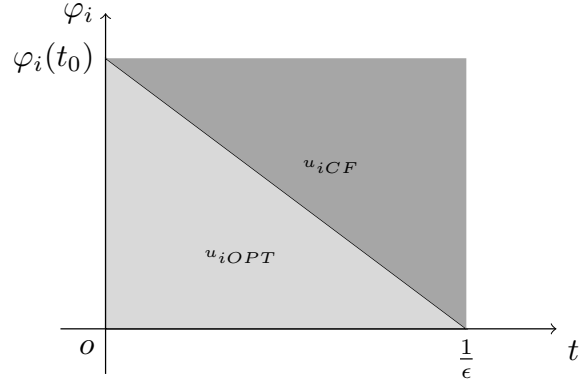

Fig. 1. The switching surface of u_i. As long as the navigation function is inside the bright area the control input of the system is $u_{i_{OPT}}$, when it is outside the control input is $u_{i_{CF}}$

Solution 1 (Goal Convergence): If 1) is more important that 2) in Problem 1, then we use $\epsilon > 0$ and $h_j \gg h_{M+1}$, $j = \{1, \ldots M\}$.

Solution 2 (Formation Maintenance): If 2) is more important that 1) in Problem 1, then we use $\epsilon = 0$ and $h_j \ll h_{M+1}$, $j = \{1, \ldots M\}$.

The switching surface of the control u_i in (3) is given in Figure 1. The controls $u_{i_{CF}}(t), u_{i_{OPT}}(t)$ are given below. $u_{i_{CF}}(t)$ is a *closed form* control input defined as:

$$u_{i_{CF}}(t) = -\frac{\partial \varphi_i}{\partial q_{a_i}}^\mathsf{T} + v_i^* \tag{4}$$

where:

$$v_i^* = \alpha_i^\dagger \frac{\chi_i}{(1 + \chi_i^2)^{sign(\chi_i)}} \tag{5}$$

and

$$\alpha_i \triangleq (\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}} \tag{6}$$

$$\begin{aligned}
\chi_i \triangleq & \sum_{j=1}^{N} \left(\frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{o_j}} - \frac{\partial f_i}{\partial q_{o_j}} \right) \dot{q}_{o_j} \\
& + \left((\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}} \right) \\
& \cdot \sum_{h=1,h\neq i}^{M} \frac{\left(\frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_h}} - \frac{\partial f_i}{\partial q_{a_h}} \right)^\mathsf{T}}{(((\gamma_{d_h} + f_h)^{k_h} + \beta_h)^{\frac{1}{k_h}+1})} \\
& + \left(\sum_{h=1,h\neq i}^{M} \frac{1}{k_i}(\gamma_{d_i} + f_i) \sum_{j=1}^{N} \frac{\partial \beta^{(i)}}{\partial q_{a_h}} - \frac{\partial f_i}{\partial q_{a_h}} \right) v_h^*
\end{aligned} \tag{7}$$

and $(\cdot)^\dagger$ is the Moore-Penrose pseudoinverse function [25]. Function φ_i is a *navigation-like function* defined as follows

$$\varphi_i \triangleq \left(\frac{(\gamma_{d_i} + f_i)^{k_i}}{(\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)}} \right)^{\frac{1}{k_i}} \tag{8}$$

where the destinations are taken into account through

$$\gamma_{d_i}(q) \triangleq \|q - d_i\|^2 \tag{9}$$

and the obstacles are taken into account through an obstacle function (OF) defined as follows:

$$\beta^{(i)} \triangleq \prod_{j=0}^{N} \beta_{O_j}^{(i)} \prod_{\substack{h=1 \\ h \neq i}}^{M} \beta_{Ah}^{(i)} \tag{10}$$

which attains positive values when no collision occur for the agent i. OF is constructed by:

$$\beta_{Ah}^{(i)} \triangleq \frac{\|q - q_{a_h}\|^2 - (r_i + r_h)^2}{\lambda_{ih}^2 - (r_i + r_h)^2} \tag{11}$$

$$\beta_{O_j}^{(i)} \triangleq \|q - q_{o_j}\|^2 - (r_i + \rho_j)^2 \tag{12}$$

$$\beta_{O0}^{(i)} \triangleq (\rho_0 - r_i)^2 - \|q - q_0\|^2 \tag{13}$$

$$f_i \triangleq \begin{cases} 0 & \text{if } \beta^{(i)} > \varepsilon \\ \bar{f}_i(1 - \frac{3\beta^{(i)2}}{\varepsilon^2} + \frac{2\beta^{(i)3}}{\varepsilon^3}) & \text{if } \beta^{(i)} \leq \varepsilon \end{cases} \tag{14}$$

where $\bar{f}_i$ is a positive parameter which regulates the values of f_i when the agent in near an obstacle. More details about the design of u_{CF}, can be found in [21].

The control u_{OPT} is responsible for the formation objective and is derived by solving the following *constraint based programming* problem.

$$\min_{u_{OPT}, \eta} \sum_{i=1}^{M+1} h_i \eta_i$$
$$s.t. \ \dot{\varphi}_1(u_{OPT}) \leq -\min(\delta_1, \gamma_{d_1}) + \eta_1$$
$$\dot{\varphi}_2(u_{OPT}) \leq -\min(\delta_2, \gamma_{d_2}) + \eta_2$$
$$\vdots \tag{15}$$
$$\dot{\varphi}_M(u_{OPT}) \leq -\min(\delta_M, \gamma_{d_M}) + \eta_M$$
$$\dot{\psi}(u_{OPT}) \leq -\min(\delta_{M+1}, \psi) + \eta_{M+1}$$

$$\eta_i \geq 0$$
$$\|u_{iOPT}\|^2 \leq u_{i_{MAX}}^2$$

where h_i is a positive weight parameter related to the i-th task and δ_i is a positive parameter that gives an achievable upper bound for the task constraint:

$$\delta_i = \sum_{l=1}^{M} \|\frac{\partial \varphi_i}{\partial q_{a_l}}\| u_{l_{MAX}} + \sum_{j=1}^{N} \|\frac{\partial \varphi_i}{\partial q_{o_j}}\| \dot{q}_{o_j}. \tag{16}$$

with $i = \{1, 2, ..., M\}$ and

$$\delta_{M+1} = \sum_{l=1}^{M} \|\frac{\partial \psi}{\partial q_{a_l}}\| u_{l_{MAX}} \tag{17}$$

The objective function of (15) has its minimum at $\sum_{i=1}^{M+1} h_i \eta_i = 0$ i.e. when $\eta_i = 0$, $i = [1, 2, ..., M + 1]$. If the task constraints in (15) are contradicting, for example requiring a formation that is too wide to pass between two obstacles then the solution to (15) will have some nonzero η_i, with larger η_i corresponding to tasks with lower priorities.

IV. CONVERGENCE ANALYSIS

In this section, we will show that the controller of Section III above is a valid solution to Problem 1. The main result is given in Theorem 1, which in turn uses the properties of Lemma 1-2. Before stating the lemmas, we make the following definition and assumption on the free space.

Assumption 1: There exists $\tau \in \mathbb{R}_+, \tau \geq \frac{1}{\epsilon}$ such that $\beta_{O_j}^{(i)}(d_i, t) > \varepsilon, \forall j = \{1, 2, ..., N\}, \forall t \geq \tau$, i.e., after $t = \tau$ the obstacles never intersect with an ε-neighborhood around the destination points.

Theorem 1: Assume that all agents are controlled by (4) and $t \geq \tau$. Then for almost all initial collision free conditions, $\lim_{t \to \infty} q_{a_i}(t) = d_i$.

Proof: Consider

$$V = \sum_{i=1}^{M} \varphi_i \tag{18}$$

as a candidate Lyapunov function. We first show that for each i,

$$\dot{\varphi}_i \leq 0 \tag{19}$$

The derivative of each φ_i is given by:

$$\dot{\varphi}_i = \frac{\partial \varphi_i}{\partial q_{a_i}} \dot{q}_{a_i} + \sum_{j=1}^{N} \frac{\partial \varphi_i}{\partial q_{o_j}} \dot{q}_{o_j} + \sum_{\substack{h=1 \\ h \neq i}}^{M} \frac{\partial \varphi_i}{\partial q_{a_h}} \dot{q}_{a_h} \tag{20}$$

Substituting q_{a_i} in (20) with (4) we get

$$\dot{\varphi}_i = -\left\|\frac{\partial \varphi_i}{\partial q_{a_i}}\right\|^2 + \frac{(\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)} v_i^* - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}} v_i^*}{((\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)})^{\frac{1}{k_i}+1}}$$
$$+ \frac{\sum_{j=1}^{N} \frac{\partial f_i}{\partial q_{o_j}} \dot{q}_{o_j} - \frac{1}{k_i}(\gamma_{d_i} + f_i) \sum_{j=1}^{N} \frac{\partial \beta^{(i)}}{\partial q_{o_j}} \dot{q}_{o_j}}{((\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)})^{\frac{1}{k_i}+1}}$$
$$- \sum_{h=1, h \neq i}^{M} \frac{\left(\frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_h}} - \frac{\partial f_i}{\partial q_{a_h}}\right)}{(((\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)})^{\frac{1}{k_i}+1})} \cdot$$
$$\cdot \frac{\left((\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}}\right)^{\mathsf{T}}}{(((\gamma_{d_h} + f_h)^{k_h} + \beta_h)^{\frac{1}{k_h}+1})}$$
$$+ \sum_{h=1, h \neq i}^{M} \frac{\frac{\partial f_i}{\partial q_{a_h}} - \frac{1}{k_i}(\gamma_{d_i} + f_i) \sum_{j=1}^{N} \frac{\partial \beta^{(i)}}{\partial q_{a_h}}}{((\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)})^{\frac{1}{k_i}+1}} v_h^* \tag{21}$$

Now, since $(((\gamma_{d_i} + f_i)^{k_i} + \beta^{(i)})^{\frac{1}{k_i}+1})$ is positive and $-\left\|\frac{\partial \varphi_i}{\partial q_{a_i}}\right\|^2$ is negative, (2) is negative semidefinite if

$$(\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)}v_i^* - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}}v_i^*$$
$$+ \sum_{j=1}^{M} \frac{\partial f_i}{\partial q_{o_j}}\dot{q}_{o_j} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\sum_{j=1}^{M}\frac{\partial \beta^{(i)}}{\partial q_{o_j}}\dot{q}_{o_j}$$
$$- \left((\frac{\partial \gamma_{d_i}}{\partial q_{a_i}} + \frac{\partial f_i}{\partial q_{a_i}})\beta^{(i)} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_i}}\right) \quad (22)$$
$$\cdot \sum_{h=1,h\neq i}^{M} \frac{\left(\frac{1}{k_i}(\gamma_{d_i} + f_i)\frac{\partial \beta^{(i)}}{\partial q_{a_h}} - \frac{\partial f_i}{\partial q_{a_h}}\right)^\mathsf{T}}{(((\gamma_{d_h} + f_h)^{k_h} + \beta_h)^{\frac{1}{k_h}+1})}$$
$$+ \left(\sum_{h=1,h\neq i}^{M} \frac{\partial f_i}{\partial q_{a_h}} - \frac{1}{k_i}(\gamma_{d_i} + f_i)\sum_{j=1}^{N}\frac{\partial \beta^{(i)}}{\partial q_{a_h}}\right)v_h^*$$

is negative semidefinite as well.

Replacing α_i with its expression in (6) and χ_i as in (7) we can derive that (22) is equal to

$$\alpha_i v_i^* - \chi_i \quad (23)$$

and is semidefinite negative when

$$\alpha_i v_i^* \leq \chi_i. \quad (24)$$

The control input (4) satisfies (24), therefore (19), since

$$\frac{\chi_i}{(1 + \chi_i^2)^{sign(\chi_i)}} \leq \chi_i \; \forall \chi_i \in \mathbb{R} \quad (25)$$

hence the derivative of the Lyapunov function candidate is semidefinte negative:

$$\dot{V} = \sum_{i=1}^{M} \dot{\varphi}_i \leq 0. \quad (26)$$

By assumption, the agents satisfy the formation constraint at the destinations. Thus by satisfying the convergence objective we will also satisfy the formation constraint. From (26) every level set of (18) is a positively invariant set which is also closed and bounded due to the properties of the potentials φ_i. By LaSalle's invariance principle, the system converges to the largest invariant subset of the set $\dot{V} = 0$. The control design implies that $\dot{V} = 0$ is equivalent to $\dot{\varphi}_i = 0$ for all i. From assumption 1, and under some additional assumptions mentioned in [21], it can be shown that the largest invariant set of the set $\dot{\varphi}_i = 0, \forall i$ is the set of desired destination points. ∎

Lemma 1: The system under the control u_i of (3) is collision free.

Proof: According to NF's definition above, $\varphi_i = 1$ on the boundaries of its domain. But in the controller we have $\dot{\varphi}_i \leq 0$, and since by assumption $\varphi_i(q_i(t_0)) < 1$, no collisions occur. ∎

Lemma 2: If $\epsilon > 0$ and we apply the control u_i of (3), then $q_i \to d_i$.

Proof: Note that in (3), we only apply $u_{i_{OPT}}$ as long as $\varphi_i(t) \leq \varphi_i(0) - t\epsilon$, $\forall i$. The codomain of each φ_i is $[0,1]$, thus, after $t = 1/\epsilon$ we have either reached the destination, or we will never use $u_{i_{OPT}}$ again. According to Lemma 1, $u_{i_{CF}}$ will now achieve $q_{a_i} \to d_i$ in the latter case. ∎

Remark 1: The controller (3) is a solution to Problem 1. In particular, convergence is provided due to Lemma 2. Formation maintenance is constantly improved since $\dot{\psi} \leq 0$ in (15). But contradicting objectives might yield $\eta_{M+1} \neq 0$. In that case convergence to d_i by Lemma 2 still guarantees $\psi \to 0$ by assumption. Collision avoidance is guaranteed by Lemma 1.

Remark 2: The controller (4) may be rendered unbounded whenever the robots get too close to moving obstacles. This can be regulated by assuming upper bounds on the velocities of the obstacles.

V. SIMULATIONS

To demonstrate the convergence and stability of the solution, four numerical solution, three in a static environment and one in a dynamic environment are presented, simulated by MATLAB scripts using an active set optimizer algorithm chosen due to the non smooth constraints. In the simulations the agents have to navigate maintaining a triangle shape formation with $\lambda_{ij} = 0.08$ in a workspace with radius 1. Figure 2(a), 2(b) 3(a) and 3(b) show qualitatively the solution proposed, Figures 4, 5 and 7 show the OFs and FFs in the different simulations giving quantitative information regarding the collision avoidance and formation constraints.

	Simulation I	Simulation II	Simulation III	Simulation IV
h	$h_1 = 10^1$ $h_2 = 10^1$ $h_3 = 10^1$ $h_4 = 10^5$	$h_1 = 10^2$ $h_2 = 10^2$ $h_3 = 10^2$ $h_4 = 10^2$	$h_1 = 10^3$ $h_2 = 10^3$ $h_3 = 10^3$ $h_4 = 10^{-1}$	$h_1 = 10^{-1}$ $h_2 = 10^{-1}$ $h_3 = 10^{-1}$ $h_4 = 10^3$
ϵ	0	10^{-4}	10^{-4}	0
	Solution 2	-	Solution 1	Solution 2

TABLE I. Simulation Parameters

As can be seen from the parameters of Table I, simulation III correspond to Solution 1, with focus on convergence to the destination, simulations I and IV correspond to Solution 2, with focus on formation keeping, and simulation II give equal weights to formation maintenance and goal convergence. The remaining simulation's parameters are collected in Table II and Figures 2 and 3 show the path executed for each simulation as lines of different colors. The agents are the blue, green and magenta circles, the obstacles are the red circles, and some intermediates agents' position are showed as faded circles.

In simulation I, a higher priority is given to the formation keeping, making the agents quickly reach their desired formation, as can be seen from the formation function in Figure 4. When the goal configuration is reached, the obstacle function for each agent reaches a fixed value due to the fact that the obstacles are fixed in the workspace.

In simulation II, an obstacle is moving in the workspace and is heading towards the agents. These avoid the oncoming collision, while maintaining the desired formation, and reach their goal configuration. As can be seen from Figure 5, the convergence to perfect formation is somewhat slower than in Figure 4, and a small error remains when passing the obstacle. After passing, the obstacle distances increase

	Simulation I			Simulation II			Simulation III & IV		
	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
$q_a(t_0)$	$\begin{bmatrix}-0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0\\-0.16\end{bmatrix}$	$\begin{bmatrix}0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0\\-0.16\end{bmatrix}$	$\begin{bmatrix}-0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0.06\\-0.23\end{bmatrix}$	$\begin{bmatrix}0\\-0.16\end{bmatrix}$
r	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015
d	$\begin{bmatrix}-0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0\\0.64\end{bmatrix}$	$\begin{bmatrix}-0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0\\0.64\end{bmatrix}$	$\begin{bmatrix}-0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0.04\\0.57\end{bmatrix}$	$\begin{bmatrix}0\\0.64\end{bmatrix}$
k	80	80	80	60	60	60	120	120	120
$\bar{f}$	1	1	1	2	2	2	1	1	1

	Obstacle 1	Obstacle 2	Obstacle 1	Obstacle 2	Obstacle 1	Obstacle 2
$q_o(t_0)$	$\begin{bmatrix}0.06\\0\end{bmatrix}$	$\begin{bmatrix}-0.1\\0.2\end{bmatrix}$	$\begin{bmatrix}0\\0.64\end{bmatrix}$	$\begin{bmatrix}-0.04\\0\end{bmatrix}$	$\begin{bmatrix}-0.5\\0.1\end{bmatrix}$	$\begin{bmatrix}0.5\\0.1\end{bmatrix}$
$\dot{q}_o(t_0)$	0	0	$\begin{bmatrix}0.1sin(10t)\\0.1cos(-10^{-4}t)\end{bmatrix}$	0	0	0
ρ	0.05	0.05	0.05	0.3	0.48	0.48

TABLE II.   Agents and Obstacles' Parameters
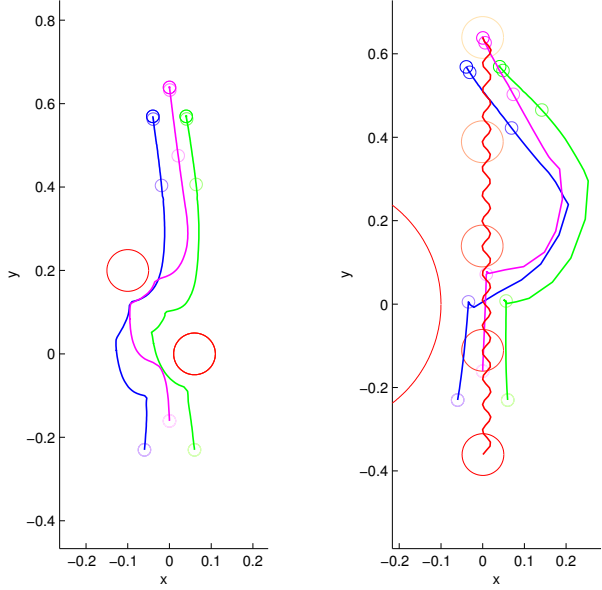
(a) Paths executed in simulation I

(b) Path executed in simulation II

Fig. 2.   Paths executed for simulation I and II
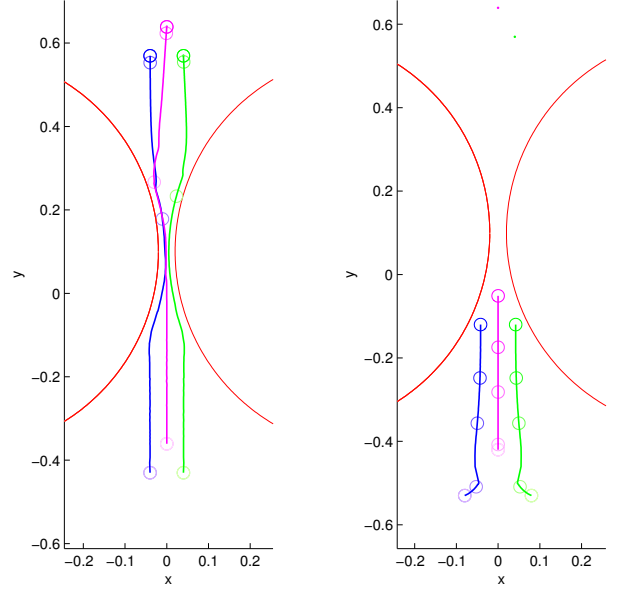
(a) Path executed in simulation III

(b) Path executed in simulation IV

Fig. 3.   Paths executed for simulation III and IV

Fig. 4.   Obstacle and formation function behaviors for simulation I

linearly when the obstacle continues away from the agents. Simulation III shows an example in which the agents can not reach the goal without breaking the formation. Thus, the priority weights come into play. Due to the higher priority on the navigation (Solution 1), $h_j \gg h_4$, $j = 1, \ldots, 3$ the formation constraint is violated in order to reach the goal. Thus, as can be seen in Figure 7 the formation function increases for some time while passing in-between the two obstacles. Then the desired formation is once again reached.

Simulation IV on the other hand, uses Solution 2 with a higher priority of the formation $h_4 \gg h_j$, $j = 1, \ldots, 3$ and $\epsilon = 0$. Here, the formation function decreases to zero, keeping the formation, while progress towards the goal is stopped. Still, collisions are avoided, as can be seen in the positive obstacle function. It should be noted that none of the simulations resulted in a switch of the controller in (3).
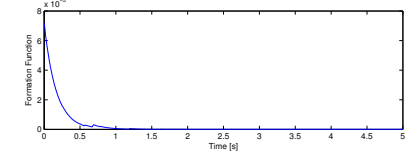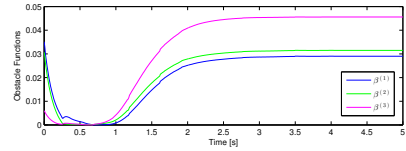
VI. CONCLUSIONS

Combining navigation-like functions with constraint based programming, we have shown how to achieve reactive navigation in formation, with provable properties in terms of goal convergence and formation maintenance. To have an elegant proof of convergence, we introduce a limit time in which the prioritized problem has to be executed, this time limit can
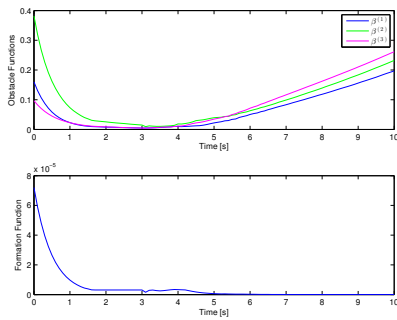
Fig. 5. Obstacle and formation function behaviors for simulation II
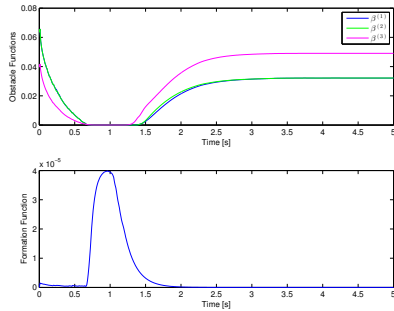

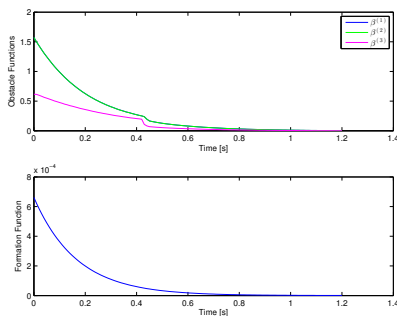Fig. 6. Obstacle and formation function behaviors for simulation III

Fig. 7. Obstacle and formation function behaviors for simulation IV

be arbitrarily extended without loss of generality, reducing the time in which the closed form controller is activated. By prioritizing different tasks, which might be in conflict with each other, tradeoffs can be made in a consistent way.

ACKNOWLEDGMENT

REFERENCES

[1] Q. Wang, H. Fang, J. Chen, Y. Mao, and L. Dou, "Flocking with obstacle avoidance and connectivity maintenance in multi-agent systems," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE, 2012, pp. 4009–4014.

[2] D. V. Dimarogonas and K. J. Kyriakopoulos, "A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems," *Automatica*, vol. 44, no. 10, pp. 2648 – 2654, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109808002343

[3] L. Guys, S. Puechmorel, and L. Lapasset, "Automatic conflict solving using bi-harmonic navigation functions," in *15th meeting of the Euro Working Group on Transportation*, September 2012.

[4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[5] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Transactions on Industrial Informatics*, 2012.

[6] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2492–2497.

[7] W. Yu, G. Chen, and M. Cao, "Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities," *Systems & Control Letters*, vol. 59, no. 9, pp. 543–552, 2010.

[8] F. Cucker and J.-G. Dong, "A general collision-avoiding flocking framework," *Automatic Control, IEEE Transactions on*, vol. 56, no. 5, pp. 1124–1129, 2011.

[9] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, no. 4, pp. 412–442, Dec. 1990. [Online]. Available: http://dx.doi.org/10.1016/0196-8858(90)90017-S

[10] P. Ögren and J. Robinson, "A model based approach to modular multi-objective robot control," *Journal of Intelligent and Robotic Systems*, vol. 63, pp. 257–282, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10846-010-9523-7

[11] C. Samson, M. Le Borgne, and B. Espiau, *Robot control: the task function approach*. Clarendon Press Oxford, England, 1991, vol. 22.

[12] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *Robotics and Automation, IEEE Transactions on*, vol. 5, no. 4, pp. 472–490, 1989.

[13] Z. Peng and N. Adachi, "Compliant motion control of kinematically redundant manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 6, pp. 831–836, 1993.

[14] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[15] R. Smits, T. Laet, K. Claes, H. Bruyninckx, and J. Schutter, "itasc: A tool for multi-sensor integration in robot manipulation," *Multisensor Fusion and Integration for Intelligent Systems*, pp. 235–254, 2009.

[16] P. Ögren, "Improved predictability of reactive robot control using control lyapunov functions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2008*. IEEE, 2008, pp. 1274–1279.

[17] P. Ögren, C. Smith, Y. Karayiannidis, and D. Kragic, "A multi objective control approach to online dual arm manipulation," in *International IFAC Symposium on Robotic Control*, 2012.

[18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

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

[20] D. Dimarogonas, S. Loizou, and K. Kyriakopoulos, "Multirobot navigation functions ii: towards decentralization," *Stochastic Hybrid Systems*, pp. 209–253, 2006.

[21] M. Colledanchise, "Stabilization and collision avoidance of non-point agents in dynamic environments: A potential field approach." Master's thesis, KTH, Automatic Control, 2012.

[22] M. C. De Gennaro and A. Jadbabaie, "Formation control for a cooperative multi-agent system using decentralized navigation functions," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.

[23] I. F. Filippidis and K. J. Kyriakopoulos, "Navigation functions for everywhere partially sufficiently curved worlds," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2115–2120.

[24] R. Romera, J. E. Valdés, and R. I. Zequeira, "Active-redundancy allocation in systems," *Reliability, IEEE Transactions on*, vol. 53, no. 3, pp. 313–318, 2004.

[25] E. H. Moore, "On the reciprocal of the general algebraic matrix," *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395.