

Swarm Coordination Based on Smoothed Particle Hydrodynamics Technique

Luciano C. A. Pimenta, Guilherme A. S. Pereira, *Member, IEEE*, Nathan Michael, *Member, IEEE*, Renato C. Mesquita, Mateus M. Bosque, Luiz Chaimowicz, *Member, IEEE*, and Vijay Kumar, *Fellow, IEEE*

Abstract—The focus of this study is on the design of feedback control laws for swarms of robots that are based on models from fluid dynamics. We apply an incompressible fluid model to solve a pattern generation task. Possible applications of an efficient solution to this task are surveillance and the cordoning off of hazardous areas. More specifically, we use the smoothed-particle hydrodynamics (SPH) technique to devise decentralized controllers that force the robots to behave in a similar manner to fluid particles. Our approach deals with static and dynamic obstacles. Considerations such as finite size and nonholonomic constraints are also addressed. In the absence of obstacles, we prove the stability and convergence of controllers that are based on the SPH method. Computer simulations and actual robot experiments are shown to validate the proposed approach.

Index Terms—Distributed robot systems, motion control, smoothed particle hydrodynamics (SPH), swarm robotics.

I. INTRODUCTION

THE use of large groups of robots in the execution of complex tasks has received much attention in recent years. Generally called *swarms*, these systems employ a large number of simple agents to perform different types of tasks, often inspired by their biological counterparts. There are some common features and requirements in the characterization of robotic swarms. First of all, systems and algorithms to control and coordinate swarms of robots must be scalable from tens to hundreds of agents and must be robust to the dynamic deletion or addition of new agents. Agents should operate asynchronously and rely

only on local sensing and communication, as the maintenance of a global state of the system is impractical. Furthermore, robots must be *anonymous* due to the challenges of uniquely identifying individual members within the swarm.

In this paper, we focus on swarm navigation and the problem to design decentralized feedback control laws that enable a swarm of robots to control to a desired pattern or shape in complex environments while avoiding collisions between robots and the environments. Several robotic applications such as surveillance, manipulation, and boundary monitoring can be addressed by our proposed methodology [1].

More specifically, we develop feedback control laws that are based on models from fluid dynamics to control swarms of robots. The main motivation for this study stems from the fact that a great variety of characteristics desirable for a group of robots can be observed in fluids. Some examples of such characteristics are 1) fluids are easily deformed, 2) fluids can easily contour around objects, and 3) the flow field and fluid phase variables can be easily manipulated in order to design the desired behaviors. As will be discussed in this paper, our solution considers a smoothed particle hydrodynamics (SPH) model for an incompressible fluid in the presence of external forces.

This paper is organized as follows. The next section discusses some related work in the field. Section III formally defines the problem, while Sections IV and V present our proposed solution. Simulations and experimental results are presented in Sections VI and VII, respectively. Finally, we conclude in Section VIII and propose possible avenues for future work.

II. RELATED WORK

The general area of motion planning and control for large groups of robots has been very active over the past few years. One of the first works that deal with the motion control of a large number of agents was proposed to generate realistic computer animations of flocks of birds (called *boids*) [2], where local interactions among neighboring agents create an emergent behavior for the whole flock.

In robotics, these interactions can be considered as a special case of the potential field approach [3], [4], in which robots are attracted by the goal and repelled by obstacles and other robots by virtual forces. Generally, attractive forces are modeled through the gradient descent of specific functions. By projecting these functions, it is possible to make the robots converge to specific regions in the workspace and even design specific shapes and patterns. In [5], for example, implicit functions and gradient descent techniques were used to allow swarms of robots to synthesize shapes and patterns in obstacle-free environments.

Manuscript received July 12, 2012; accepted December 5, 2012. Date of publication January 9, 2013; date of current version April 1, 2013. This paper was recommended for publication by Associate Editor K. Kyriakopoulos and Editor G. Oriolo upon the evaluation of the reviewers comments. This work was supported in part by Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Brazil) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (Brazil). The work of R. C. Mesquita, G. A. S. Pereira, and L. Chaimowicz was supported by a CNPq Scholarship.

L. C. A. Pimenta, G. A. S. Pereira, R. C. Mesquita, and M. M. Bosque are with the Engineering School, Universidade Federal de Minas Gerais, Belo Horizonte, MG, 31270-901, Brazil (e-mail: luciano.pimenta@gmail.com; gpereira@ufmg.br; renato@cpdee.ufmg.br; mateusmb@gmail.com).

N. Michael is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: nmichael@cmu.edu).

L. Chaimowicz is with the Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-010, Brazil (e-mail: chaimo@dcc.ufmg.br).

V. Kumar is with the Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104 USA, on leave at the White House Office of Science and Technology Policy, Washington, DC 20500 USA (e-mail: kumar@cis.upenn.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2012.2234294

Some works also consider the case in which robots will perform some orbits or move within the target, as shown in [6].

The methodology that is presented in this paper was built on the main ideas proposed in [7] and [8], where we used the SPH model for fluid simulation to control robotic agents of a swarm in such a way that the group behaves as a fluid. This study builds on [8], whose methodology has already been successfully applied by some authors in applications such as the coordination of aerial sensor networks [9], trajectory following [10], and swarm shape control [11]. The main differences between this study and the conference paper [8] are that here we provide further details of our theoretical results and also propose a new methodology to address dynamic obstacles in the environment. We also present new simulations to evaluate robustness to localization errors and scalability along with new experimental results with a group of 12 differential drive robots.

The idea to mimic the behavior of gases, fluids, and solids has been previously explored by some researchers. Spears *et al.* [12] proposed a framework called *Physicomimetics* to control swarms of robots that are based on these behaviors for tasks such as exploration and obstacle avoidance [13] and surveillance [14]. Shimizu *et al.* [15] also explored similar ideas considering robots as particles in a fluid using Stokesian dynamics to model the behavior of the system. These methods inspired models that have also been used in other fields such as dense crowd simulation [16], amenity space design [17], and cloth simulation [18].

In [19], the use of SPH to control a swarm of robots is proposed. They considered a task of maximum sensor coverage in a 2-D environment that was filled with obstacles and objects of interest. In [19], the SPH equations for compressible fluids were used to mimic the behavior of air at 20°C. In parallel to our research, in [20], the use of the SPH method to model a robotic swarm as a fluid and control its flow by tuning the flow parameters is also proposed. The authors present simulations of coverage, dispatching through waypoints, and flocking.

Different from [19] and [20] and other previous work in this area, we propose a methodology to compute external fluid forces that are based on harmonics functions. By using the finite-element method (FEM) in this computation, static obstacles of generic geometries may be modeled. Additionally, we choose to model the swarm as an incompressible fluid (where [19] discusses compressible fluid models) in order to favor cohesive ensemble motion. We also address issues that are related to the finite size and nonholonomic constraints associated with the vehicles employed in the experimentation.

In this study, we consider the pattern generation problem and design controllers that drive the swarm to converge to a desired shape. Several robotic applications such as surveillance, manipulation, and boundary monitoring can be addressed by our proposed methodology. For example, in [21], a flock of aerial vehicles are used to track a chemical plume that is released in the atmosphere.

SPH shares several advantageous properties with existing approaches that are detailed in the literature for controlling swarms of robots while also providing unique capabilities. The approach is decentralized, and thus scalable, as it only requires information from robots within a local neighborhood. Although this

characteristic is common to other works [4], [22], the SPH method provides an effective way to control the density of the robots (also observed in [11]). This property makes the method suitable to control swarms in rigid or nonrigid formations to a specified target location or region. SPH may also be comparable with some leader-following approaches, as it is possible to include a negative density particle that acts as a “virtual leader,” as suggested in [10]. Real-time obstacle avoidance is enabled by the introduction of large density particles at the borders of a detected obstacle. Finally, the SPH method may be applicable to approaches that build upon high-level abstractions to navigate groups of robots [23], [24] by providing a mechanism to enable collective motion while controlling interrobot interactions.

III. PATTERN GENERATION PROBLEM

The so-called *pattern generation problem* may be stated as follows.

Problem 1: Given N robots and any initial configuration, the geometry of the environment with static obstacles defining a compact domain $\Omega \subset \mathbb{R}^2$, and the image of a curve $\Gamma \subset \Omega$, find a decentralized controller which enables the robots, without colliding with static obstacles and each other, to form the pattern described by Γ .

Two different robot models should be considered. The first one is the simplified fully actuated, holonomic, point robot model:

$$\ddot{\mathbf{q}}_i = \mathbf{u}_i$$

where $\ddot{\mathbf{q}}_i$ is the robot’s acceleration, and \mathbf{u}_i is the control input. The second is the finite-size (circular with radius R), nonholonomic, robot model:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}$$

where the robot configuration is determined by x, y , and orientation θ ; the forward velocity is denoted by v ; and the angular velocity is denoted by ω .

It may also be desirable to maintain the group connectivity during the task execution. By connectivity, we mean the ability of each agent to send and receive information from other agents. Since we are considering only decentralized strategies, this is possible only if the agents maintain a minimum distance from each other. To simplify the problem, we make the following assumptions.

Assumption 1: The number of robots is sufficiently large to form the desired pattern and, at the same time, sufficiently small to guarantee enough space along the curve for all of the robots.

Assumption 2: Global localization and the complete map of the environment is available to every agent.

Assumption 3: The range and bearing to and velocity of neighboring robots within a distance D are available to the agents. This distance D is small in comparison to the size of the environment, which implies that robot i is not generally able to detect the entire group.

The pattern generation problem becomes infeasible if Assumption 1 is not verified. Our interest in this study is to be able to visually recognize the pattern that is based on the presence of the robots along the curve. However, one could formalize the problem differently in terms of pattern sampling. In this way, it is possible to define the minimum number of required robots by applying the Nyquist–Shannon sampling theorem to the curve parametric equations. We require Assumption 2 as state estimation and mapping are outside of the scope of this study. As robots in general have communication hardware, we believe Assumption 3 is reasonable in many scenarios. The required neighbor information (range, bearing, and velocity) for the devised controller may then be available by means of communication. We now discuss the proposed methodology to solve the formulated problem.

IV. SMOOTHED PARTICLE HYDRODYNAMICS

SPH is a *mesh-free particle* numerical method which was originally introduced in [25] and [26] to solve problems in astrophysics. It is a particle numerical method since it employs a finite set of disordered discrete particles to represent the state of the simulated system. By disordered discrete particles, we mean that it is not necessary to have particles organized according to a fixed order, as nodes in a grid for example. It is mesh-free due to the fact that it is not necessary to generate a mesh to represent the connectivity of the particles as in the case of the FEM. SPH is considered as a *Lagrangian* method, which means that the particles are not fixed in space while the material is moving. The particles are associated with the material and move with the flow. Due to all of these characteristics, this method is extensively used to solve fluid dynamics problems [27], where issues such as large deformation, moving interfaces between different materials, moving boundaries, and free surfaces appear often. Such issues can be difficult to address with other numerical methods using finite elements and differences.

SPH is based on the integral representation of a function

$$f(\mathbf{q}) = \int_{\Upsilon} f(\mathbf{q}') \delta(\mathbf{q} - \mathbf{q}') d\mathbf{q}' \quad (1)$$

where Υ is the volume that contains \mathbf{q} , and $\delta(\mathbf{q} - \mathbf{q}')$ is the Dirac delta function.

If the Delta function is replaced by a smoothing function $W(\mathbf{q} - \mathbf{q}', h)$, then the integral representation is approximated by

$$f(\mathbf{q}) \approx \langle f(\mathbf{q}) \rangle = \int_{\Upsilon} f(\mathbf{q}') W(\mathbf{q} - \mathbf{q}', h) d\mathbf{q}' \quad (2)$$

where $\langle f(\mathbf{q}) \rangle$ is an approximation of $f(\mathbf{q})$, W is the so-called *smoothing kernel function* or simply *kernel* in the SPH literature, and h is the *smoothing length* that defines the influence area of W .

In order to guarantee accuracy in the numerical simulation, the kernel is chosen to satisfy the following properties:

$$\begin{aligned} \int_{\Upsilon} W(\mathbf{q} - \mathbf{q}', h) d\mathbf{q}' &= 1 \\ \lim_{h \rightarrow 0} W(\mathbf{q} - \mathbf{q}', h) &= \delta(\mathbf{q} - \mathbf{q}'). \end{aligned} \quad (3)$$

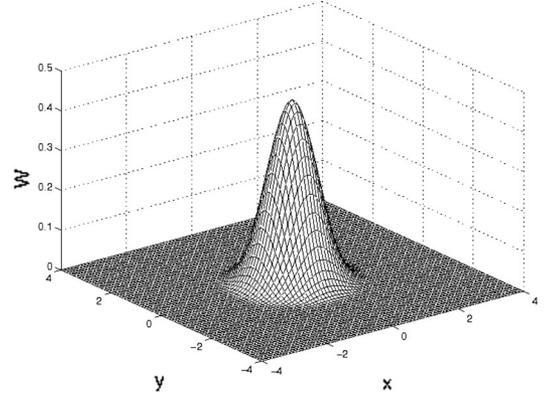


Fig. 1. Graphic of the kernel function W with $h = 1$.

The smoothing function is generally chosen to be smooth (as its gradient is needed in most applications), nonnegative, even, and monotonically decreasing with the increase of distance from the particle. Furthermore, it is chosen to have compact support that is controlled by the parameter h in order to avoid the necessity to compute integrals over the whole solution domain. The spline kernel satisfies all of these properties and has been used successfully for years by researchers in the numerical methods community. In this study, we use 2-D cubic splines:

$$W(\mathbf{q}, h) = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}\kappa^2 + \frac{3}{4}\kappa^3, & \text{if } 0 \leq \kappa \leq 1 \\ \frac{1}{4}(2 - \kappa)^3, & \text{if } 1 \leq \kappa \leq 2 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\kappa = \|\mathbf{q}\|/h$. It can be observed that the function support is determined by $2h$. Fig. 1 shows the appearance of this function when it is centered at the origin with $h = 1$.

The continuous integral in (2) can be converted to a summation over the N particles in the support domain of \mathbf{q} by considering that a particle j has a finite volume ΔV_j , which is related to the mass m_j , of the particle by

$$m_j = \Delta V_j \rho_j \quad (5)$$

where ρ_j is the density of the particle. Thus, using ΔV_j instead of $d\mathbf{q}'$:

$$\langle f(\mathbf{q}) \rangle \approx \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{q}_j) W(\mathbf{q} - \mathbf{q}_j, h). \quad (6)$$

The error in approximating the integral representation of a function by the summation of the function evaluated at particle locations weighted by interpolation kernels depends on the disorder of the particles and is normally $O(h^2)$ or better [28].

Spatial derivatives of f , such as the gradient, can also be approximated. If integration by parts is used in the simplification process, it is possible to write the spatial derivative of f in terms of the gradient of the kernel

$$\langle \nabla_{\mathbf{q}} f(\mathbf{q}) \rangle \approx \sum_{j=1}^N \frac{m_j}{\rho_j} f(\mathbf{q}_j) \nabla_{\mathbf{q}} W(\mathbf{q} - \mathbf{q}_j, h) \quad (7)$$

where $\nabla_{\mathbf{q}}$ is the gradient taken with respect to \mathbf{q} .

It is interesting to observe that the particle approximation in (6) and (7) introduces mass and density into the equations. Since density is a key variable in hydrodynamic problems, this particle approximation can be conveniently applied in such problems. According to [27], this is a motivating reason for the popularity of the SPH method for fluid dynamics problems. There are three continuum governing equations of fluid dynamics: 1) conservation of mass, 2) conservation of momentum, and 3) conservation of energy. For *inviscid compressible fluids*, in the absence of heat flux, these equations, in the Lagrangian description, are given by

$$\frac{D\rho}{Dt} = -\rho\nabla \cdot \mathbf{v} \quad (8)$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{\nabla P}{\rho} \quad (9)$$

$$\frac{De}{Dt} = -\left(\frac{P}{\rho}\right)\nabla \cdot \mathbf{v} \quad (10)$$

where \mathbf{v} is velocity $D\mathbf{q}/Dt$, P is the hydrostatic pressure, and e is the internal energy per unit of mass. The operator D/Dt is the *total time derivative* that is physically the time rate of change following a moving fluid element. This derivative is composed of two factors: 1) the time fluctuation of the flow property itself and 2) the variation of the flow property due to the movement of the fluid element. Mathematically, this is written as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla). \quad (11)$$

An additional *equation of state* must be used to fully characterize a fluid. For many compressible fluids, the model of an ideal gas can be used. In this case, the following equation of state can be applied:

$$P = (\gamma - 1)\rho e \quad (12)$$

where γ is the ratio of specific heats, a parameter which depends on the gas being simulated.

In the SPH method, the continuum equations of fluid dynamics are converted into a set of ordinary differential equations, where each one controls the evolution of an attribute of a specific particle. There are several variants of the SPH particle equations. Each variant is obtained by the application of different identities, algebraic manipulation, and the particle approximation scheme in the continuum equations. Next, we show a well-known version of SPH which has been successfully used in fluid dynamic simulations for years. It leads to higher accuracy due to its symmetry, i.e., there is a symmetric force between pairs of particles. The complete derivation is found in [27]

$$\rho_i = \sum_j m_j W(\mathbf{q}_i - \mathbf{q}_j, h) \quad (13)$$

$$\frac{d\mathbf{v}_i}{dt} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} + \mathbf{f}_i \quad (14)$$

$$\frac{de_i}{dt} = \frac{1}{2} \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \mathbf{v}_{ij} \cdot \nabla_i W_{ij} \quad (15)$$

where $W_{ij} = W(\mathbf{q}_i - \mathbf{q}_j)$, $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and \mathbf{f}_i is the sum of external forces that are normalized by the mass m_i . It should be clear that e_i is only a component of the total energy of the system. The total energy also takes into account the kinetic energy and the energy that is related to the external forces. The operator d/dt is the ordinary time derivative. As the SPH method is a Lagrangian approach, D/Dt is equivalent to d/dt in the particle equations. The term Π_{ij} is an artificial viscosity term that is usually added to allow shock wave simulation and to avoid particle penetration, i.e., to prevent particles from occupying the same position. There are several variants of this viscosity term with the most common variation given by [28]:

$$\Pi_{ij} = \begin{cases} \frac{1}{\bar{\rho}_{ij}}(-\xi_1 \bar{c}_{ij} \mu_{ij} + \xi_2 \mu_{ij}^2), & \text{if } \mathbf{v}_{ij} \cdot \mathbf{q}_{ij} < 0 \\ 0, & \text{if } \mathbf{v}_{ij} \cdot \mathbf{q}_{ij} > 0 \end{cases} \quad (16)$$

where

$$\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|^2 + \eta^2}. \quad (17)$$

In (16), $\bar{\rho}_{ij}$ is the average between the densities of particles i and j , ξ_1 and ξ_2 are viscosity constants, \bar{c}_{ij} is the average speed of sound, and η^2 is a term added to avoid singularities. The term η^2 should be small enough to avoid severe smoothing of the viscosity term. Usually, this term is made equal to $0.01h^2$.

The speed of sound of a particle i , which represents the speed at which sound travels through the fluid element represented by the particle, is given by

$$c_i = \sqrt{\frac{\gamma P_i}{\rho_i}}. \quad (18)$$

The motion of *incompressible fluids*, such as water, can also be simulated using the SPH equations. The key idea is to make a compressible fluid behave like a nearly incompressible one. This can be done by employing the equation of state below [29]:

$$P_i = B_i \left[\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right] \quad (19)$$

where ρ_0 is the reference density (1000 kg/m³ in the case of water), and B_i is the bulk modulus.¹ The bulk modulus is a property which characterizes the compressibility of the fluid. When simulating incompressible fluids by means of the SPH method, the bulk modulus is computed to guarantee a small Mach number,² M (typically 0.1–0.01). The following expression may be used [29]:

$$B_i = \left(\frac{\|\mathbf{v}\|_{\max}}{M} \right)^2 \rho_i \quad (20)$$

where $\|\mathbf{v}\|_{\max}$ is the maximum velocity of the flow. For liquids, the speed of sound of a particle i , which represents the speed at which sound travels through the fluid element represented by

¹The bulk modulus may be expressed by $B = -V\partial P/\partial V$.

²The Mach number is given by v/c , where v is the speed of an object moving through the fluid, and c is the speed of sound in the fluid.

the particle, is given by

$$c_i = \sqrt{\frac{B_i}{\rho_i}}. \quad (21)$$

In [30], the bulk modulus is computed via

$$B_i = \frac{200\rho_i gH}{\gamma} \quad (22)$$

where H is the maximum fluid depth, and g is the gravitational constant. The speed of sound is also adapted in [30]

$$c_i = \sqrt{\frac{\gamma(P_i + B_i)}{\rho_i}}. \quad (23)$$

One should observe that when (19) is used, large values of pressure are necessary to change density. This is the effect that endows the system with the desired behavior of incompressible fluids.

In a usual SPH simulation, the differential equations that are presented before are integrated over time by means of finite-difference methods [30]. In this study, we use the SPH technique to control swarms of robots. This is done by considering each robot as a particle and, in this case, the actual movement of the robots is responsible for the time integration.

V. PATTERN GENERATION TASK SOLUTION

We treat each robot of the team as an SPH particle that is subjected to an external force and now discuss the derivation of decentralized control laws that are based on the SPH equations. The resulting controllers only require the knowledge of the gradient of a potential function at the location of robot i , the position and velocity of the robot i itself and of its robots, and the SPH mass and density parameters associated with neighboring robots. For a robot i with configuration $\mathbf{q}_i = [x_i, y_i]^T$, we define \mathcal{N}_i as the set of robots in the neighborhood of robot i :

$$\mathcal{N}_i = \{j \neq i \mid \|\mathbf{q}_j - \mathbf{q}_i\| < D\} \quad (24)$$

where the distance D is determined by the kernel support size, which in the case of the kernel in (4) is given by $D = 2h$.

Our approach consists of two steps. In the first step, we compute a global potential function. This potential function is responsible to drive the robots to the desired pattern. The second step consists of controlling the robots by using control laws that are based on the SPH equations. These equations provide interaction forces among the agents of the group. Sections V-A and B describe each of these steps, respectively.

A. Global Potential Functions

Our approach relies on the computation of a global potential function. In this section, we present two examples of such functions: *harmonic functions* [7] and *shape functions* [31]. As detailed in [32] and [33], harmonic functions are free of local minima. Although saddle points may exist, for this study, they are not an issue as any perturbation will drive the system away from them. The main advantage of using harmonic functions

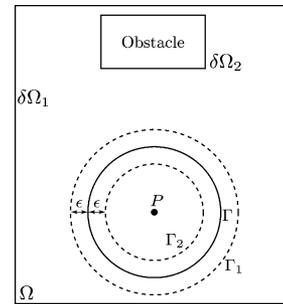


Fig. 2. Domain example.

is the possibility of computing them numerically in a computationally efficient manner, even in the case of environments with complex geometry. The computation is done by using the FEM in the solution of Laplace's equation. The efficiency of the FEM is because of its ability to work properly with unstructured meshes of elements which are used to exactly decompose the solution domain. Further details can be found in our previous work [7], [34]. In Section V-D, we show stability and convergence when considering shape functions in obstacle-free environments.

1) *Harmonic Functions*: If a safety factor ϵ is defined such that the desired pattern is represented by a region between two curves Γ_1 and Γ_2 , we can define a harmonic function which drives the robots toward the goal region and, at the same time, drives the robots away from the obstacles. If the desired pattern Γ is parameterized by a function $s(x, y) = 0$, then Γ_1 is such that $s(x, y) = \epsilon$, and Γ_2 is such that $s(x, y) = -\epsilon$. Fig. 2 presents an example of a domain with an obstacle and a circular pattern with a safety factor added. In practice, this safety factor provides a means to capture state and environment uncertainty in the definition of the feedback control laws.

Harmonic functions are solutions to Laplace's equation. In order to guarantee uniqueness of the solution, we must define boundary conditions. We use constant Dirichlet boundary conditions such that a maximum value is obtained at the boundaries of the configuration space, and a minimum value is obtained at the desired pattern. This boundary value problem is given by

$$\begin{cases} \nabla^2 \phi = 0 \\ \phi(\Gamma_1) = \phi(\Gamma_2) = 0 \\ \phi(\partial\Omega_1) = \phi(\partial\Omega_2) = \phi(P) = V_c \end{cases} \quad (25)$$

where ϕ is the harmonic function, V_c is a positive constant, and P is a point that is defined inside the pattern in the case of closed curves to guarantee convergence from the interior of the pattern (see Fig. 2).

By using the boundary conditions that are presented in (25), we guarantee convergence of the integral curves of $-\nabla\phi$ to the desired pattern. Therefore, a robot that follows the field lines defined by $-\nabla\phi$ reaches the target (the region delimited by Γ_1 and Γ_2) in a finite time without any collision with obstacles.

Remark 1: The computation in (25) can be seen as the solution of an analogous electrostatic problem that considers a homogeneous isotropic medium in the absence of charge density [35].

In this case, ϕ corresponds to the scalar electric potential and $-\nabla\phi$ corresponds to the electric field.

Remark 2: Since the harmonic functions we compute here are designed to satisfy the properties of navigation functions (see [33]), this first step of our approach may be replaced by any other navigation function computation method.

2) *Shape Functions:* We now consider shape functions that describe star-shaped sets in obstacle-free environments. As defined in [36], star-shaped sets are those that are characterized by the possession of a distinguished “center point” $\hat{\mathbf{q}}$ from which all rays cross their boundary once and only once. Star-shaped sets are topologically equivalent to disks.

According to [31], given a desired pattern Γ , a shape function ϕ is a positive semidefinite function with a minimum value equal to 0 at the boundary Γ . For a desired curve parameterized by a function $s(x, y) = 0$, we have

$$\phi = s(x, y)^2 \quad (26)$$

as a candidate shape function. As proposed in [31], we will use $\phi = s(x, y)^2$ such that we have the following.

- 1) $s(x, y)$ is at least twice differentiable.
- 2) $s(x, y)$ has a unique minimum at $\hat{\mathbf{q}}$.

For this case, it is possible to devise stability and convergence proofs (see Section V-D).

B. Controllers Based on a Fluid Model

The second stage of our methodology consists of the application of decentralized controllers that drive the robots to the region where the pattern is located and distributes them inside it. These controllers are derived by considering each robot as an SPH particle subjected to an external force. We consider an incompressible fluid model as this allows for a high-level way of controlling the connectivity of the swarm. We begin by presenting a controller for fully actuated holonomic point robots and then extend this controller to permit evaluation on differential drive vehicles in experimentation.

1) *Holonomic Point Robot Abstraction:* Our controller is derived by considering each robot as an SPH particle $\mathbf{q}_i = [x_i, y_i]^T$ subjected to an external force that is generated from the descent gradient of a global potential function. In this abstraction, we consider vehicles with second-order dynamics. Under the assumption of fully actuated holonomic point robots, each robot’s acceleration is given by

$$\ddot{\mathbf{q}}_i = \mathbf{u}_i(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t) \quad (27)$$

where $\bar{\mathbf{q}} = [\mathbf{q}_1^T, \dots, \mathbf{q}_N^T]^T$ is the configuration of the group. While N is the total number of robots, we will show later that the control law for agent i depends only on the agents in the neighborhood \mathcal{N}_i .

The control law for each robot is given by

$$\mathbf{u}_i(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{b}_i - \zeta \mathbf{v}_i + k \mathbf{f}_i \quad (28)$$

where

$$\mathbf{b}_i = - \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} \quad (29)$$

$\mathbf{v}_i = d\mathbf{q}_i/dt$, k and ζ are positive tuning constants and \mathbf{f}_i is given by a vector $-\nabla\phi$. In (28), we include a dissipative damping term proportional to the robot velocity \mathbf{v}_i to ensure system stability.

We consider vector fields of the form

$$\mathbf{f}_i = \begin{cases} -\frac{\nabla\phi(\mathbf{q}_i)}{\|\nabla\phi(\mathbf{q}_i)\|^\beta}, & \text{if } \nabla\phi(\mathbf{q}_i) \neq 0 \\ 0, & \text{if } \nabla\phi(\mathbf{q}_i) = 0 \end{cases} \quad (30)$$

where β is a nonnegative integer number. In (29), the SPH conservation of momentum equation [see (14)] is used. In this study, we use the density ρ_i that is defined in (13), the cubic spline kernel W that is defined in (4), the artificial viscosity Π_{ij} that is defined in (16), the speed of sound that is defined in (23), and the equation of state that determines the pressure P_i for incompressible fluids (19) with B_i given in (22).

It is important to mention that $\mathbf{u}_i(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}})$ in (28) can be computed by taking into account only robots in the neighborhood \mathcal{N}_i defined in (24) due to the compact support of the kernel W that guarantees that robots outside the given neighborhood do not contribute to the sum in (29).

Remark 3: If we use a harmonic function as the global potential function, the proposed solution is analogous to the solution of a problem where a charged fluid is confined in a region where an electrostatic field is applied. Moreover, if the FEM is used to compute the harmonic function, this solution establishes a weak coupling³ between the FEM and the SPH.

2) *Finite-Size, Nonholonomic Robots:* We now describe how our approach may be adapted to take into account practical robot issues. The first issue that we address is the finite size of actual robots. The static obstacles are directly taken into account as we plan our potential functions in the robot configuration space. We also assume that our robots are circular in shape with radius R . Given two robots, we guarantee that the robots do not collide with each other if $\|\mathbf{q}_{ij}\| \geq 2R + \varepsilon$, where ε is a safety factor. In fact, we will show in Section VI-A that this safety factor may represent a position uncertainty due to localization errors. The collision avoidance of our approach is performed by the artificial viscosity term in (16), with

$$\mu_{ij} = \frac{h \mathbf{v}_{ij} \cdot \mathbf{q}_{ij}}{(\|\mathbf{q}_{ij}\| - (2R + \varepsilon))^2}. \quad (31)$$

This adaptation guarantees a repulsive term in (29) between robots which are moving toward each other. This term is repulsive since $\Pi_{ij} \geq 0$ and $\nabla_i W_{ij}$ point in the direction of $-\mathbf{q}_{ij}$. Note that $\Pi_{ij} \rightarrow \infty$ when $\|\mathbf{q}_{ij}\| \rightarrow (2R + \varepsilon)$, i.e., when the robots are about to collide. Indeed, it is hard to analytically prove the collision avoidance property of the artificial viscosity. However, no collisions were observed in the many simulations that are considered in Section VI.

We must also consider the nonholonomic constraints of the differential drive platforms that are used in the experiments. Differential drive platforms can be controlled by specifying their linear and angular velocities v and ω , respectively, and are subjected to the no-slip constraint $\dot{x} \sin(\theta) - \dot{y} \cos(\theta) = 0$, where

³This coupling is said to be weak as the FEM is executed only once and does not take into account the current distribution of the SPH particles.

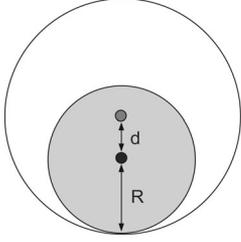


Fig. 3. Feedback linearization point. The light gray circle represents a circular robot with radius R . Assuming a feedback linearization point at a distance d from the center, the white circle with radius $R' = R + d$ is used to provide collision avoidance.

θ is the robot orientation. Therefore, the following model may be used:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (32)$$

In order to treat the differential drive model in (32) as a kinematic point-model system with finite size, we redefine the system output as $[x_d, y_d]^T = [x + d \cos(\theta), y + d \sin(\theta)]^T$, which corresponds to the position of the point $[d, 0]^T$ in the robot frame. Therefore

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -d \sin(\theta) \\ \sin(\theta) & d \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (33)$$

The robot may then be controlled via feedback linearization [37], [38]:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{\sin(\theta)}{d} & \frac{\cos(\theta)}{d} \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_d^{\text{ref}} \\ \dot{y}_d^{\text{ref}} \end{bmatrix}. \quad (34)$$

Therefore, by the application of (34) into (33), we obtain

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = \begin{bmatrix} \dot{x}_d^{\text{ref}} \\ \dot{y}_d^{\text{ref}} \end{bmatrix}.$$

Note that the evolution of the robot orientation $\theta(t)$ is not controlled. Thus, this is, in fact a partial feedback linearization.

As our controllers are devised for fully actuated second-order robots but must control kinematic differential drive robots, we integrate the acceleration inputs in (28):

$$\begin{bmatrix} \dot{x}_{d_i}^{\text{ref}} \\ \dot{y}_{d_i}^{\text{ref}} \end{bmatrix} = \int \mathbf{u}_i(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) dt. \quad (35)$$

Each robot is represented in its configuration space by the feedback linearization point $[x_{d_i}, y_{d_i}]^T$ such that the physical extent of the robot lies within the circle of center $[x_{d_i}, y_{d_i}]^T$ and radius $R' = R + d$ (see Fig. 3). Our approach is adapted such that the SPH particles are placed at the points $[x_{d_i}, y_{d_i}]^T$. Moreover, we replace the robot radius R in (31) by $R' = R + d$.

C. Virtual Particles

The external force in (28) \mathbf{f}_i drives the robots toward the goal and aims to avoid collisions between robots and static obstacles. When controlling multiple robots, due to the presence

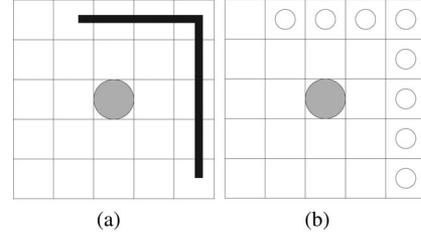


Fig. 4. Multiple virtual particles in an occupancy grid. (a) Robot and obstacles. (b) Virtual particles from occupied cells.

of interparticle forces, \mathbf{b}_i , the external force, \mathbf{f}_i , may not be enough to avoid collisions. We add temporary virtual particles at the boundaries of the workspace to provide collision avoidance. One option is to take advantage of the collision avoidance property provided by the artificial viscosity. There are several ways to implement this virtual particle idea. A first idea is to create a temporary virtual particle at the closest boundary point \mathbf{p} . We then adapt the term \mathbf{b}_i in (29) such that

$$\begin{aligned} \mathbf{b}'_i = & - \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij}(h) \\ & - \lambda \Pi_{ip} \nabla_i W_{ip}(h') \end{aligned} \quad (36)$$

where λ is a positive constant, j iterates only through the N particles that represent real robots, and p refers to the virtual particle. Notice that, in this case, the virtual particle does not change the density ρ_i , nor does it have its own density. The other terms necessary to compute Π_{ip} are $\bar{\rho}_{ip} = \rho_i$ and $\bar{c}_{ip} = c_i$.

Instead of using a single virtual particle, another option is to assign virtual particles to each cell with an obstacle in a local occupancy grid. This option was found to be the most robust during the experiments (see Fig. 4).

Another possible implementation of virtual particles is to consider very dense virtual particles. These particles may be considered just like the other particles in the conservation of momentum equation (29). The key idea is that particles which are denser than the others are repulsive particles. To understand this fact one should agree that particles which are denser than the reference density ρ_0 produce positive values of pressure according to the equation of state (19).

Since the gradient $\nabla_i W_{ij}$ at particle i points toward particle j , it is clear that terms with positive values of pressure in (29) repel particle i .

To create very dense particles, we propose to assign a high value of mass m_d to these particles. The density of a particle i is computed according to the conservation of mass

$$\rho_i = \sum_j m_j W(\mathbf{q}_i - \mathbf{q}_j, h) \quad (37)$$

where

$$W(\mathbf{q}, h) = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}\kappa^2 + \frac{3}{4}\kappa^3, & \text{if } 0 \leq \kappa \leq 1 \\ \frac{1}{4}(2 - \kappa)^3, & \text{if } 1 \leq \kappa \leq 2 \\ 0, & \text{otherwise} \end{cases} \quad (38)$$

where $\kappa = \|\mathbf{q}\|/h$. We require that the density of the virtual particle $\rho_d \gg \rho_0$ by letting $\rho_d = \sigma\rho_0 = m_d W(0, h)$, where $\sigma \gg 1$. Therefore, we compute m_d using

$$m_d = \left(\frac{7\pi h^2}{10} \right) \sigma\rho_0. \quad (39)$$

The density of a particle j with mass given by (39) is guaranteed to be larger than or equal to $\sigma\rho_0$. Thus, the term proportional to the pressure P_j in (29) will be positive, and consequently, it will be a repulsive term.

We can also use virtual particles to address collision avoidance in the presence of dynamic obstacles. Dynamic obstacles are those that are unknown *a priori* and/or are able to move. It should be clear that addressing dynamic obstacles is not the main topic of this study and the proposed solution has limitations. The main issue is the fact that the robots may be trapped by the dynamic obstacles. Since the repulsive terms provide forces which are parallel to the segment which links the robot to the obstacle particle, it is possible to find local minima. One may propose new strategies that are based on forces that act perpendicularly to the referred segment to solve this issue. This is out of the scope of this study and may be a possible future direction of research.

D. Stability and Convergence Analysis

Our stability and convergence analysis is built upon the results in [31] and follows a similar methodology. Similarly, we assume obstacle-free environments and that \mathbf{f}_i in (28) is given by $-\nabla\phi$, where ϕ is the shape function that is described in Section V-A2 for the case of smooth star-shaped patterns. We also assume that the robots are represented by identical SPH particles with mass m .

As in [31], we define the function $\phi_S(\bar{\mathbf{q}})$ as a measure of performance:

$$\phi_S(\bar{\mathbf{q}}) = k \sum_i \phi(\mathbf{q}_i) \quad (40)$$

with $k > 0$. The function ϕ_S provides a measure of how close the team is to Γ . The greater the value of this function, the greater the distance of the team to the pattern.

The following Lemma concerning the Hessian of ϕ_S , \mathbf{H}_{ϕ_S} , will be useful in this section.

Lemma 1: Given a star-shaped boundary Γ as described in Section V-A2, the Hessian of the function ϕ_S , \mathbf{H}_{ϕ_S} is positive semidefinite on Γ .

Proof: [31] According to (40) and (26), we have

$$\phi_S = k \sum_i^N s(\mathbf{q}_i)^2 = k \sum_i^N s_i^2. \quad (41)$$

Thus, the Hessian is given by

$$\mathbf{H}_{\phi_S} = 2k (\mathbf{H}_{\phi_S}^I + \mathbf{H}_{\phi_S}^{II}) \quad (42)$$

where

$$\begin{aligned} \mathbf{H}_{\phi_S}^I &= \text{diag}(\mathbf{H}_{\phi_1}^I, \dots, \mathbf{H}_{\phi_i}^I, \dots, \mathbf{H}_{\phi_N}^I) \\ \mathbf{H}_{\phi_S}^{II} &= \text{diag}(\mathbf{H}_{\phi_1}^{II}, \dots, \mathbf{H}_{\phi_i}^{II}, \dots, \mathbf{H}_{\phi_N}^{II}) \end{aligned}$$

are $2N \times 2N$ block diagonal matrices. Each block is given by

$$\begin{aligned} \mathbf{H}_{\phi_S}^I &= (\nabla_i s_i)(\nabla_i s_i^T) \\ \mathbf{H}_{\phi_S}^{II} &= s_i \mathbf{H}_{s_i} \end{aligned}$$

where \mathbf{H}_{s_i} is the Hessian of s_i . Since $s_i = 0$ at Γ , we conclude that \mathbf{H}_{ϕ_S} is positive semidefinite at Γ . ■

The minimum value $\phi_S = 0$ is obtained when all of the robots reach the desired boundary. Therefore, the primary objective of our controller should be to minimize ϕ_S . The next proposition ensures that our system equilibrium points satisfy the necessary condition for ϕ_S to be at an extremum.

Proposition 1: Given a system of N point robots with dynamics $\ddot{\mathbf{q}}_i = \mathbf{u}_i(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}, t)$ and a control law determined by (28), where $\mathbf{f}_i = -\nabla\phi$ and ϕ is a shape function, then the system equilibrium points satisfy the necessary condition for the shape discrepancy function to be at an extremum.

Proof: Since the system is in equilibrium, we have $\ddot{\mathbf{q}}_i = 0$ and $\dot{\mathbf{q}}_i = 0, i = 1, \dots, N$. Consequently, for every robot, $\Pi_{ij} = 0$, and $\mathbf{u}_i = 0$. Therefore

$$\sum_i \mathbf{u}_i = \sum_i \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} - k \nabla \phi_i \right] = 0.$$

Since $\nabla_i W_{ij} = -\nabla_j W_{ij}$

$$\sum_i \mathbf{u}_i = k \sum_i \nabla \phi_i = 0. \quad (43)$$

However, $k \sum_i \nabla \phi_i = 0$ is the necessary condition for ϕ_S to be at an extremum. ■

Proposition 2: Consider the positive semidefinite function:

$$V = \phi_S + \sum_i e'_i + \frac{1}{2} \bar{\mathbf{v}}^T \bar{\mathbf{v}} \quad (44)$$

where $\bar{\mathbf{v}} = \dot{\bar{\mathbf{q}}} = [\mathbf{v}_1^T, \dots, \mathbf{v}_N^T]^T$, and e'_i is the part of the internal energy that is related to conservative forces such that [see (15)]

$$\frac{de'_i}{dt} = \frac{1}{2} \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \mathbf{v}_{ij} \cdot \nabla_i W_{ij}. \quad (45)$$

Consider also the set $\Omega_c = \{\mathbf{x} \in X | V(\bar{\mathbf{q}}, \bar{\mathbf{v}}) \leq c\}$, where X is the state space that is defined by $\mathbf{x} = [\mathbf{q}_1^T, \mathbf{v}_1^T, \dots, \mathbf{q}_N^T, \mathbf{v}_N^T]^T$, and $c \in \mathbb{R}^+$. Given the system of robots that are defined in Proposition 1 with any initial condition $\mathbf{x}_0 \in \Omega_c$, the system converges to an invariant set $\Omega_I \subset \Omega_c$ such that the points in Ω_I satisfy the necessary condition for the measure function ϕ_S to be at an extremum.

Proof: Since V is continuous, we conclude that Ω_c is closed for some $c > 0$. In addition, due to the fact that $\phi_S + \sum_i e'_i \leq c$ and $\bar{\mathbf{v}}^T \bar{\mathbf{v}} \leq c$, we conclude that Ω_c is compact.

We have that

$$\dot{V} = \sum_i (k \nabla \phi_i^T \dot{\mathbf{q}}_i + \mathbf{v}_i^T \dot{\mathbf{v}}_i) + \sum_i \frac{de'_i}{dt}. \quad (46)$$

By using (28) and (45), and the fact that $\nabla_i W_{ij} = -\nabla_j W_{ji}$, at Ω_S . Let $\Pi_{ij} = \Pi_{ji}$, $\dot{\mathbf{q}}_i = \mathbf{v}_i$, and $\dot{\mathbf{v}}_i = \ddot{\mathbf{q}}_i$, one has that

$$\begin{aligned} \dot{V} &= \sum_i k \nabla \phi_i^T \mathbf{v}_i \\ &+ \sum_i \mathbf{v}_i^T \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} \right] \\ &+ \sum_i \mathbf{v}_i^T (-\zeta \mathbf{v}_i - k \nabla \phi_i) \\ &+ \sum_i \frac{1}{2} \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \mathbf{v}_{ij}^T \nabla_i W_{ij} \\ &= \sum_i \mathbf{v}_i^T \left[-\sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} + \Pi_{ij} \right) \nabla_i W_{ij} - \zeta \mathbf{v}_i \right] \\ &+ \sum_i \frac{1}{2} \mathbf{v}_i^T \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \\ &+ \sum_j \frac{1}{2} \mathbf{v}_j^T \sum_i m \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla_j W_{ji} \\ &= -\sum_i \zeta \mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m \Pi_{ij} \mathbf{v}_i^T \nabla_i W_{ij} \\ &\quad - \sum_j \frac{1}{2} \sum_i m \Pi_{ji} \mathbf{v}_j^T \nabla_j W_{ji} \\ &= -\sum_i \zeta \mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m \Pi_{ij} \mathbf{v}_{ij}^T \nabla_i W_{ij}. \end{aligned}$$

Our kernel is such that

$$\nabla_i W_{ij} = -\frac{\|\nabla_i W_{ij}\| \mathbf{q}_{ij}}{\|\mathbf{q}_{ij}\|}. \quad (47)$$

By using the fact that $\Pi_{ij} > 0$ when $\mathbf{v}_{ij}^T \mathbf{q}_{ij} < 0$ and $\Pi_{ij} = 0$ otherwise, we conclude that

$$\dot{V} = -\sum_i \zeta \mathbf{v}_i^T \mathbf{v}_i - \sum_i \frac{1}{2} \sum_j m \Pi_{ij} \mathbf{v}_{ij}^T \nabla_i W_{ij} \leq 0. \quad (48)$$

By using LaSalle's invariance principle, we conclude that for any $\mathbf{x}_0 \in \Omega_c$ the system converges asymptotically to the largest invariant set Ω_I that is contained in $\Omega_o = \{\mathbf{x} \in X | \dot{V} = 0\}$, which corresponds to $\mathbf{v}_i = 0 \forall i$, and $\Omega_o \subset \Omega_c$. Since Ω_I contains all equilibrium points in Ω_c and based on Proposition 1, we conclude that all points in Ω_I satisfy the necessary condition for ϕ_S to be at an extremum. ■

Proposition 3: Consider the set Ω_S that is defined by

$$\Omega_S = \{\mathbf{x} \in X | \phi(\mathbf{q}_i) = 0, \mathbf{v}_i = 0, \rho_i = \rho_0, i = 1, \dots, N\}$$

where ϕ is a shape function. Given the system of N robots defined in Proposition 1, the set Ω_S is a stable submanifold and $\Omega_S \subset \Omega_I$.

Proof: The potential energy of the system is given by $U = \phi_S + \sum_n e'_n$. First, we show that the gradient of U is equal to 0

$$\frac{de'_i}{dt} = \frac{\partial e'_i}{\partial t} + \frac{\partial e'_i}{\partial \mathbf{q}_i} \mathbf{v}_i + \sum_{j \neq i} \frac{\partial e'_i}{\partial \mathbf{q}_j} \mathbf{v}_j. \quad (49)$$

But the temporal partial derivative is zero and de'_i/dt is given by (45). By using $\nabla_i W_{ij} = -\nabla_j W_{ji}$ in (45), we can write, after some algebra

$$\frac{\partial e'_i}{\partial \mathbf{q}_i} = \frac{1}{2} \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \quad (50)$$

$$\frac{\partial e'_i}{\partial \mathbf{q}_j} = \frac{1}{2} m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_j W_{ji}. \quad (51)$$

Therefore

$$\frac{\partial \sum_n e'_n}{\partial \mathbf{q}_i} = \sum_j m \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij}. \quad (52)$$

After using the state equation (19) and (22)

$$\frac{\partial \sum_n e'_n}{\partial \mathbf{q}_i} = \sum_j m \left[\kappa \frac{\rho_i^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_i} + \kappa \frac{\rho_j^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_j} \right] \nabla_i W_{ij}$$

where κ is a positive constant.

If $\rho_i = \rho_j = \rho_0$, which is a necessary condition for a point in Ω_S , then this gradient is null. For a shape function, it is shown in [31] that $\nabla_i \phi_i = 0$ at Γ ($\phi(\mathbf{q}_i) = 0$). Therefore, $\nabla U = 0$ at Ω_S .

Now, we need to show that the Hessian of U , $\mathbf{H}_U = \mathbf{H}_{\phi_S} + \mathbf{H}_{\sum_i e'_i}$, is positive semidefinite when $\mathbf{q}_i \in \Gamma$ and $\rho_i = \rho_0 \forall i$. It is proved in Lemma 1 that the $2N \times 2N$ matrix \mathbf{H}_{ϕ_S} is positive semidefinite when $\phi(\mathbf{q}_i) = 0$. Therefore, we need to prove that $\mathbf{H}_{\sum_i e'_i} \geq 0$ when $\rho_i = \rho_0$.

The second derivatives are given by

$$\begin{aligned} \frac{\partial^2 \sum_n e'_n}{\partial \mathbf{q}_i \partial \mathbf{q}_i} &= \sum_j \frac{\partial^2 \sum_n e'_n}{\partial \rho_j \partial \mathbf{q}_i} \frac{\partial \rho_j}{\partial \mathbf{q}_i} + \frac{\partial^2 \sum_n e'_n}{\partial \rho_i \partial \mathbf{q}_i} \frac{\partial \rho_i}{\partial \mathbf{q}_i} \\ &+ \sum_j m \left[\kappa \frac{\rho_i^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_i} + \kappa \frac{\rho_j^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_j} \right] \frac{\partial \nabla_i W_{ij}}{\partial \mathbf{q}_i} \\ \frac{\partial^2 \sum_n e'_n}{\partial \mathbf{q}_k \partial \mathbf{q}_i} &= \sum_j \frac{\partial^2 \sum_n e'_n}{\partial \rho_j \partial \mathbf{q}_i} \frac{\partial \rho_j}{\partial \mathbf{q}_k} + \frac{\partial^2 \sum_n e'_n}{\partial \rho_i \partial \mathbf{q}_i} \frac{\partial \rho_i}{\partial \mathbf{q}_k} \\ &+ \sum_j m \left[\kappa \frac{\rho_i^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_i} + \kappa \frac{\rho_j^{\gamma-1}}{\rho_0^\gamma} - \frac{\kappa}{\rho_j} \right] \frac{\partial \nabla_i W_{ij}}{\partial \mathbf{q}_k}. \end{aligned}$$

After manipulations and using the fact that $\rho_i = \rho_0$:

$$\mathbf{H}_{\sum_i e'_i} = \frac{m^2 \gamma \kappa}{\rho_0^2} \mathbf{A} \mathbf{A}^T \geq 0 \quad (53)$$

where

$$\mathbf{A} = \begin{pmatrix} \sum_k \frac{\partial W_{1k}}{\partial x_1} & \frac{\partial W_{12}}{\partial x_1} & \dots & \frac{\partial W_{1N}}{\partial x_1} \\ \sum_k \frac{\partial W_{1k}}{\partial y_1} & \frac{\partial W_{12}}{\partial y_1} & \dots & \frac{\partial W_{1N}}{\partial y_1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial W_{N1}}{\partial x_N} & \frac{\partial W_{N2}}{\partial x_N} & \dots & \sum_k \frac{\partial W_{Nk}}{\partial x_N} \\ \frac{\partial W_{N1}}{\partial y_N} & \frac{\partial W_{N2}}{\partial y_N} & \dots & \sum_k \frac{\partial W_{Nk}}{\partial y_N} \end{pmatrix}.$$

Therefore, Ω_S is a stable submanifold and since $\mathbf{v}_i = 0$ for all i , $\Omega_S \subset \Omega_I$. ■

Proposition 4: For any smooth star shape, Γ , the system of N robots defined in Proposition 1 with fixed ρ_i such that $\rho_i = \rho_0, i = 1, \dots, N$, converges to the desired boundary for any $\mathbf{x}_0 \in \Omega_c$.

Proof: If $\rho_i = \rho_0 \forall i$, for all time t , then the equilibrium of the system is given by $\nabla_i \phi(\mathbf{q}_i) = 0 \forall i$. Proposition 2 guarantees the system converges to Ω_I . The function ϕ is a shape function and according to [31] $\nabla_i \phi(\mathbf{q}_i) = 0$ if and only if $\mathbf{q}_i \in \Gamma$. Therefore, $\Omega_I \equiv \Omega_S \equiv \Gamma$. ■

Remark 4: We are able to show that the system moves toward the desired pattern in an obstacle-free environment. However, in general, it will converge to an invariant set which may be different from the desired pattern. If the number of robots is too large, for example, it should be expected that some of these robots will not converge.

Remark 5: Obtaining proofs for the case of external forces derived from harmonic functions is out of the scope of this study. In this case, the gradients have discontinuities exactly at the desired pattern.

VI. NUMERICAL SIMULATIONS

In this section, we validate our approach with simulations developed in C++ computer language [39] and OpenGL [40]. First, we present two simulations with nonpoint robots and static obstacles in the presence of localization errors. Kinematic constraints and other imperfections such as actuator saturation are not considered. The main goal of these simulations is to validate the robustness of the proposed approach to noise in the localization system. A third simulation illustrates the system behavior in the presence of dynamic obstacles. Finally, we show a fourth simulation with a very large group of robots to demonstrate scalability of the method. In the last two cases, we considered ideal agents in the sense that no practical issues such as saturation, noise, kinematic constraints, finite size, etc., were implemented. Harmonic functions were used as global potential functions in all simulations. We use the open source software *Triangle* [41] to generate triangular meshes and the software FEMM (FEM Magnetics [42]), version 4.0, for the FEM computations. Videos of simulations and experiments are available at: <http://www.cpdee.ufmg.br/~coro/movies/sph/>.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
h	0.05 m
ξ_1	1
ξ_2	2
η^2	$0.01h^2$
γ	1
ζ	50
k	300
m_i	3.189 kg
H	$1/98$ m
g	9.8 m/s^2
ρ_0	1000 kg/m^3

A. Robustness to Localization Errors

In the two examples that are presented in this section, we used the parameters that are shown in Table I. The values of ξ_1, ξ_2 , and η^2 are recommended values that are described in [28]. Since g is the gravity acceleration, its value is a standard value. The damping ζ was tuned to avoid intense oscillatory behavior close to the target region. The values of γ and H were experimentally tuned by observing the system behavior during simulations. We verified that these parameters influenced the proximity of the robots during the task execution but were not decisive in the accomplishment of the task.

The external force gain k was determined in order to provide a force component larger than those obtained from pressure gradient. This means that converging to the target has priority over controlling density.

The most important parameters to tune are the mass m_i and the parameter h . These parameters are used in the computation of the density, and together with the reference density ρ_0 , they affect directly the distance the robots try to keep from their neighbors.

As the value $2h$ determines the size of the robot's neighborhood, the smoothing length h is determined according to the hardware of the robots. If the robots receive state information from their neighbors via communication, we set $2h$ as the communication radius. In the following simulations, we assumed a communication radius of 0.1 m.

We also assumed circular small robots with radius $R = 5$ mm. This value corresponds to $0.1h$, i.e., the communication radius is 20 times larger than the physical radius of the robot. This relation is reasonable for actual robotic platforms. Regarding robustness to localization errors, we designed the parameter ε in (31) in order to avoid collisions even if a Gaussian random noise with zero mean and standard deviation $\sigma_N = 2$ mm is added to the estimate of the robots position. Indeed, this noise represents an imprecise localization system, as $3\sigma_N$ is larger than the robots radii and given by

$$\varepsilon = 6\sigma_N.$$

One may interpret the aforementioned equation as a robot radius growing strategy. Given the properties of a zero mean Gaussian noise, we can assume that the estimate of a robot position will be confined to a circle that is centered at the correct position

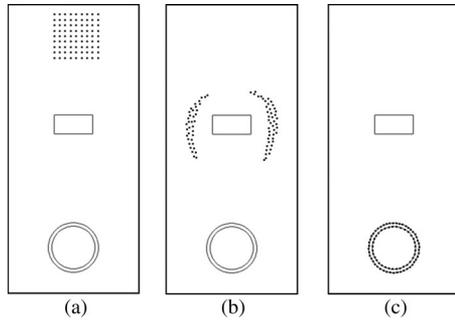


Fig. 5. Simulation with 81 point robots in a simple environment from a starting configuration (a) to the goal (c), with intermediate configuration (b).

TABLE II
SUCCESS RATE IN THE PRESENCE OF LOCALIZATION
ERRORS—SIMPLE ENVIRONMENT

Noise Standard Deviation (σ_N)	Success Rate
0.002	100%
0.005	100%
0.006	90%
0.007	60%

with radius equal to $3\sigma_N$. Therefore, we grow the robot size by adding $\varepsilon/2 = 3\sigma_N$ to its actual radius.

The localization noise and robot radius were also taken into account to compute the mass m_i . If the distance between two robots is smaller than $2R + \varepsilon$ (0.022 m, in our example), the chance of having a collision increases. Thus, we compute m_i so that the robots repel each other before reaching such a distance value. We arbitrarily chose the reference density $\rho_0 = 1000 \text{ kg/m}^3$ (this is the density of water) and compute m_i to guarantee repulsion for distances smaller than $0.5h$, in this case, 0.025 m:

$$m_i = \frac{1000}{W(0, 0.05) + W(0.5, 0.05)}$$

where W is given by (4), resulting in a value of mass equal to 3.189 kg.

1) *Example 1:* We first evaluated the system behavior in the presence of localization errors in a simple environment with a single rectangular static obstacle and 81 robots. A trial is presented in Fig. 5. Table II summarizes the obtained results for different values additive Gaussian noise that is introduced into the robot state estimates (position and velocity). We ran ten trials for each standard deviation σ_N with success that is determined by the convergence of all robots to the target without collisions between robots, walls, or obstacles.

One can note that the robots performed well even in the presence of localization errors with a standard deviation higher than expected, validating the robustness of our approach. The parameters of the controller were set assuming $\sigma_N = 0.002$, but we observed 100% success rate in the case of $\sigma_N \leq 0.005$.

2) *Example 2:* A trial with 81 robots in a maze-like environment is presented in Fig. 6. We followed the same procedure that is detailed in Example 1 to evaluate robustness. Table III summarizes the obtained results for different values of noise standard deviation in this complex environment. Again, the task was well

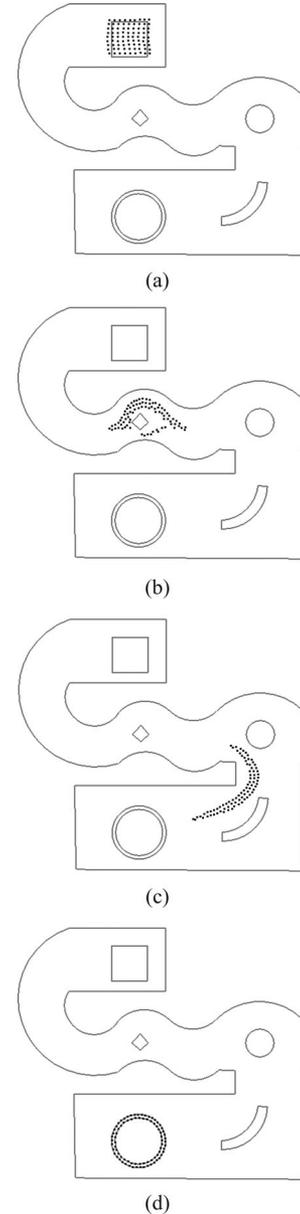


Fig. 6. Simulation with 81 point robots in a maze-like environment from a starting configuration (a) to the goal (d), with intermediate configurations (b) and (c).

TABLE III
SUCCESS RATE IN THE PRESENCE OF LOCALIZATION
ERRORS—COMPLEX ENVIRONMENT

Noise Standard Deviation (σ_N)	Success Rate
0.002	100%
0.004	100%
0.005	90%
0.0055	70%
0.006	40%

executed with higher uncertainty than expected $\sigma_N \leq 0.004$. From this, we can conclude that our approach can be successfully used in complex geometric environments in the presence of reasonable localization errors.

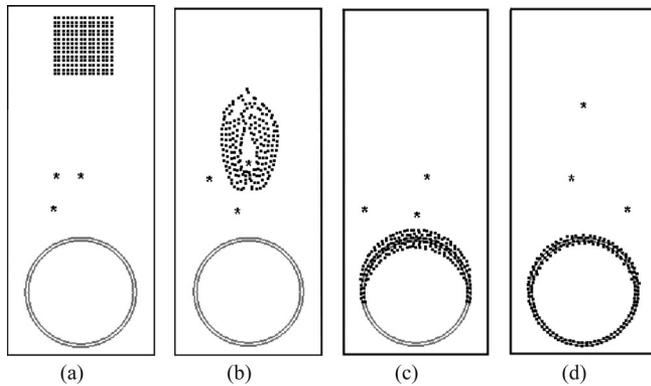


Fig. 7. One hundred ninety six robots generating a circle pattern in the presence of dynamic particle obstacles.

B. Dynamic Obstacles

Fig. 7 presents different time samples of a simulation that considers three dynamic particle obstacles. These obstacles are represented by stars (*) and are both unknown *a priori* and mobile. Two of these obstacles have periodic horizontal movement, and one of them has a periodic vertical movement. In this case, we used the very dense particles idea to allow the swarm to avoid obstacles (see Section V-C). We used $\sigma = 10$ to compute the mass of the particle obstacles. The robots are initially distributed in a 14×14 matrix.

From Fig. 7, one can conclude that the idea of considering dynamic obstacles as very dense particles is feasible. When a particle obstacle reaches the range of action of a swarm particle, i.e., when the distance between these two particles is smaller than $2h$, the swarm particle is subject to the influence of a virtual repulsive force and the obstacle is properly avoided. It is interesting to mention that even in the target region, the particles are still able to avoid the dynamic obstacles.

In this simulation, the value of the swarm particle mass was chosen to be $\rho_0 h^2$. This mass value provides repulsion between agents if a robot has more than four neighbors at distances smaller than or equal to h . Except for the mass, the other parameters were those presented in Table I.

C. Very Large Groups

In this section, we show a simulation in a complex environment similar to the one in Fig. 6 with a very large group. We considered 1600 robots that navigate in a maze-like environment. In order to fit such a number of robots into this environment, we decreased the smoothing length to 0.02 m. This value of h allows the agents to move closer to each other. As in the last example, we have also set $m_i = \rho_0 h^2$, and the other parameters were those presented in Table I. Fig. 8 shows a typical trial with this simulation setup. All of the robots converged without collisions, as expected. This simulation demonstrates the scalability of our approach.

VII. EXPERIMENTS

In this section, we present experimental results that verify the proposed approach for finite size, nonholonomic robots. For all

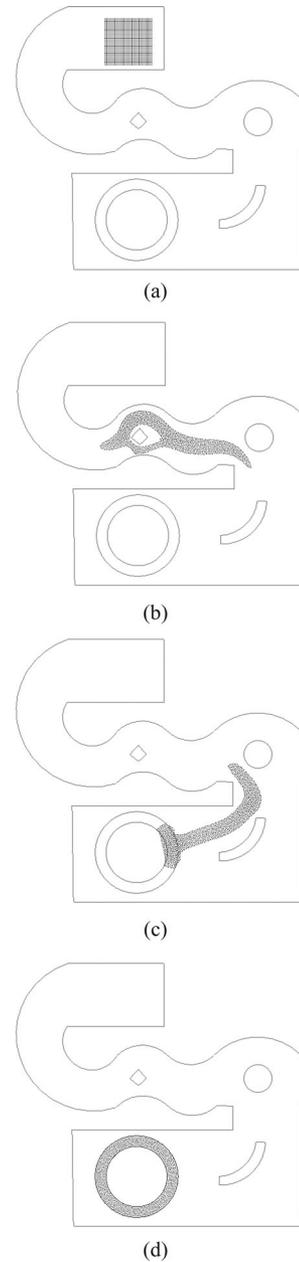


Fig. 8. Simulation with 1600 point robots in a maze-like environment from a starting configuration (a) to the goal (d), with intermediate configurations (b) and (c).

experiments, the static, global vector field is computed using the FEM, which, in our case, relies on a Delaunay Triangulation-based mesh (see [35] for details).

A. First Set of Experiments

1) *Experimental Setup*: In [8], we presented experimental results that are obtained using the experimental testbed for multirobot experiments developed at the GRASP (General Robotics, Automation, Sensing, and Perception) Laboratory at the University of Pennsylvania, Philadelphia. This testbed is composed of differential drive, kinematically controlled robots called *Scarabs* (see Fig. 9) and an overhead

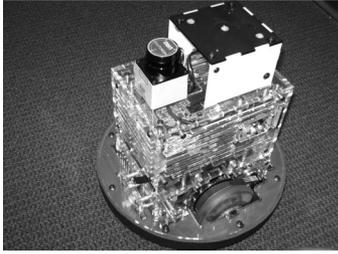


Fig. 9. The $20 \times 13.5 \times 22.2$ cm³ differential drive robotics platform.

tracking system that provides pose information in a global reference frame.

Each Scarab is equipped with an onboard 1-GHz computer, power management system, wireless communication, and is actuated by stepper motors. The tracking system consists of seven IEEE 1394 cameras, computers, and tracking targets at the top of the robots (see Fig. 9). An extended Kalman filter (EKF) is used to fuse the information from multiple cameras. An EKF also runs in each robot to fuse the local odometry and the overhead tracking information. This system has been successfully used to track tens of robots simultaneously with a position error of approximately 2 cm and an orientation error of 5° at 29 Hz in a $9 \times 6 \times 6$ m³ room.

At the software level, the GRASP platform uses the open-source software that is developed by the Player/Stage/Gazebo (PSG) project [43]. Player is a network server for robot control. It provides the interfaces such that one can easily have access to the robot's sensors and actuators over an IP network. The fluid-based approach was successfully implemented in the given testbed. The algorithms were coded in C++ in the form of a Player driver. Further details of this infrastructure are described in [44].

2) *Results:* In this set of experiments, we used a team of seven *Scarabs*. The team of robots were provided with a map of the environment which was defined by the boundaries of the experimental area and static obstacles. A vector field that was based on harmonic functions was computed offline. Each robot computed its location in the map that is based on localization information from the overhead tracking system and current velocity from its motor controller. This information was broadcast over the network together with the necessary SPH parameters (mass and density) to the other robots. To emulate the concept of neighborhoods in the smaller experimental area, each robot ignored messages from robots a distance greater than 2 m. At every update of the control algorithm, each robot computed its current SPH state based on its local information and the information received over the network. Additionally, each robot used virtual particles that are based on the map within a region $2h' \times 2h'$, with $h' = 0.3$ m. During the experimentation, the algorithm was found to be more robust when multiple virtual particles were defined. Each virtual particle was assigned to each cell with an obstacle in a local occupancy grid (see Fig. 4).

A vignette of one trial run of the implementation is shown in Fig. 11. The team of robots started in an initial configuration [see Fig. 11(a)] and was given a circular goal formation. The control

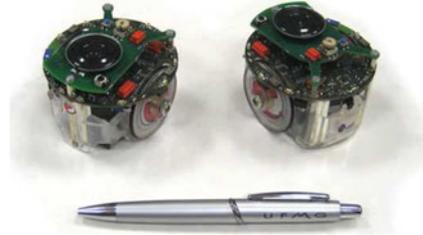


Fig. 10. Two *E-Pucks* from the VeRLab Swarm. Each *E-puck* has a 7-cm diameter and is equipped with encoders and bluetooth communication, among other resources.

law executed at an update rate of 10 Hz while the interagent network communications executed at a higher rate of 20 Hz to accommodate the asynchronous system design. One can see that two robots did not converge to the goal. This was due to limitations of the minimum allowed velocities of the actual robots. As one of the robots already reached the goal and was not moving, the two other robots were unable to converge.

B. Second Set of Experiments

1) *Experimental Setup:* After verifying our approach with a group of seven robots, we decided to run experiments with a larger team. Thus, we implemented our solution in the VeRLab⁴ Swarm, which is composed of 12 *E-Puck* robots, such as the ones that are presented in Fig. 10. These differential drive, circular robots are designed for education and research [45]. Although the robots are equipped with several sensors, such as infrared sensors, odometry, and camera, these resources were not used for localization in our experiments. Alternatively, we built a visual-based global localization system composed of a set of external cameras that track the pose of each robot. To facilitate the tracking, each robot is outfitted with a 2-D, monochromatic bar code. Each bar code is tracked with the help of the ARToolkitPlus library,⁵ as detailed in [46].

To facilitate the code development and simulation, we again use the tools from Player project [43] and developed an *E-Puck* interface to support this study [46]. With such an interface, it was possible to control the robots using standard, Linux-based computers via Bluetooth communication. We worked with three computers for the experiments in this paper: The first computer was responsible for vehicle tracking while the remaining two computers communicated with vehicles via Bluetooth (six vehicles per computer). The reason for using two different computers controlling six robots each is that the Bluetooth protocol, which is used to communicate with the *e-pucks*, allows a device to simultaneously communicate with up to seven other devices.⁶ Note that the controls for each robot are computed by separate processes/threads with interrobot communication enabled via UDP/IP protocol. Thus, there was no synchronization among the processes and each robot could be controlled by an individual computer. At 10 Hz, each robot process connects to the localization server (using a Player interface), gets the respective

⁴<http://www.verlab.dcc.ufmg.br/>

⁵<http://www.hitl.washington.edu/artoolkit/>

⁶<http://developer.bluetooth.org/>

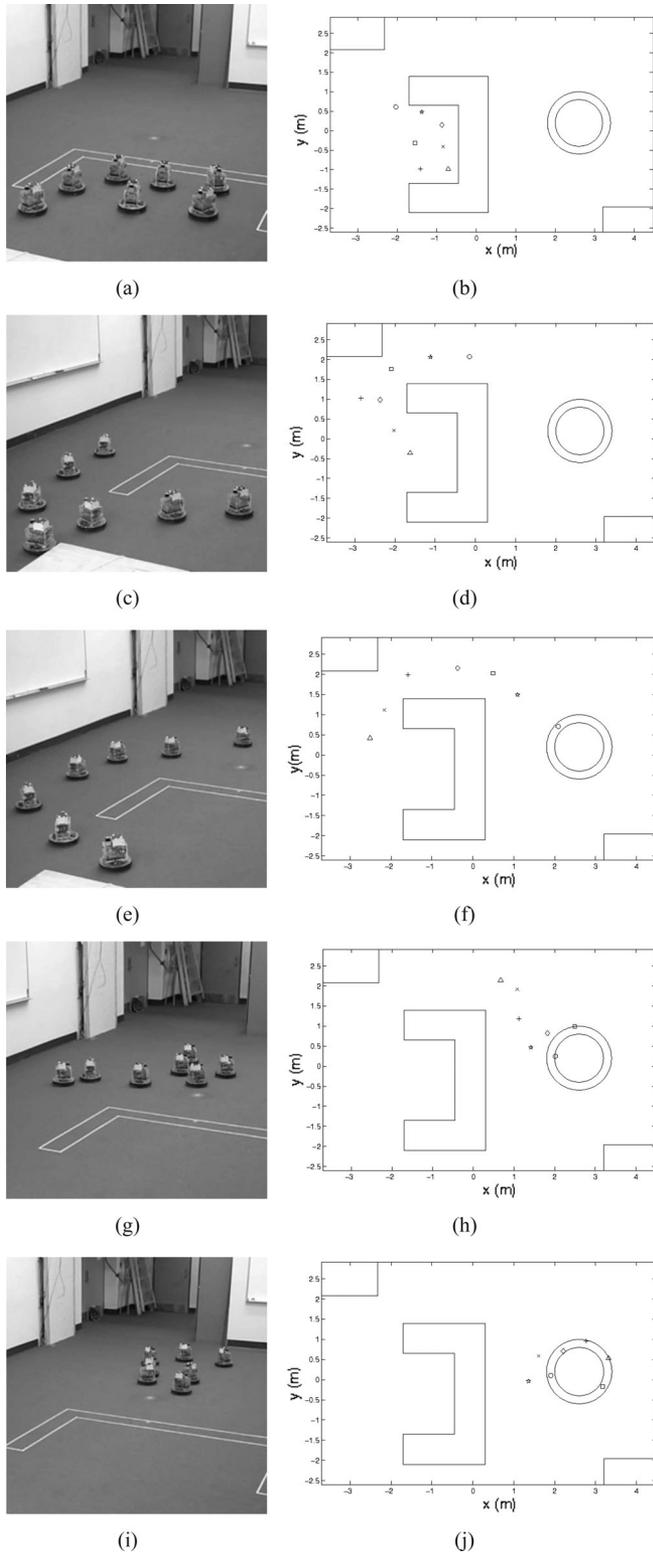


Fig. 11. *Experimental Results.* A team of seven robots, starting from an initial configuration [see Fig. 11(a)], control around an obstacle [see Figs. 11(c), (e), and (g)] to a goal circular formation [see Fig. 11(i)]. The results in the configuration space are also shown in Figs. 11(b), (d), (f), (h), and (j).

robot pose, and broadcasts this information along with the velocities from the robot encoders and the SPH parameters (mass and density).

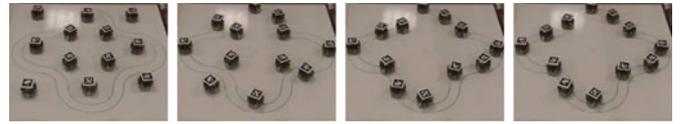


Fig. 12. Snapshots from the first experiment.

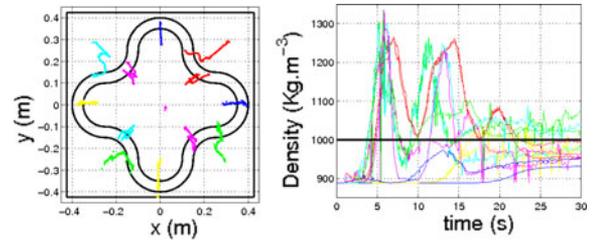


Fig. 13. Data for the experiment in Fig. 12. Path of each robot in the swarm (at left) and each robot density evolution (at right).

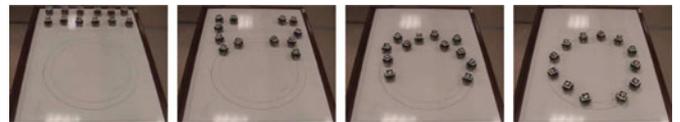


Fig. 14. Snapshots from the second experiment.

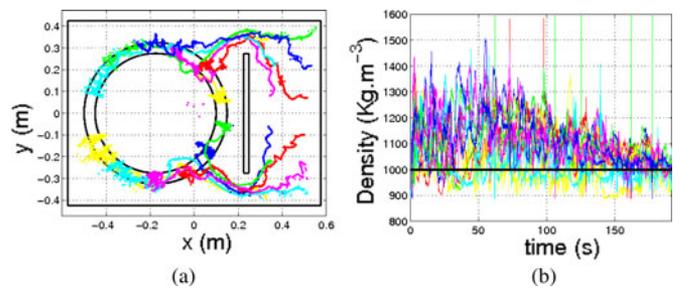


Fig. 15. Data for the experiment in Fig. 14. The path of each robot in the swarm (left) and the evolution of the density of individual robots (right). Notice that the robots start behind the rectangular obstacle and move toward the circular target.

We performed experiments with the 12 robots in two different workspaces: one that is free of obstacles with a star-shaped target and another one with a single rectangular obstacle and a circular target. As the robots are circular, we assume a simplified 2-D configuration space (orientation is ignored) that is obtained by simply enlarging the workspace obstacles by the robot radius.

2) *Results:* Fig. 12 shows four snapshots of the first experiment while Fig. 13 presents the robots' paths in the corresponding configuration space as seen by the localization system. Note that the swarm converges to the target and collisions are avoided. Fig. 13 also presents, at the right-hand side, the evolution of the robots' density. As expected, this variable converges to values close to the reference value of 1000 kg/m^3 .

Another set of experiments was performed with a different configuration space. Snapshots of a trial where all robots start behind the obstacle is presented in Fig. 14. The robots' paths and densities are depicted in Fig. 15. Again, note that the swarm reached the target while avoiding collisions. The densities converged to values close to the reference value.

The spikes and noise present in Fig. 15 follow directly from the noise in the localization system (i.e., increased uncertainty in the position estimate resulted in higher noise in the density). Note that the density [see (37)] is a nonlinear function of the position of the robot and of its neighbors. We should mention that during the experiments certain regions of the workspace were subject to worse position estimation than others due to issues in our localization system.

VIII. CONCLUSION

In this study, we have proposed a novel approach to control swarms of robots. Our approach is scalable and does not require the labeling of individual robots. Therefore, all of the robots run the same software and the success of the task execution is not dependent on specific members of the group. The approach is also robust to the dynamic deletion and addition of new agents.

We propose to use the SPH simulation technique to create an abstract representation of the swarm as an incompressible fluid subjected to an external force. Control of the incompressible fluid density provides a loose way of controlling the swarm connectivity. We also use the FEM to compute harmonic functions that determine force external to the fluid. These forces are mainly responsible to drive the swarm to a desired region of the workspace. As obstacles may have generic geometries, the use of the FEM allows for efficient function computation. By means of a weak coupling between the FEM and the SPH, the derived controllers are decentralized in the sense that only local information is used by each robot of the group, including the gradient of the harmonic function at the location of the robot as well as the position, velocity, mass, and density of the robot and its neighbors.

The proposed approach was successfully instantiated in a pattern generation task. Although we presented our approach considering patterns as 1-D sets, it is straightforward to implement it in the case of patterns that are determined by other types of subsets of \mathbb{R}^2 such as 2-D sets, point collections, and even disjoint sets. Techniques to accommodate actual robot features such as finite size and nonholonomic constraints (more specifically a nonslipping constraint) are proposed. This is accomplished by using feedback linearization and by adapting the artificial viscosity term in the SPH formulation. The vector field that is computed from the designed harmonic function helps to avoid collisions between robots and static obstacles. However, the vector field design may be not enough to ensure that no collisions occur. A strategy that places virtual particles at the boundaries of the obstacles is proposed to provide collision avoidance. By using the idea of very dense SPH particles, the method is also able to deal with dynamic obstacles.

Computer simulations and actual robot experiments were performed showing the efficacy of the method. Based on the results of the presented simulations, we can conclude that the proposed approach is able to deal with localization errors, dynamic obstacles, and swarms with more than a thousand agents. The actual experiments were carried out using two different experiment infrastructures: one built at the GRASP Laboratory at the University of Pennsylvania and one at the Universidade Federal de

Minas Gerais (UFMG), Brazil. These experiments suggest that our strategy works well for small groups of robots.

The main limitation of the method is in the number of parameters as the success in the task execution depends on the parameter tuning. Although our method worked well in environments with some narrow passages, such as in Fig. 14, issues may arise if the parameters are not properly selected. The mass, kernel parameter h , and reference density are relevant in this case due to their impact on interrobot spacing. The external force gain k is also relevant as it determines the vehicle bias toward the target and should, in principle, balance the SPH forces in the case of narrow passages to avoid deadlocks. Nevertheless, we found that it was not difficult to tune these variables in our simulations and experiments. In Section VI-A, we detail the tuning procedure that was used to generate our results. Although we do not have an optimal strategy for this process, the results that were obtained with the presented procedure were satisfactory.

Another limitation of the method is the amount of required information. Each robot needs global localization, a complete map of the environment, and the range, bearing, and velocity of neighboring robots. However, these requirements are consistent with most multirobot path planning approaches. We assume that all of this information may be provided by a different layer of software in the robotic architecture.

In principle, our method is not limited by the geometry of the pattern or of the environment as long as the number of robots is adequate, the parameters are properly tuned, and Laplace's equation is numerically solved with the necessary numerical precision.

A possible direction for future work is to consider different cooperative tasks with new fluid models. We are interested in studying how the effects of defining different values of density over the environment may enable applications that require different concentrations of agents in different regions of the environment. Phase transitions may also be an interesting topic to investigate. In the problem of object manipulation, where the robots must transport an object from one point to another, for example, we could require that the fluid transform phase into a solid after the robots have surrounded the object and then transport the object by controlling the vehicles as a solid. We are also interested in studying the impact of actuator saturation and speed limits on the controller performance.

REFERENCES

- [1] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics* (Lecture Notes in Computer Science Series), E. Sahin and W. Spears, Eds. Heidelberg/Berlin: Springer-Verlag, 2005, vol. 3342, pp. 10–20.
- [2] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. Conf. Comp. Graph.*, Assoc. Comput. Mach., 1987, pp. 25–34.
- [3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Intl. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, Mar. 1986.
- [4] R. C. Arkin and T. Balch, "Aura: Principles and practice in review," *J. Exp. Theo. AI*, vol. 9, no. 2–3, pp. 175–189, 1997.
- [5] M. A. Hsieh, L. Chaimowicz, and V. Kumar, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 5, pp. 691–701, Sep. 2008.
- [6] L. Sabattini, C. Secchi, C. Fantuzzi, and D. de Macedo Possamai, "Tracking of closed-curve trajectories for multi-robot systems," in *Proc.*

- IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Taipei, Taiwan, Oct. 2010, pp. 6089–6094.
- [7] L. C. A. Pimenta, M. L. Mendes, R. C. Mesquita, and G. A. S. Pereira, “Fluids in electrostatic fields: An analogy for multi-robot control,” *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1765–1768, Apr. 2007.
- [8] L. C. A. Pimenta, N. Michael, R. C. Mesquita, G. A. S. Pereira, and V. Kumar, “Control of swarms based on hydrodynamic models,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, May 2008, pp. 1948–1953.
- [9] A. Shaw and K. Mohseni, “A fluid dynamic based coordination of a wireless sensor network of unmanned aerial vehicles: 3-D simulation and wireless communication characterization,” *IEEE Sens. J.*, vol. 11, no. 3, pp. 722–736, Mar. 2011.
- [10] D. Lipinski and K. Mohseni, “A master-slave fluid cooperative control algorithm for optimal trajectory planning,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 3347–3351.
- [11] S. Zhao, S. Ramakrishnan, and M. Kumar, “Density-based control of multiple robots,” in *Proc. Amer. Contr. Conf.*, San Francisco, CA, Jun. 2011, pp. 481–486.
- [12] W. Spears, D. Spears, R. Heil, W. Kerr, and S. Hettiarachchi, “An overview of physicomimetics,” *Swarm Robotics* (Lecture Notes in Computer Science Series), E. Sahin and W. Spears, Eds. Heidelberg/Berlin: Springer-Verlag, 2005, vol. 3342, pp. 84–97.
- [13] W. Kerr, D. Spears, W. Spears, and D. Thayer, “Two formal gas models for multi-agent sweeping and obstacle avoidance,” in *Formal Approaches to Agent-Based Systems* (Lecture Notes in Computer Science Series), M. Hinchey, J. Rash, W. Truszkowski, and C. Rouff, Eds. Heidelberg/Berlin: Springer-Verlag, 2005, vol. 3228, pp. 111–130.
- [14] W. Kerr and D. Spears, “Robotic simulation of gases for a surveillance task,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 2905–2910.
- [15] M. Shimizu, A. Ishiguro, T. Kawakatsu, Y. Masubuchi, and M. Doi, “Coherent swarming from local interaction by exploiting molecular dynamics and stokesian dynamics methods,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Las Vegas, NV, May 2003, vol. 2, pp. 1614–1619.
- [16] R. Narain, A. Golas, S. Curtis, and M. C. Lin, “Aggregate dynamics for dense crowd simulation,” *ACM Trans. Graph.*, vol. 28, no. 5, pp. 122:1–122:8, Dec. 2009.
- [17] M. Okada and Y. Homma, “Amenity design for congestion reduction based on continuum model of swarm,” in *Proc. Int. Conf. Mechatron. Tech.*, 2009, pp. 1–6.
- [18] E. Sifakis, S. Marino, and J. Teran, “Globally coupled collision handling using volume preserving impulses,” in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comp. Animat.*, Pasadena, CA, 2008, pp. 147–153.
- [19] J. R. Perkinson and B. Shafai, “A decentralized control algorithm for scalable robotic swarms based on mesh-free particle hydrodynamics,” in *Proc. IASTED Int. Conf. Robot. Appl.*, 2005, pp. 1–6.
- [20] M. R. Pac, A. M. Erkmen, and I. Erkmen, “Control of robotic swarm behaviors based on smoothed particle hydrodynamics,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, San Diego, CA, Oct. 2007, pp. 4194–4200.
- [21] W. J. Pisano, D. A. Lawrence, and K. Mohseni, “Concentration gradient and information energy for decentralized UAV control,” in *Proc. AIAA Guid. Navigat. Contr. Conf.*, 2006, pp. 3283–3296.
- [22] A. Howard, M. J. Matarić, and G. S. Sukhame, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Proc. Int. Sym. Distrib. Auton. Syst.*, Fukuoka, Japan, Jun. 2002, pp. 299–308.
- [23] M. Kloetzer and C. Belta, “Hierarchical abstractions for robotic swarms,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 952–957.
- [24] C. Belta, G. A. S. Pereira, and V. Kumar, “Abstraction and control for swarms of robots,” in *Robotics Research* (Springer Tracts in Advanced Robotics Series), P. Dario and R. Chatila, Eds. Heidelberg/Berlin: Springer-Verlag, 2005, vol. 15, pp. 224–233.
- [25] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *Astronom. J.*, vol. 82, no. 12, pp. 1013–1024, Dec. 1977.
- [26] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics: Theory and application to non-spherical stars,” *Month. Notic. Royal Astronom. Soc.*, vol. 181, pp. 375–389, Nov. 1977.
- [27] G. R. Liu and M. B. Liu, *Smoothed Particle Hydrodynamics—A Meshfree Particle Method*. Singapore, World Scientific, 2003.
- [28] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annu. Rev. Astron. Astrophys.*, vol. 30, pp. 543–574, 1992.
- [29] J. J. Monaghan, “Simulating free surface flow with SPH,” *J. Comput. Phys.*, vol. 110, pp. 399–406, 1994.
- [30] T. M. Roy, “Physically based fluid modeling using smoothed particle hydrodynamics,” Master’s thesis, Graduate College, Univ. Illinois, Champaign, IL, 1995.
- [31] M. A. Hsieh and V. Kumar, “Pattern generation with multiple robots,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May. 2006, pp. 2442–2447.
- [32] C. I. Connolly, J. B. Burns, and R. Weiss, “Path planning using Laplace’s equation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1990, pp. 2102–2106.
- [33] K. P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis, “Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field,” *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, vol. 30, no. 2, pp. 187–196, Mar. 2000.
- [34] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos, “On computing complex navigation functions,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, May 2005, pp. 3463–3468.
- [35] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos, “Robot navigation based on electrostatic field computation,” *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1459–1462, Apr. 2006.
- [36] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.
- [37] C. Belta, G. A. S. Pereira, and V. Kumar, “Control of a team of car-like robots using abstractions,” in *Proc. IEEE Conf. Decis. Contr.*, Maui, HI, Dec. 2003, pp. 1520–1525.
- [38] J. P. Desai, J. Ostrowski, and V. Kumar, “Controlling formations of multiple mobile robots,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, May 1998, pp. 2864–2869.
- [39] B. Stroustrup, *The C++ Programming Language, Third ed.* Reading, MA: Addison-Wesley, 2000.
- [40] D. Shreiner, M. Woo, J. Neider, and T. Davis, *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R), Ver. 2, Fifth ed.* Reading, MA: Addison-Wesley, 2005.
- [41] J. Shewchuk, “Triangle: Engineering a 2D quality mesh generator and delaunay triangulator,” in *Applied Computational Geometry Towards Geometric Engineering* (Lecture Notes Comput. Sci.), M. Lin and D. Manocha, Eds., Heidelberg/Berlin, Germany: Springer-Verlag, 1996, vol. 1148, pp. 203–222.
- [42] D. Meeker, *Finite Element Method Magnetics—User’s Manual*, May 2004, ver. 4.0.
- [43] B. Gerkey, R. T. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *Proc. Int. Conf. Adv. Robot.*, Coimbra, Portugal, Jun. 2003, pp. 317–323.
- [44] N. Michael, J. Fink, and V. Kumar, “Experimental testbed for large multi-robot teams: Verification and validation,” *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 53–61, Mar. 2008.
- [45] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, “The e-puck, a robot designed for education in engineering,” in *Proc. Conf. Auton. Robot. Syst. Competit.*, Castelo Branco, Portugal, May 2009, pp. 59–65.
- [46] R. F. Garcia and L. Chaimowicz, “An experimental infrastructure for swarms of robots,” in *Proc. Brazil. Symp. Intell. Autom.*, Brasilia, Brazil, Sep. 2009, pp. 1–6, (in Portuguese).



Luciano C. A. Pimenta received the B.S., M.Sc., and Ph.D. degrees in electrical engineering from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2003, 2005, and 2009, respectively.

He is currently an Assistant Professor with the Department of Electronic Engineering at UFMG. His research interests include robotics, multirobot systems, and control theory. From April 2007 to June 2008, he was a visiting Ph.D. student with the General Robotics, Automation, Sensing, and Perception Laboratory, University of Pennsylvania, Philadelphia.



Guilherme A. S. Pereira (M'12) received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 1998, 2000, and 2003, respectively.

He was, from November 2000 to May 2003, a Visiting Scientist with the General Robotics, Automation, Sensing, and Perception Laboratory, University of Pennsylvania, Philadelphia. Since July 2004, he has been an Assistant Professor with the Department of Electrical Engineering, Federal University of Minas Gerais, where he is the Director of the Computer Systems and Robotics Laboratory, which is one of the laboratories that compose the Group for Research and Development of Autonomous Vehicles, UFMG. His research interests include cooperative robotics, robot navigation, autonomous vehicles development, computer vision, and distributed sensing.

Dr. Pereira is a Member of Sociedade Brasileira de Automática. He received the Gold Medal Award from the Engineering School of UFMG for garnering first place among the electrical engineering students in 1998.

Dr. Pereira is a Member of Sociedade Brasileira de Automática. He received the Gold Medal Award from the Engineering School of UFMG for garnering first place among the electrical engineering students in 1998.



Nathan Michael (M'09) is an Assistant Research Professor in the Robotics Institute at Carnegie Mellon University. Prior to this appointment, he was a Research Assistant Professor in the Department of Mechanical Engineering at the University of Pennsylvania. He received a Ph.D. from the Department of Mechanical Engineering at the University of Pennsylvania in 2008 and transitioned into a position in the Research Faculty in 2010. His research interests include the topics of estimation and control for ground and aerial robots with extensions to multi-

robot systems.



Renato C. Mesquita was born in Belo Horizonte, Brazil, in 1959. He received the B.S. and M.S. degrees in electrical engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 1982 and 1986, respectively, and the Dr. degree in electrical engineering from the Federal University of Santa Catarina, Florianópolis, Brazil, in 1990.

Since 1983, he has been with the Department of Electrical Engineering, Federal University of Minas Gerais, where he is currently an Associate Professor.

His main research interest is in the area of electromagnetic field computation. He has authored or coauthored more than 150 journal and conference papers in this area.



Mateus M. Bosque received the B.S. degree in control and automation engineering from the Federal University of Minas Gerais, Belo Horizonte, Brazil, in 2010.

He is currently with Federal University of Minas Gerais. He has experience in robotics and computer programming. His current research interests include information technology and the development of computer systems to process financial data.



Luiz Chaimowicz (M'12) received the Ph.D. degree in computer science from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2002.

From 2003 to 2004, he held a Postdoctoral Research Appointment with the General Robotics, Automation, Sensing, and Perception Laboratory, University of Pennsylvania, Philadelphia. He is currently an Associated Professor with the Department of Computer Science, UFMG, where he co-directs UFMG's Computer Vision and Robotics Laboratory, which

conducts research on several aspects of computer vision and mobile robotics. His recent research focuses on the coordination and control of large groups of robots and the development of collision-free navigation strategies for these groups.



Vijay Kumar (F'06) is the UPS Foundation Professor in the School of Engineering and Applied Science at the University of Pennsylvania, and on sabbatical leave at White House Office of Science and Technology Policy where he serves as the assistant director for robotics and cyber physical systems. He received his Bachelors of Technology from the Indian Institute of Technology, Kanpur and his Ph.D. from The Ohio State University in 1987. He has been on the Faculty in the Department of Mechanical Engineering and Applied Mechanics with a secondary appointment in

the Department of Computer and Information Science at the University of Pennsylvania since 1987.

He served as the Deputy Dean for Research in the School of Engineering and Applied Science from 2000-2004. He directed the GRASP Laboratory, a multidisciplinary robotics and perception laboratory, from 1998-2004. He was the Chairman of the Department of Mechanical Engineering and Applied Mechanics from 2005-2008. He then served as the Deputy Dean for Education in the School of Engineering and Applied Science from 2008-2012.

Dr. Kumar is a Fellow of the American Society of Mechanical Engineers (ASME) and the Institute of Electrical and Electronic Engineers. He has served on the editorial boards of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, ASME Journal of Mechanical Design, the ASME Journal of Mechanisms and Robotics and the Springer Tract in Advanced Robotics (STAR). He is the recipient of the 1991 National Science Foundation Presidential Young Investigator award, the 1996 Lindback Award for Distinguished Teaching (University of Pennsylvania), the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, the 2012 ASME Mechanisms and Robotics Award, the 2012 IEEE Robotics and Automation Society Distinguished Service Award and a 2012 World Technology Network Award. He has won best paper awards at DARS 2002, ICRA 2004, ICRA 2011, and RSS 2011 and has advised doctoral students who have won Best Student Paper Awards at ICRA 2008, RSS 2009, and DARS 2010. He is also a Distinguished Lecturer in the IEEE Robotics and Automation Society and an elected member of the Robotics and Automation Society Administrative Committee (2007-2012). His research interests are in robotics, specifically multi-robot systems, and micro aerial vehicles.