A Standardized Testing-Ground for Artificial Potential-Field based Motion Planning for Robot Collectives

Leng-Feng Lee and Venkat Krovi
Mechanical & Aerospace Engineering, State University of New York at Buffalo
318 Jarvis Hall, Buffalo NY 14260 USA
{llee3, vkrovi}@eng.buffalo.edu

Abstract— In this paper, we examine and evaluate artificial-potential field approaches for motion planning of robot collectives with formation-maintenance requirements. To this end, we demonstrate the practical use of construction of the Navigation Function (NF) to serve as the "standardized testing-ground." The NF allows a designer to merge multiple local limited-range potential-functions uniformly into a well-behaved "potential space"- without creating multiple local-minima or irregular scaling at the obstacles. A MATLAB based Graphical User Interface (GUI) is created to aid the interactive creation of the "potential-space" for user-defined workspaces. Within this common test-ground, one is then able to systematically compare and evaluate the performance of various formation maintenance algorithms for robot collectives. In particular, we evaluate the performance of artificial-potential based formation-maintenance algorithms for wheeled mobile robot collectives with 3- and 10-members.

***Keywords*:** *Robot Collectives, Motion Planning, Potential Field.*

I. Introduction

Ongoing revolutions in computing effectiveness and miniaturization of processors/sensors/actuators in the past decade have facilitated the deployment of networked distributed collectives of mobile robots in numerous applications from reconnaissance, foraging, herding to cooperative payload transport.

In recent years, the study of such *groups of multiple autonomous mobile robots exhibiting cooperative behavior* has emerged as an active and challenging research area. Groups in nature (from flocking birds, schooling fish to colonies of ants) appear to make use of a distributed control architecture in which individuals not only respond to their sensed environment (with limited ranges), but also respond to (or are constrained by) the behavior of their neighbors [1, 2]. Recent literature has identified the ability of relatively simple constraints such as: (1) attraction to neighbors up to a maximum distance, (2) repulsion from neighbors if too close, and (3) alignment or velocity matching with neighbors as playing the principal role in maintaining a group formation [1]. In addition, these constraints may also be employed to explain a 'high-level emergent' group behavior such as finding a food source, or move to higher temperature area, while avoiding obstacles (corresponding to a 'gradient climbing' problem). Thus, there is a significant interest within the robotic community to better understand the biological imperative and exploit the same by incorporating similar principles in artificial robot collectives.

However, the diverse application arenas come with varying requirements for "cooperation". Consider a case where a network of robots is to be used in a mapping/reconnaissance mission vs. one where teaming is desired for moving and manipulating large payloads. While deploying a group of robots (over a single robot) has distinct advantages in both cases, the cooperation and coordination requirements are significantly different. Considerable research attention has been focused on the former case where only loose formation maintenance is required [2, 3]. In this paper, however, we examine the latter case which has more stringent requirements on formation design and maintenance.

Artificial Potential Field (APF) approaches provide an intuitive way to model and analyze the behavior of group of robot with many desirable characteristics. In lieu of a deliberative/explicit trajectory or actuator-input computation (prior to the motion), the motion plan is reactively/implicitly specified in terms of the "dynamic-interaction behavior" of the robot system i.e. by how the robot interacts and responds to the sensed environment [4]. At the group level, the workspace is modeled as a "potential space" where the global minimum is at the destination, thus converting group-level motion-planning into a gradient decent problem. At an individual level, interactions of individuals with their neighbors and environment can also be modeled using potential functions, such that formation of the group can be maintained while achieving group motion. Despite these many advantages, APF approaches face the fundamental challenges in hierarchically combining these multiple levels of control within a common framework to realize truly decentralized yet scalable cooperation of such robot collectives.

In this paper, we examine and evaluate adaptation of potential field approaches for motion planning of robot collectives with emphasis on formation maintenance. *The crux is to select or create a "potential space" that can be used as a "standardized test-ground" for studying various formation-control algorithms. The Navigation Function (NF), in our* *opinion, is ideally suited for our implementation in that it allows us to combine the effects of multiple local potential approaches (with limited ranges-of-influence) to create a "well-behaved potential field"*. There are numerous facets to this problem, as summarized pictorially in Fig. 1, that need to be carefully studied. However, in this paper, we focus our attention on sphere worlds with multiple point mass robots and stationary obstacles and target.

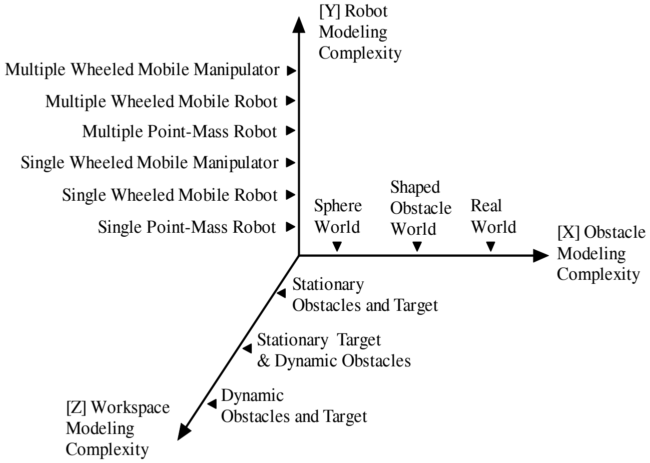

[Y] Robot Modeling Complexity
Multiple Wheeled Mobile Manipulator
Multiple Wheeled Mobile Robot
Multiple Point-Mass Robot
Single Wheeled Mobile Manipulator
Single Wheeled Mobile Robot
Single Point-Mass Robot
Sphere World
Shaped Obstacle World
Real World
[X] Obstacle Modeling Complexity
Stationary Obstacles and Target
Stationary Target & Dynamic Obstacles
Dynamic Obstacles and Target
[Z] Workspace Modeling Complexity

Fig. 1: Challenges entailed in multi-robot motion planning.

The rest of this paper is organized as follows: Section 2 surveys the variety of APF approaches and their key desirable features prior to introducing the navigation function. Section 3 recapitulates the principal formulation features of the navigation function within a convenient MATLAB Graphical User Interface (GUI) interface. Section 4 presents formulation, simulation results and comparative evaluation of potential-based formation- maintenance algorithms for groups of robots operating within such a test-ground. Section 5 concludes the paper with a discussion of the results.

II. Literature Review

In the past decades, *Artificial Potential Field (APF)* methods have gained popularity among robotic researchers especially in the mobile robot arena [5-10] due to its mathematical simplicity and elegance. Beginning in the early 80's, Khatib's work [7] focused on defining a potential field on the configuration space with a minimum at the goal configuration and potential hills at all obstacles. In such a potential field, the robot is attracted to the target while being repelled by obstacles in the workspace (The vector-sum determining the direction and speed of travel). Thus, the *gradient of the combined artificial potential* (from the multiple potential functions) serves as the input force driving the robot to its desired destination while avoiding collisions with obstacles.

Typically, the attractive potential field and the repulsive potential fields are formulated separately, and the total potential field of the workspace is obtained by linear superposition of the two fields. Examples of potential functions designed with this idea are Krogh's GPF function [11], Khatib's FIRAS function [7], Superquadric artificial potential function [12], Ge and Cui's New potential function [13], or the Harmonic potential function [8, 14]. These are called the *local potential approaches* since only local gradient information is needed. Such approaches are very attractive from a computational point of view since no processing is required prior to motion. Further, it is easy to specify a dynamic behavior that tends to avoid obstacles.

The main drawback of such a potential field approach is that when obstacles are present, the potential field may not be convex and local minima that can 'trap' the robot may exist at points away from the target. This local minimum is the result of unpredictable shape of total potential field after the superposition of attractive and repulsive potential fields. The second disadvantage of this approach is that it is difficult to predict the actual trajectory. In practice, one has to carefully pick the constants in the potential function in order to guarantee obstacle avoidance. Furthermore, the generated trajectories are usually far from being optimal by any measure.

Many of later approaches were developed to help overcome some of these limitations. Volpe and Khosla introduced a *Superquadric Artificial Potential Functions* [15], which model a wide range of shapes range from rectangles to ellipses. *Harmonic Potential Functions* [8] seeks to create a potential field without local minima that most potential field methods suffered from. Ge and Cui, proposed a new potential function that take into consideration of moving obstacles and moving target. The potential field is both a function of positions and velocities [6] of the robot, obstacles, and target. Another common problem found in most potential field methods, the GNRON problem (goals non-reachable with obstacle nearby) identified by Ge and Cui [13] and Volpe and Khosla [12], can be handled using their proposed potential fields respectively. Detailed studies of these local potential approaches, their limitations, and their characteristics were presented in [16].

Many of these limitations such as the multiple local minima of the irregular potential scaling at obstacles can be overcome by using the Navigation Function (NF), first introduced by Rimon and Koditschek in their series of papers [10, 17-19]. However, the NF is a 'global' strategy – constructing a configuration-space potential field free of local minima comes at the cost of losing the simplicity and computational advantages of the original local potential fields. Hence, while it may not well-suited for real-time applications, the NF based composite potential-field nevertheless retains many of the other desirable features and serves as an ideal testing-ground for evaluation of motion planning algorithms for robots collectives.

III. Navigation Function (NF)

A. *Properties of the Navigation Function*

An NF is defined as follows: Let Ψ be a robot free configuration space, and let $\mathbf{q}_{Tar}$ be a goal point in the interior of Ψ. A map $\varphi:\Psi \to [0,\ 1]$ is a Navigation Function if it is [10]:

(1) Smooth on Ψ, i.e., at least a $\mathbb{C}^2$ function.

(2) Polar at $\mathbf{q}_{Tar}$, i.e., has a unique minimum at $\mathbf{q}_{Tar}$ on the path-connected component of Ψ containing $\mathbf{q}_{Tar}$.

(3) Admissible on Ψ, i.e., uniformly maximal on the boundary of Ψ.

(4) A Morse Function.

Although a navigation function provide a workspace with a global minimum, an "essential" global convergence is not guaranteed. A global convergence means convergence from almost all initial configurations. In fact, it was shown that there exist at least as many saddle points as there are internal obstacles [10]. However, these spurious unstable equilibrium points need not cause any practical difficulties since only "few" such initial configurations. Further, for a group of robots with formation constraints, the chances of getting stuck on these saddle points is further decreased: if one robot stuck on the saddle point, other robots in the same formation will drive that robot out of the saddle point via the formation constraints.

B. *Navigation function for a sphere world*

For this paper, we focus on construction of an NF in a sphere world. The sphere world is a compact connected subset of E^n whose boundary is formed from the disjoint union of a finite number, $M+1$, of $n-1$ spheres. We largely follow the development in [10, 18] in the rest of the subsection.

Let $\mathrm{A}(\mathbf{q},\rho)$ denote a Euclidean *n*-dimensional disc with center at $\mathbf{q}\in E^n$, and radius ρ. A Euclidean Sphere World is formed by removing from a large *n*-dimensional disc, $A_0(0,\rho_0)$ (i.e. centered at (0, 0) with radius ρ_0), *M*-numbered of disc-like "Hill", $A_j(\mathbf{q}_j,\rho_j)$ for $j=1\cdots M$, called the obstacles. The bounded workspace $E^n - A_0$ is referred as the zeroth obstacle. The free configuration space (or simply, configuration space) ,Ψ that remains after removing all the internal obstacles from A_0 is:

$$\Psi = A_0 - \bigcup_{j=1}^{M} Obstacle_j \tag{1}$$

For Ψ to be a valid sphere world, the obstacles' closure must be disjoint and be contained in the interior of A_0. Thus, for this sphere world, there are $M+1$ centers, q_i and radii ρ_i, for $i=1\cdots M+1$. Also, for this example, the Bounding Function is:

$$\beta_0(q) = -\left\|\mathbf{q}-\mathbf{q}_0\right\|^2 + \rho_0^2 \tag{2}$$

and spherical obstacle function given by:

$$\beta_j(q) = -\left\|\mathbf{q}-\mathbf{q}_j\right\|^2 + \rho_j^2\ ,\quad for\ j=1\cdots M. \tag{3}$$

These formulas are expressed in terms of the implicit representation of the constituent shape, which assumed to be known. We are now ready to construct the NF for the sphere world which defined as:

$$\varphi_\kappa(\mathbf{q}) = \begin{cases} \left(\rho_\kappa \circ \sigma_1 \circ \dfrac{\gamma_\kappa}{\beta}\right)(\mathbf{q}) & \text{for } \beta > 0 \\ 1 & \text{for } \beta \le 0 \end{cases} \tag{4}$$

where γ_κ is the distance-to-target function, given by:

$$\gamma_\kappa(\mathbf{q}) = \left\|\mathbf{q}-\mathbf{q}_{Tar}\right\|^{2\kappa} \tag{5}$$

In which $\mathbf{q}$ denoted the position of the robot, $\mathbf{q}_{Tar}$ denoted the position of the target, and $\left\|\mathbf{q}-\mathbf{q}_{Tar}\right\|$ is the Euclidean norm and $\kappa > 0$ is a control parameter. β in Eq.(4) denotes the product of obstacle function which is given by:

$$\beta = \prod_{i=0}^{M} \beta_i \tag{6}$$

where β_i is given in Eq.(2) and Eq.(3). On the other hand, $\sigma_\lambda(x) = x/(\lambda + x)$ and $\rho_\kappa(x) = x^{1/\kappa}$ are called the analytic switch function and sharpening function respectively, and they are both conditioning functions. For example, the analytic switch function $s(q,\lambda)$ performs the following operation:

$$s(\mathbf{q},\lambda) = \left(\sigma_\lambda \circ \frac{\gamma}{\beta}\right)(\mathbf{q}) = \frac{\gamma(\mathbf{q})}{\lambda\beta(\mathbf{q}) + \gamma(\mathbf{q})} \tag{7}$$

Equation (7) has the following properties: it vanishes exactly at the zeros of γ, achieves its upper bound of unity exactly at the zeroes of β, and varies smoothly between the two elsewhere. Now using the sharpening function, combined with Eq.(7), we obtained the following NF in the sphere world:

$$\varphi_\kappa(\mathbf{q}) = \begin{cases} \dfrac{\left\|\mathbf{q}-\mathbf{q}_{Tar}\right\|^2}{\left[\left\|\mathbf{q}-\mathbf{q}_{Tar}\right\|^{2\kappa} + \beta(\mathbf{q})\right]^{1/\kappa}} & \text{for } \beta > 0 \\ 1 & \text{for } \beta \le 0 \end{cases} \tag{8}$$

The value of $\varphi_\kappa(\mathbf{q})$ varies between $[0,\ 1]$ for $\beta > 0$. β is the product of obstacle functions as defined in Eq.(6), $\beta \le 0$ constitute the points "inside" the obstacle, and thus gives a maximum value of 1.

Here, we see that navigation function provides a solution to the local minima found in local potential field approaches; nevertheless it remains a global method. This means that it losses the advantages of local potential fields approach where only local information is needed. Further, while the repulsive potential function in local potential approaches have a limited range of influence (which is a desired feature), all obstacles in the navigation function contributes to the final shape of the potential field, no matter how far they located.

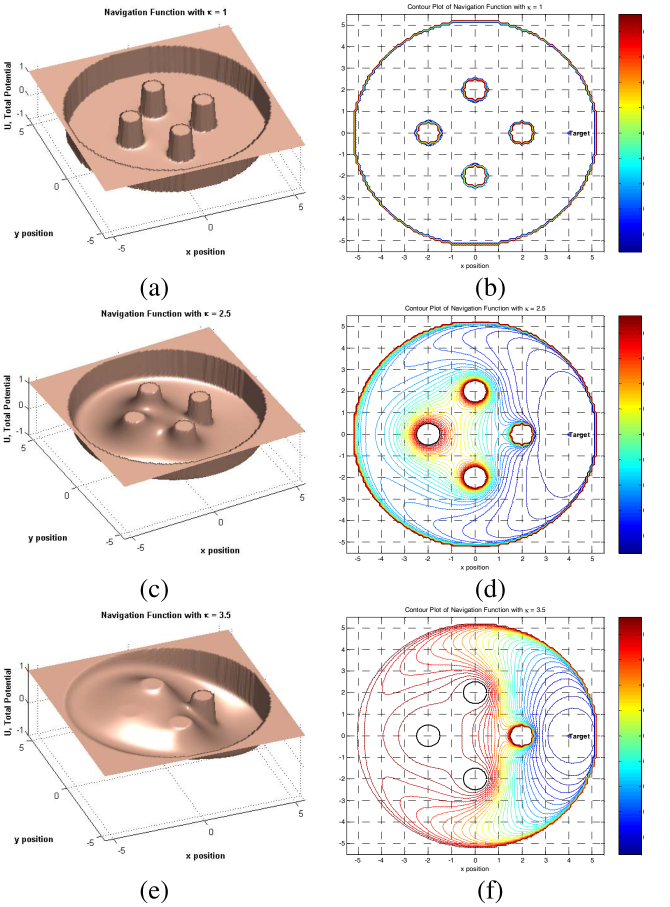


Fig. 2: The 3D visualization of a navigation function of a sphere world with four obstacles with (a) $\kappa = 1.0$; (c) $\kappa = 2.5$; and (e) $\kappa = 3.5$, and their corresponding contour plots are given in (b), (d), and (f).

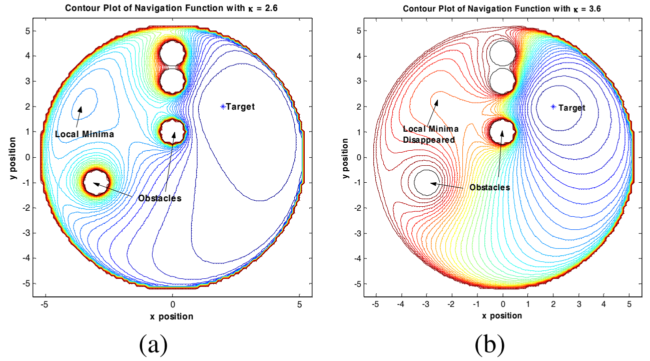
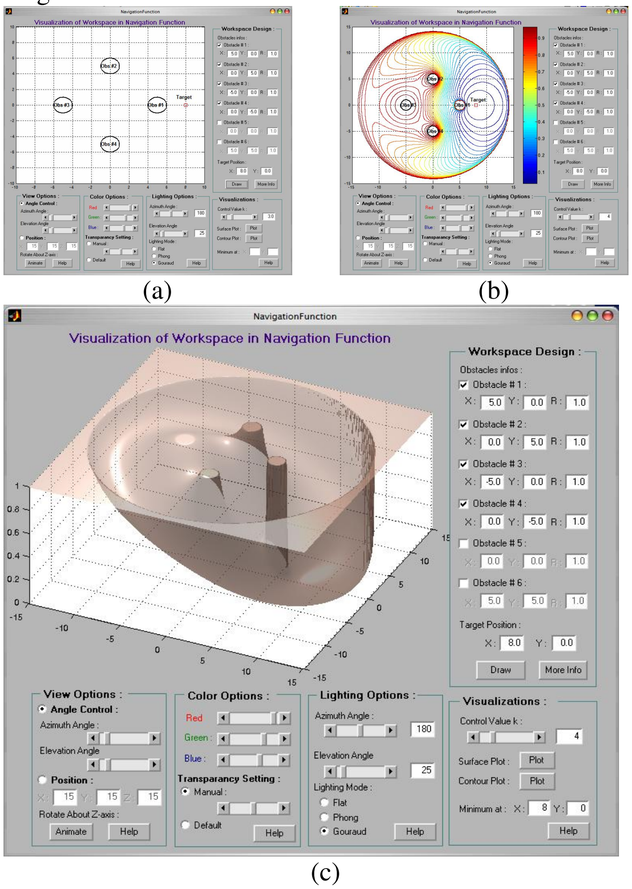


Fig. 3: Contour plot of a potential field for a workspace with four obstacles created using navigation function. In (a), local minima exist with a κ value of 2.6; and (b) Local minima disappeared as κ value increased to 3.6.

Further, Eq.(8) has the characteristic that it can provide a potential field with a unique minimum adjustable using a single tunable parameter κ . As an example, 3D potential field of a workspace with four obstacles and their contour are plotted in Fig. 2. In these figures, obstacles are located at $(2,\ 0)$, $(0,\ 2)$, $(-2,\ 0)$, and $(0,\ \text{-}2)$, with a radius of 0.5 , and target is located at $(4,\ 0)$. Fig. 2(a)-(f) show that potential field generated from navigation function has a smooth contour only with a proper selected κ value. For some arrangements of the obstacles, undesired local minima may exist at low value of κ . For example, Fig. 3(a) shows the contour plot where a κ value of 2.6 results in an undesirable local minima at (-3.5, 2); which disappears as κ value is increased to 3.6 as shown in Fig. 3(b). Therefore, selecting a suitable κ value is critical but can be done offline. Hence, we develop a MATLAB GUI to aid the designer.

C. *Design of a MATLAB GUI for Navigation Function*

In the previous section, we have shown that an "adequate" κ value is necessary for finding a potential field without local minima – however, this value can vary significantly depending upon the workspace and the number, size and positioning of the various obstacles. Hence, it was useful to develop a tool to quickly determine the necessary κ value for a given workspace in order to subsequently evaluate motion planning algorithms for groups-of-robots. The interactive MATLAB-based Graphical User Interface (GUI), shown in Fig. 4, allow the designer to alter the size, number and location of various obstacles, change the κ value, visualize the contours, and interact with a 3D plot of the navigation function.



☑ Obstacle # 1 : X: 5.0 Y: 0.0 R: 1.0
☑ Obstacle # 2 : X: 0.0 Y: 5.0 R: 1.0
☑ Obstacle # 3 : X: -5.0 Y: 0.0 R: 1.0
☑ Obstacle # 4 : X: 0.0 Y: -5.0 R: 1.0
□ Obstacle # 5 : X: 0.0 Y: 0.0 R: 1.0
□ Obstacle # 6 : X: 5.0 Y: 5.0 R: 1.0
Target Position : X: 8.0 Y: 0.0

● Angle Control :
○ Position : X: 15 Y: 15 Z: 15

Transparency Setting : ● Manual : ○ Default

Lighting Options : Azimuth Angle : 180 Elevation Angle 25
Lighting Mode : ○ Flat ○ Phong ● Gouraud

Visualizations : Control Value k : 4
Minimum at : X: 8 Y: 0

Fig. 4: The MATLAB GUI that allows a designer: (a) place obstacles of various sizes in the workspace; (b) visualize the contour of the navigation function of the workspace; and (c) visualize the potential field in 3D, and change κ value to obtain a smooth potential field with a unique minimum.

IV. Formulation and Simulation Results

In this paper, we implemented potential field approaches for motion planning of robot collectives. A "team" of differentially-driven nonholonomic wheeled mobile robots (NH-WMR) are shown in Fig. 5(a). Typically, the nonholonomic constraints of the individual wheels combine with the differentially driven architecture to limit the possible motion of such robots. Thus, any point along the wheel axle of the differentially driven wheels cannot move in the direction of the axis. However, all other points are not bound by this constraint which allows reduction of the NH-WMRs to an equivalent point mass, as is done in this paper.

A. Dynamic Equations Formulation

The dynamic equation of this group of n point-mass robot *without* formation maintenance is given by:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{u} - \mathbf{K}\dot{\mathbf{q}} \tag{9}$$

where $\mathbf{M} = diag\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, a $2n \times 2n$ diagonal mass matrix; $\mathbf{u} = -\mathbf{K}_f \nabla_{\mathbf{q}} U$, the input to the system is the negative gradient of the potential field U, and $\mathbf{K} = diag\{k_{x_1}, k_{y_1}, \cdots, k_{x_n}, k_{y_n}\}$, also a $2n \times 2n$ diagonal control matrix and $\nabla_{\mathbf{q}} U = \frac{\partial U}{\partial \mathbf{q}} = \left[\frac{\partial U}{\partial x_1}, \frac{\partial U}{\partial y_1}, \frac{\partial U}{\partial x_2}, \frac{\partial U}{\partial y_2}, \cdots, \frac{\partial U}{\partial x_n}, \frac{\partial U}{\partial y_n}\right]^T$. Equation (9) can also be written in the following matrix form:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\mathbf{M}^{-1}\mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{u} \tag{10}$$

with $\mathbf{u} = -\mathbf{K}_f \nabla_{\mathbf{q}} U$.

To include the formation maintenance in the dynamics equations, we note that this group of independent mobile robots moving together in formation and coupled together by constraint dynamics can alternatively be viewed as a constrained mechanical system. Hence, the dynamics of group of robots can be formulated as Lagrange Equation of the First Kind as:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{11}$$

$$\mathbf{M}(\underset{\sim}{\mathbf{q}})\dot{\mathbf{v}} = \mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u}) - \mathbf{J}(\mathbf{q})^T \boldsymbol{\lambda} \tag{12}$$

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \tag{13}$$

where $\mathbf{q}$ is the n-dimensional vector of generalized coordinates; $\mathbf{v}$ is the n-dimensional vector of generalized velocities; $\mathbf{M}(\mathbf{q})$ is the $2n \times 2n$ dimensional inertia matrix; $\mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u})$ is the n-dimensional vector of external forces; $\mathbf{u}$ is the vector of input forces, which is $-\mathbf{k}_f \nabla_{\mathbf{q}} U$; $\mathbf{C}(\mathbf{q}, t)$ is a m -dimensional vector of holonomic constraints; and $\mathbf{J}(\mathbf{q}) = \partial \mathbf{C}(\mathbf{q}) / \partial \mathbf{q}$ is the *Jacobian* matrix.

As a result, the formulation and computation of motion plans for such collectives in a potential field may be treated as being equivalent to simulating the forward dynamics of a constrained multi-body mechanical system. By doing so, we can link and leverage the extensive literature on formulation and implementation of computational simulation of multibody systems [20-23]. Specifically, the constrained dynamics system may now be solved by using three methods: (i) direct Lagrange multiplier elimination approach [22, 24]; (ii) Penalty formulation approach [25]; or (iii) Constraint manifold projection approach [26-28]. In particular, the penalty formulation approach is very closely analogous to the potential field approach in [2, 3] and will be investigated in our work. A detailed treatment of the other two approaches can be found in [29].

In the penalty-formulation approach, the holonomic constraints are relaxed and replaced by linear/non-linear spring with dampers. This allows the constraint equations to be incorporated as an auxiliary dynamical system penalized by a large factor (as shown in Fig. 5(a)). Here, the Lagrange multipliers are explicitly approximated as the force of a virtual spring or damper based on the extent of the constraint violation and assumed spring stiffness and damping constant. The restoring force, which is proportional to the extent of the constraint violation, is expressed as:

$$\boldsymbol{\lambda} = \mathbf{K}_S \mathbf{C}(\mathbf{q}) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q}) \tag{14}$$

where $\mathbf{K}_S$ is the spring constant term, and is given by $\mathbf{K}_S = diag\{K_{S1}, K_{S2}, \cdots, K_{Sn}\}$, an $n \times n$ diagonal matrix; $\mathbf{K}_D$ is the damping constant term, where $\mathbf{K}_D = diag\{K_{D1}, K_{D2}, \cdots, K_{Dn}\}$, also an $n \times n$ diagonal matrix; and $\mathbf{C}(\mathbf{q})$ is the vector of constraint violation in the direction of the respective $\boldsymbol{\lambda}$. Substituting Eq.(14) into Eq.(12), the dynamic equation using penalty-formulation can be expressed as:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{M}^{-1}\left[\mathbf{f}(\mathbf{q}, \mathbf{v}, t, \mathbf{u}) - \mathbf{J}^T\left(\mathbf{K}_S \mathbf{C}(\mathbf{q}) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q})\right)\right] \end{bmatrix} \tag{15}$$

By doing this, the Lagrange multiplier is eliminated from the list of $n+m$ unknowns, leaving a system of $2n$ first order ODEs. On one hand, we note that this may creates a stiff dynamic equation with poor numerical conditioning, when a large penalty factor is selected. On the other hand, this spring force only approximates the true value of the constraint forces. Thus it may not be able to maintain tight formation, if a relative small penalty factor is selected. Note that with this formulation, we can also allow for shape change by letting:

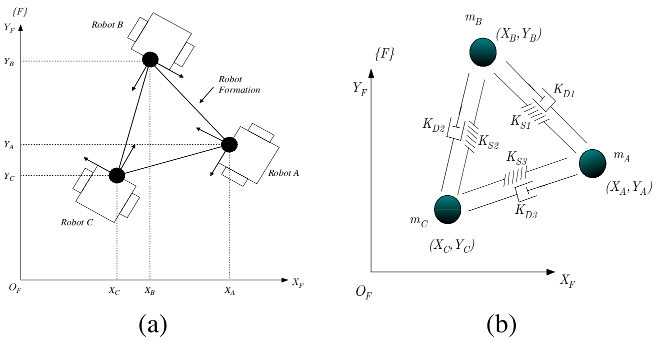
$$\boldsymbol{\lambda} = \mathbf{K}_S \mathbf{C}(\mathbf{q}, t) + \mathbf{K}_D \dot{\mathbf{C}}(\mathbf{q}, t) \tag{16}$$

where the constraint matrix $\mathbf{C}$ is now $\mathbf{C}(\mathbf{q}, t)$, a function of both position and time. Another advantage of approximating the Lagrange multiplier using the penalty approach is that fully decentralized formulation can be obtained without much effort. For example, the dynamics equation of a group of three point-mass robots, denote as A, B, and C ($n = 3$), can be formulated using Eq.(16), and further decentralized as:

$$\begin{bmatrix}\dot{\mathbf{q}}_A\\ \ddot{\mathbf{q}}_A\end{bmatrix}_{4\times1}=\begin{bmatrix}\mathbf{v}_A\\ \mathbf{M}_A^{-1}\left[\mathbf{E}_A\mathbf{u}_A-\mathbf{K}_A\dot{\mathbf{q}}_A-\mathbf{J}_A^T\left(\mathbf{K}_{S,A}\mathbf{C}+\mathbf{K}_{D,A}\dot{\mathbf{C}}_A\right)\right]\end{bmatrix}$$
$$\begin{bmatrix}\dot{\mathbf{q}}_B\\ \ddot{\mathbf{q}}_B\end{bmatrix}_{4\times1}=\begin{bmatrix}\mathbf{v}_B\\ \mathbf{M}_B^{-1}\left[\mathbf{E}_B\mathbf{u}_B-\mathbf{K}_B\dot{\mathbf{q}}_B-\mathbf{J}_B^T\left(\mathbf{K}_{S,B}\mathbf{C}+\mathbf{K}_{D,B}\dot{\mathbf{C}}_B\right)\right]\end{bmatrix}\qquad(17)$$
$$\begin{bmatrix}\dot{\mathbf{q}}_C\\ \ddot{\mathbf{q}}_C\end{bmatrix}_{4\times1}=\begin{bmatrix}\mathbf{v}_C\\ \mathbf{M}_C^{-1}\left[\mathbf{E}_C\mathbf{u}_C-\mathbf{K}_C\dot{\mathbf{q}}_C-\mathbf{J}_C^T\left(\mathbf{K}_{S,C}\mathbf{C}+\mathbf{K}_{D,C}\dot{\mathbf{C}}_C\right)\right]\end{bmatrix}$$

where $\mathbf{u}_i=\nabla_{q_i}U$, $\dot{\mathbf{C}}_i=[\mathbf{J}_i][\mathbf{q}_i]$, and $K_{S,i},\ K_{D,i}$ with $i=A,B,C$ are the compliance matrices for springs and dampers respectively.

The three dynamic sub-systems shown in Eq.(17), can be simulated in a distributed manner if at every time step: (1) either the information pertaining to $\mathbf{C}(\mathbf{q})$, the extend of the constraint violation, is made available explicitly or (2) computed by exchanging information between the robots. The sole coupling between the three sub-parts is due to the Lagrange multipliers, which are now explicitly calculated using the virtual spring.



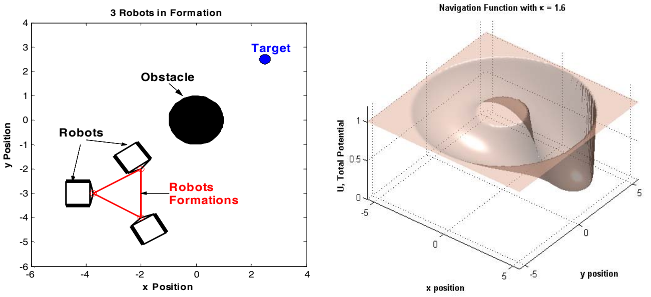
(a) (b)

Fig. 5: (a) Three nonholonomic wheeled mobile robot in a triangular formation can be treated as three point mass robots by selecting a point away from the axle; and (b) The formation constraint is satisfied by approximating the Lagrange multiplier as springs and dampers.

B. Case studies

We performed three case studies using the formulation presented above on the common test course, shown in Fig. 6(a). Initially, we positioned three robots at $(-2,-3)$, $(-2,-4)$, and $(-2.866,-3.5)$ respectively with respect to an obstacle of radius 1.5 at the origin and the target located at $(2.5,\ 2.5)$. The potential field of the workspace, created using navigation function with $\kappa=1.6$, is shown in Fig. 6(b).

We performed the simulation using MATLAB's fixed time-step solver, in consideration of actual implementation, where the information from sensors is evaluated in a specific fixed time-intervals. In particular, ODE5 (a fifth order Runge-Kutta Method) was chosen for its highest accuracy among other fixed time-step solvers. A fixed time step of 1×10^{-3} seconds was used for the total simulation time of 8 seconds.



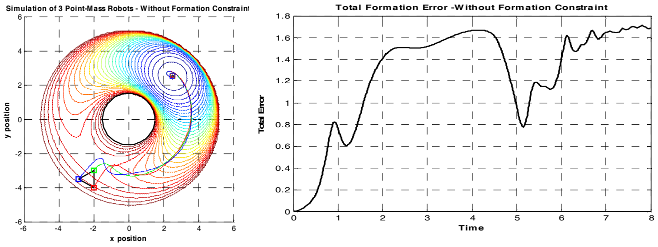
(a) (b)

Fig. 6: (a) The simulation setup for case studies, shown here an obstacle between a group of three point-mass robots and the target location; and (b) the potential field of the workspace generated using the navigation function in 3D view.

As a baseline reference, the first case study is done using the dynamics equation *without incorporating the formation constraints*, as given in Eq.(10). The gradient information at each simulation step is obtained numerically for each robot. As the simulation shows, in the absence of the formation constraint the results depict 3 individual mobile robots converging to the common minimum. To quantify how well the formation is maintained, we study the total formation error, is given by:

$$\Delta_{Error}=\sqrt{\sum_{i=1}^{M}\left(c-\bar{c}\right)_i^2}\qquad(18)$$

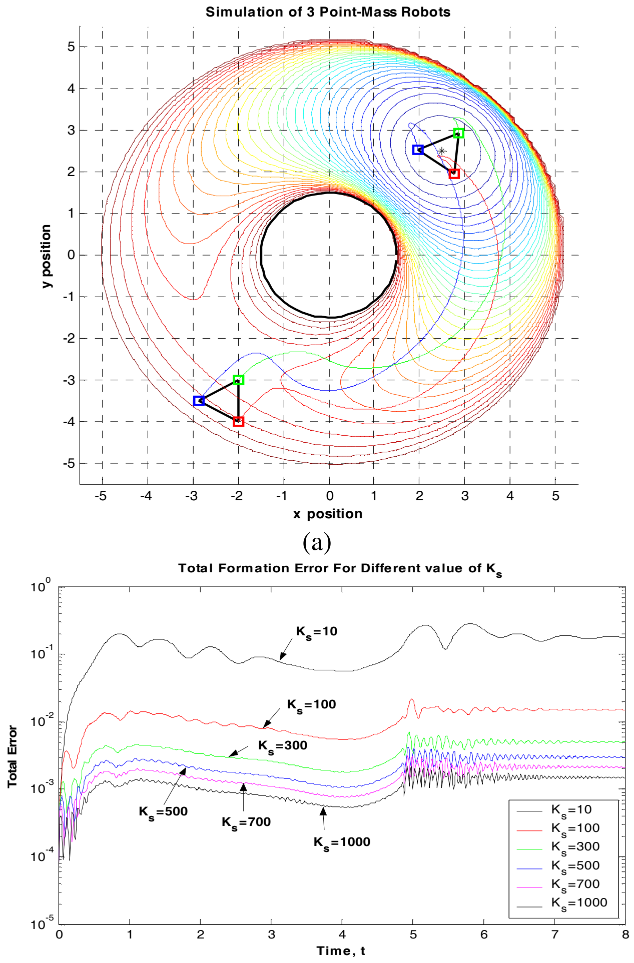
where Δ_{Error} denote the total formation error, M denote number of holonomic constraints in the equation, c is the Euclidean distance of each holonomic constraint at each instant, and $\bar{c}$ is the desired Euclidean distance for each holonomic constraint. Fig. 8 depicts the results of the second case study – where the constrained equations in Eq.(15) are simulated under identical conditions.



(a) (b)

Fig. 7: (a) Simulation result showing the three robots traveling to the target by following the negative gradient of the potential, without formation maintenance; and (b) The total formation error throughout the simulation.

We notice that the formation error shows significant dependence on the value of the spring stiffness K_S and is order of 10^{-2} for $K_S=100$ .To further examine formation constraint maintenance, we study the parametric effect of raising the value of K_S. Fig. 8(b) shows the reduction in the total formation error as the value of K_S increases. We obtain a total formation error in the order of 10^{-3} with $K_S=1000$, with a simulation time-step of 10^{-3}. However, for value of $K_S > 1000$, the simulation become unstable as the equation stiffness increases requiring a reduction in the simulation step size. Several other studies, including formation-expansion and formation topological changes were also performed and discussed in [16].

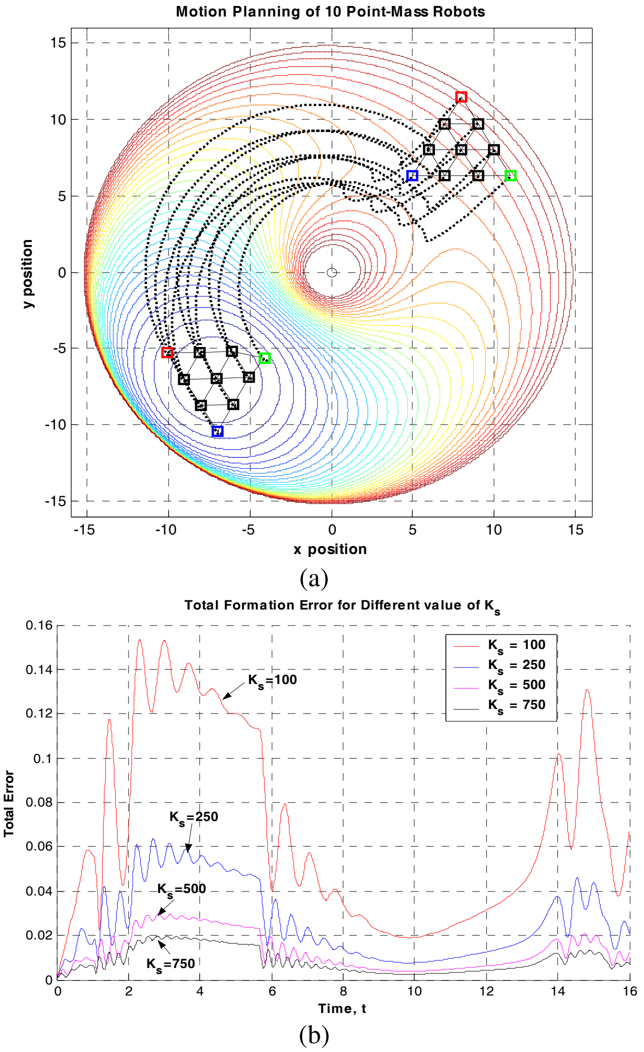


(a)

(b)

Fig. 8: (a) The simulation result showing the three robots in a triangular formation move from their initial position to the target position while maintaining formation; and (b) The total formation error for different value of spring constant K_S of the three robots in triangular formation case study.

For this paper, we also report a third case study, with 10 point-mass robots moving in formation that was performed to help study the scalability of this formulation. The initial locations of the robots are given by: (8, 11.46), (7, 9.73), (9, 9.73), (6, 8), (8, 8), (10, 8), (5, 6.26), (7, 6.26), (9, 6.26), (11, 6.26). The target is locate at (-7, -7), the obstacle is located at (0, 0) with a radius of 1.5, and the potential field is created using navigation function with $\kappa = 1.7$. The sample simulation result, shown in Fig. 9(a) and the study of the parametric changes in spring constant K_S on the total formation error, shown in Fig. 9(b), are very similar to the previous case study.



(a)

(b)

Fig. 9: (a) The simulation result of the 10 robots forming a interconnected triangular formation in a workspace with one obstacle; and (b) The total formation error plot for different value of K_S used in this simulation.

Hence, in practice, the selection of this spring constant value become critical as one tries to reconcile conflicting requirements for tight formation maintenance and avoiding ill-conditioned (stiff) formulations. Other approaches, such as constraint manifold projection [16, 29] exist but tend to be less amenable for decentralized implementations and/or more expensive in terms of computations-per-iteration. However, it is oftentimes more desirable to use these approaches in terms of their reduced overall formulation stiffness which makes for faster and stable overall computations.

IV. Discussion & Conclusion

In this paper, we demonstrate the practical use of construction of a navigation function as the "standardized potential space" for evaluation of multi-robot cooperative payload transport algorithms. The navigation function provides a means to merge various limited-range local-potential-fields into a "well-behaved" C-space potential function with a unique global minimum (that can be tuned using a parameter κ), and other useful characteristics. However, this comes at the cost of requiring complete knowledge of the workspace in order to construct the navigation function.

An interactive GUI interface was developed to alleviate the mathematical/computational complexity and tedium involved in merger of arbitrary local-limited-range potential functions into the navigation function. The tool allows us to quickly model the workspace and obstacle environment, select an appropriate κ value, and visualize the resulting navigation functions. Such standardization of testing-grounds allows algorithms for coordinated control of multi-robot collectives to be evaluated. This, for example, allowed us to study effects of potential/behavior-based constraints on performance of formation-maintenance operations. Other similar studies, leveraging this framework, are also currently underway.

ACKNOWLEDGEMENT

We gratefully acknowledge the support from The Research Foundation of State University of New York and National Science Foundation CAREER Award (IIS-0347653) for this research effort.

REFERENCES

[1] Ogren, P., Egerstedt, M., and Hu., X., "A Control Lyapunov Function Approach to Multi-Agent Coordination," IEEE Transactions on Robotics and Automation, vol. 18, pp. 847-851, 2002.

[2] Ogren, P., Fiorelli, E., and Leonard, N. E., "Cooperative Control of Mobile Sensor Networks:Adaptive Gradient Climbing in a Distributed Environment," IEEE Transactions on Automatic Control, vol. 49, pp. 1292-1302, 2004.

[3] Ogren, P. and Leonard, N. E., "Obstacle Avoidance in Formation," in Proceedings of IEEE International Conference on Robotics and Automation, 2003, pp. 2492-2497.

[4] Kumar, V., Zefran, M., and Ostrowski, J., "Intelligent Motion Planning and Control," in Handbook of Industrial Robotics, S. Nof, Ed.: John Wiley and Sons, 1999.

[5] Cao, Y. U., Fukunaga, A., and Kahng, A., "Cooperative Mobile Robotics: Antecedents and Directions," Autonomous Robots, vol. 4, pp. 1-23, 1997.

[6] Ge, S. S. and Cui, Y. J., "Dynamic Motion Planning for Mobile Robots Using Potential Field Method," Autonomous Robots, vol. 13, pp. 207-222, 2002.

[7] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," International Journal of Robotic Research, vol. 5, pp. 90-98, 1986.

[8] Kim, J.-O. and Khosla, P. K., "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," IEEE Transactions on Robotics and Automation, vol. 8, pp. 338-349, 1992.

[9] Leonard, N. E. and Fiorelli, E., "Virtual Leaders, Artificial Potentials and Coordinated Control of Groups," in Proceedings of IEEE Conference on Decision and Control, 2001, pp. 2968-2973.

[10] Rimon, E. and Koditschek, D. E., "Exact Robot Navigation Using Artificial Potential Functions," IEEE Transactions on Robotics and Automation, vol. 8, pp. 501-518, 1992.

[11] Krogh, B., "A Generalized Potential Field Approach to Obstacle Avoidance Control," in Proceedings of ASME Conference of Robotic Research: The Next Five Years and Beyond, 1984.

[12] Volpe, R. and Khosla, P., "Manipulator Control with Superquadric Artificial Potential Functions: Theory and Experiments," IEEE Transactions on Systems, Man and Cybernetics, vol. 20, pp. 1423-1436, 1990.

[13] Ge, S. S. and Cui, Y. J., "New Potential Functions for Mobile Robot Path Planning," IEEE Transactions on Robotics and Automation, vol. 16, pp. 615-620, 2002.

[14] Connolly, C. I., "Applications of Harmonic Functions to Robotics," in Proceedings of IEEE International Symposium on Interlligent Control, 1992, pp. 498-502.

[15] Khosla, P. and Volpe, R., "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," in Proceedings of IEEE International Conference on Robotics and Automation, 1988, pp. 1778-1784.

[16] Lee, L.-F., "Decentralized Motion Planning Within an Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-Robot Collectives.," Mechanical & Aerospace Engineering, State University of New York at Buffalo, NY, Buffalo, 2004.

[17] Koditschek, D. E. and Rimon, E., "Robot Navigation Functions on Manifolds with Boundary," Advances in Applied Mathematics, vol. 11, pp. 412-442, 1990.

[18] Rimon, E. and Koditschek, D. E., "The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Sphere Worlds," in Proceedings of IEEE International Conference on Robotics and Automation, 1989, pp. 21-26.

[19] Rimon, E. and Koditschek, D. E., "The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds," Transactions of the American Mathematical Society, vol. 327, pp. 71-115, 1991.

[20] Arnold, V., Mathematical Methods of Classical Mechanics, 2 ed. New York: Springer-Verlag, 1989.

[21] Ascher, U., Chin, H., Petzold, L., and Reich, S., "Stabilization of Constrained Mechanical System with DOEs and Invariant Manifolds," Mechanical Structures & Machines, vol. 23, pp. 135-158, 1995.

[22] García de Jalón, J. and Bayo, E., Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge. New York: Springer-Verlag, 1994.

[23] Haug, E. J., Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods: Allyn and Bacon Series in Engineering, Prentice Hall, 1989.

[24] Witkin, A., Gleicher, M., and Welch, W., "Interactive Dynamics," in Proceedings of 1990 Symposium on Interactive 3D Graphics, 1990, pp. 11-21.

[25] Wang, J., Gosselin, C. M., and Cheng, L., "Modeling and Simulation of Robotic Systems with Closed Kinematic Chains using the Virtual Spring Approach," Multibody System Dynamics, vol. 7, pp. 145-170, 2000.

[26] Khan, W. A., Tang, C. P., and Krovi, V., "Modular and Distributed Forward Dynamic Simulation of Constrained Mechanical Systems - A Comparative Study," Mechanism and Machine Theory, 2006. (In press)

[27] Sarkar, N., Yun, X., and Kumar, V., "Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots," International Journal of Robotic Research, vol. 13, pp. 55-69, 1994.

[28] Yun, X. and Sarkar, N., "Unified Formulation of Robotic Systems with Holonomic and Nonholonomic Constraints," IEEE Transactions on Robotics and Automation, vol. 14, pp. 640-650, 1998.

[29] Lee, L.-F., Bhatt, R. M., and Krovi, V., "Comparison of Alternate Methods for Distributed Motion Planning of Robot Collectives within a Potential-Field Framework," in Proceedings of IEEE International Conference on Robotics and Automation, 2005.