Multi-Robot Coordination using Generalized Social Potential Fields

Russell Gayle William Moss Ming C. Lin Dinesh Manocha
Department of Computer Science
University of North Carolina at Chapel Hill

***Abstract*— We present a novel approach to compute collision-free paths for multiple robots subject to local coordination constraints. More specifically, given a set of robots, their initial and final configurations, and possibly some additional coordination constraints, our goal is to compute a collision-free path between the initial and final configuration that maintains the constraints. To solve this problem, our approach generalizes the social potential field method to be applicable to both convex and nonconvex polyhedra. Social potential fields are then integrated into a "physics-based motion planning" framework which uses constrained dynamics to solve the motion planning problem. Our approach is able to plan for over 200 robots while averaging about 110 ms per step in a variety of environments.**

I. Introduction

Multiple robots and agents are increasingly used in different applications to cooperatively solve complex tasks. These include assembly, manufacturing, swarm or distributed robotics, virtual prototyping, animation, traffic engineering, crowd simulation, etc. Given a set of rigid or articulated robots with their initial and final configurations, the goal is to compute a collision-free path for each robot. We assume that an accurate geometric model of the robot and obstacles is available to us and that we also know their position precisely.

As compared to single-robot motion or path planning, the multiple-robot problem is more challenging in terms of theoretical complexity as well practical implementations. One of the main challenges is coordination among the different robots, as they attempt to reach the goal configurations or complete a task. This can be in the form of collision avoidance, grouping and formations, or collectively carrying out a task. Various factors contribute to this added complexity of multi-robot planning. Most notably, in a multi-robot system each robot's degrees of freedom (DOFs) contribute to the total number of DOFs of the system. This property makes it challenging to develop a complete planner even for systems with only a few robots.

Prior techniques for motion planning and coordination in multi-robot systems can be classified into *centralized* and *decentralized* methods. The centralized methods group the robots into one large system and reduce the problem to planning the motion of a single composite robot with high DOFs. Coordination requirements can further constrain the search space and result in rather narrow passages. Decentralized methods reduce this overall complexity by exploiting the independent nature of DOFs between robots. Typically, each robot initially plans a path independently of the other robots. Then, a coordination step is required to ensure that different robots do not collide with each other while traversing their paths. However, the overall approach is susceptible to reliability issues and may not work well in practice [1], [2].

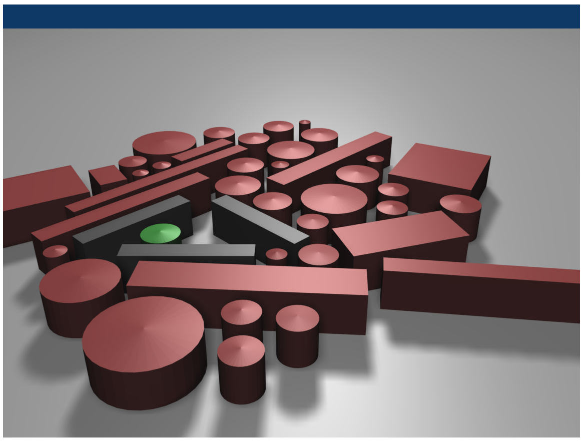

Fig. 1. **Guarding and escorting**: This image is from a guarding and escorting scenario. 35 aggressive robots (in red) are trying to reach an important robot (in green). The black robots are attempting to stay in a formation while protecting the important robot. Their goal is to escort it across the environment. Our social force model allows the aggressive robot to move toward the important robot while avoiding collisions and distance coordination constraints are used to help maintain the formation. This benchmark took 132.9 s with an average step time of 42 ms.

A. Main Results

In this paper, we present an efficient approach that computes collision-free paths for multiple robots and maintains local coordination constraints. Specifically, we address the problem: Given n robots with initial and goal configurations as well as coordination constraints, find a collision-free path from the initial to goal configurations that maintains the constraints. This work considers coordination constraints including general repulsive or attractive behaviors, grouping, formations, local navigation, collision avoidance, and distance constraints. While much of the related work for multiple robots assumes point, point-like, circular or cylindrical robot primitives, our approach can handle arbitrary robot and obstacle shapes, including non-convex polyhedra.

This work was supported in part by ARO Contracts DAAD19-02-1-0390 and W911NF-04-1-0088, NSF awards 0400134, 0429583 and 0404088, DARPA/RDECOM Contract N61339-04-C-0043 and Intel. The first author is also supported by a Department of Energy High-Performance Computer Science Fellowship administered by the Krell Institute.

{rgayle,wmoss,lin,dm}@cs.unc.edu
http://gamma.cs.unc.edu/CSP

Like other decentralized approaches, each robot plans and coordinates its motion independently from other robots. Instead of a separate coordination phase, our approach interleaves coordination with motion generation through the use of coordination constraints.

Our approach builds upon physics-based motion planning (PMP) [3], [4] and social potential fields (SPF) [5], [6]. PMP solves motion planning problems through the use of a constraint-directed physical simulation. Social potential fields are a variation of the potential field methods whose resulting motion appears to exhibit various social behaviors.

We generalize and reformulate social potential fields to be integrated with the PMP framework and also to be applicable to a more general robot primitive. The social potentials will act as a constraint on the system, allowing the robot to change its current trajectory and move toward or away from other robots or obstacles. The PMP framework supports additional constraints, such as creating and maintaining group formations and other behaviors. To reduce the effects of local minima, an *excitation factor* has also been included. When a robot's velocity is low and it is not at its goal, excitation will increase the strength of the forces leading toward its goal to encourage it to escape the minima.

The method is relatively simple to implement and provides practical solutions to a wide variety of complex problems. Fig. 1 highlights an example of our approach applied to a group of robots (in black) guarding and escorting an important robot (in green). Additional benchmarks and more detailed results are given in Sec. V.

B. Organization

The remainder of the paper is organized as follows. In Section II we review prior work in the area and briefly describe how our work relates to it. Section III describes the main concepts and the framework that our approach is built on. We describe the coordination algorithm in Section IV. In section V we highlight its performance and analyze the results.

II. Related Work

Motion planning has been extensively studied for more than three decades during which a wide variety of planning frameworks and solutions have been proposed. For the general background and theory of motion planning and coordination, we refer readers to the following texts [7], [8]. The remainder of this section briefly describes prior work related to motion planning or coordination of multiple robots. We provide a more detailed comparison between our work and prior work in Sec. V-C.1.

As briefly mentioned earlier, prior methods for multiple robots are often classified into *centralized* and *decentralized* planners. The centralized approaches aggregate all the individual robots into one large composite system and apply single-robot motion planning algorithms; implicitly coordinating the motion of the individual robots. These include exact motion planners, potential field methods, or sampling-based planners such as Probabilistic Roadmap Methods (PRM) [9], Rapidly-exploring Random Trees (RRT) [10], or their variations. Other specialized algorithms are known for specific cases, such as for pairs or triples of robots in low-density workspaces [11] or flocking behavior [12].

Decentralized planners compromise completeness by using a decoupled approach [13], [14], [15]. Generally, these approaches plan for each robot individually and then perform a velocity tuning step in order to avoid collisions and other coordinate motions along these paths [16], [17]. Alternatively, other schemes such as coordination graphs [18], incremental planning [2], integration with replanning [19], and velocity obstacles [20] can help to ensure that no collisions occur along the paths even in cases with complex dynamics.

Potential field methods [21] and related path modification methods [22] have also been a popular choice for multiple robots [23], [24] due to their relative ease and effectiveness at coordinating motion between multiple robots [25]. In the potential field method, a navigation function is used as a potential field. As the robot follows the gradient of the potential field, it moves toward the goal. Social potential fields alter the navigation function such that the resulting motion reflects social behaviors [5]. Our work is most closely related to this approach.

III. Background and Overview

In this section, we introduce and define several of the concepts of our approach, including physics-based motion planning, the use of social potential fields, and the combination of these two concepts.

A. Physics-Based Motion Planning

Physics-based Motion Planning (PMP), and similarly Constraint-based Motion Planning [3], pose the motion planning problem much like a boundary value problem, where the robot's initial and goal configurations represent the boundaries. The problem is reduced to constrained dynamical simulation and the path is computed iteratively by solving the constraints on the system. The underlying constraints apply *artificial forces* to the robot that guide it toward its goal. Thus, the general motion planning problem is reduced to finding a set of constraints which will guide the robot toward its goal. The advantage of PMP is that motion planning is no longer purely geometric, but also takes into account mechanical and dynamical properties as well as constraints associated with the simulation.

More specifically, for the multi-robot problems, let there be a set of N robots, $\mathcal{R} = \{r_1 \dots r_N\}$. The physical state, $\mathbf{q}_i(t)$, of robot $r_i(t)$ at time t can be described by its degrees of freedom (DOFs) and the first-order derivatives of the DOFs with respect to time, i.e. $\mathbf{q}_i(t) = [\mathbf{x}_i(t)\ \dot{\mathbf{x}}_i(t)]^T$. The robots are in an environment $\mathcal{E}$ with a set of possibly moving obstacles $\mathcal{O}$. A set of m constraints, $\mathcal{C} = \{c_i \dots c_m\}$ are defined such to act on robots in $\mathcal{R}$ and obstacles in $\mathcal{O}$. The constraints include both the coordination constraints and other hard constraints such as non-penetration of other bodies. Given an initial state of the system $\mathcal{Q}(t_{initial}) =$

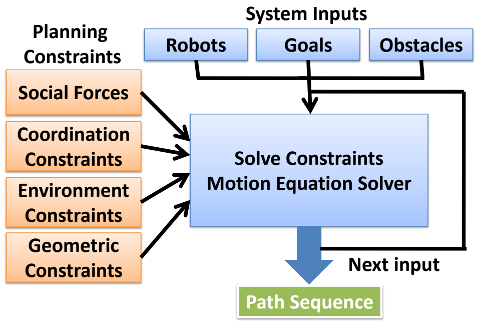

Fig. 2. **Planner Architecture**: The physics-based motion planning cores uses the robots, goals, obstacles, and a set of planning constraints as its input. By solving the constraints and applying the resulting forces, motion is generated and the system is updated. The resulting value is a waypoint along the path and serves as the next input.

$\{\mathbf{q}_i(t_{initial}) \ldots \mathbf{q}_n(t_{initial})\}$, and a final state $\mathcal{Q}(t_{final}) = \{\mathbf{q}_i(t_{final}) \ldots \mathbf{q}_n(t_{final})\}$, the PMP algorithm begins by first initializing the robots to $\mathcal{Q}(t_{initial})$. At each step, PMP updates the state vector $\mathcal{Q}(t)$ by solving the constraints and integrating a robot's motion equation. The Euler-Newton laws of motion are used as our robot's motion equation, though other motion equations could also be used:

$$\mathbf{F}_{net} = \frac{d(m\dot{\mathbf{x}})}{dt} = \sum_{s_i} \mathbf{f}_{\mathbf{s_i}}$$

and

$$\tau_{net} = I\frac{d\omega}{dt} = \sum_{s_i} \mathbf{d}_{\mathbf{s_i}} \times \mathbf{f}_{\mathbf{s_i}}$$

where each force f_{s_i} occurs at a site s_i on the body and $\mathbf{d}_{\mathbf{s_i}}$ is vector from the site s_i to the origin of the inertial frame, I, and ω is the angular velocity. This process continues iteratively until the final state or a local minima is reached.

B. Social Potential Fields

As briefly mentioned earlier, potential field methods [26] are a popular choice for real-time motion planning of multiple robots due to their efficiency and simplicity. Social potential fields (SPF) [5] are a variation of potential fields which characterize the potential function by inverse-force laws, rather than a navigation function. The motion of resulting systems have been shown to exhibit emergent "social" behaviors, such as clustering, guarding, or escorting.

In the initial work, given point robots $r_1, r_2, \ldots, r_n$ with positions $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the force between robots i and j is defined as:

$$\mathbf{F}^{soc}(r_i, r_j) = \left(\sum_{k=1}^{L} \frac{c_{i,j}^{(k)}}{d(i,j)^{\sigma_{i,j}^{(k)}}}\right)\left(\frac{\mathbf{x}_j - \mathbf{x}_i}{d(i,j)}\right) \quad (1)$$

Thus, the social force acting on r_i from r_j is the summation of L force laws, where $c_{i,j}^{(k)}$ is the force coefficient for the kth law, $\sigma_{i,j}^{(k)}$ is the inverse power of the kth law, and $d(i,j) = ||\mathbf{x}_j - \mathbf{x}_i||$. Here, the sign of the force coefficient determines if the force acts in a attractive (positive) or repulsive (negative) manner. These force laws can be easily assigned based on desired behavior to an individual robot, a group of robots, or even an entire set of robots.

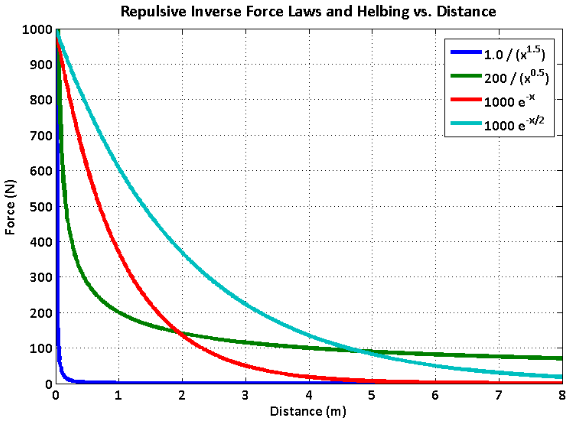

Fig. 3. **Social Potential Fields**: This figure compares Inverse-force laws (dark-blue and green curves) and Helbing's social force laws (red and cyan curves) with differing parameters. The Inverse-force plots initially decline rapidly, and then the rate of decline slows greatly. On the other hand, the Helbing plots do not smoothen out quite as quickly and do not grow indefinitely as distance approaches 0.

Other variations on the Eq. 1 have been proposed, such as that by Helbing et al [6]. The behavior of Helbing's social force law is similar to that of Eq. 1, but also has been shown to exhibit emergent behavior of lane formation for point-based robot primitives. This force is defined as:

$$\mathbf{F}^{soc}(r_i, r_j) = \alpha e^{-d(i,j)/\beta}\mathbf{n}_{i,j} \quad (2)$$

where α is our force coefficient, β is the force fall-of distance, and $\mathbf{n}_{i,j}$ is the normalized vector between $\mathbf{x}_i$ and $\mathbf{x}_j$. Similarly, the sign of α determines whether the force is repulsive or attractive. And, like Eq. 1, several force terms could be used instead of just one for both repulsive and attractive behaviors.

C. Our Approach

As previously mentioned, this work builds upon and combines key ideas of physics-based motion planning and social potential fields. First, PMP is an useful framework for solving motion planning problems with potential fields and multiple robots. This is due to the fact that potential fields can be sampled and transformed into artificial force fields. In PMP, an artificial force field subsequently applies forces directly onto the robot to influence its motion. In this manner, the artificial force field is a type of constraint in itself.

We utilize this property to establish a general local avoidance function based on social potential fields. We generalize the approach to consider both convex and nonconvex polyhedra instead of point-like bodies. Our implementation, we considered the use of both types of social potential functions, Eq. 1 and 2. We achieved similar results with both, and decided to use Eq. 2 since its magnitude is bounded as distance approaches 0 (See Fig. 3). This helps greatly for simulation stability since large forces result in stiff systems. Also, the parameters of Eq. 2 are more intuitive

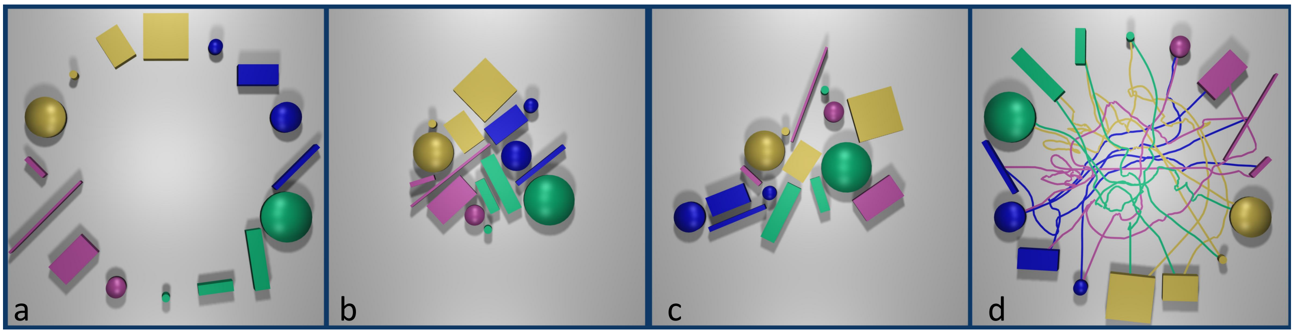

Fig. 4. **Antipodal Robots**: In this benchmark, 16 polygonal robots, colored by starting quadrant, start along the edge of a circle. Each robot must travel to a position across the circle and to a different orientation. This sequence shows four steps in the planning. (a) The robots are at their initial position. (b-c) The robots are in the process of moving across the circle, where social forces help prevent collision while also leading toward the anti-podal position. (d) The agents arrive at their anti-podal positions and we see the paths they traveled.

to tune, since they carry physical meaning. Briefly, α is the maximum magnitude of the social force in Newtons and β is the distance at which the force is approximately $\frac{1}{3}$ of its maximum. Finally, there parameters can also allow for algorithmic improvements which can help with performance.

On top of PMP and SPF, many coordination constraints such as maintaining formations can be easily represented and quickly computed to give the robot the appearance of more advanced behaviors. Finally, the framework is relatively simple to implement given a simulation framework and can usually be easily parallelized for improved performance or distributed control (as shown in Fig. 2).

IV. Local Coordination with Generalized Social Forces

Social forces have been shown to work well for coordination and planning of point-based primitives. However, little work has been done on a similar formulation for a general robot. In this section, we generalize the concept of social forces and integrate it along with other coordination constraints into the physics-based motion planning framework.

A. *Generalized Social Potential Fields*

As previously mentioned, social potential fields are an useful tool for motion and coordination which resemble various complex behaviors. Here, we propose a generalized version of social potential feilds which are applicable to non-point primitives. For a general body, the effect of a social potential field can be felt with varying magnitudes across the entire body. The potential field will not only apply a force, but also induce a torque on the robot.

From Eq. 2 we first define the value of the social potential field from robot r_j at an arbitrary point in space, $\mathbf{x}$ by

$$\mathbf{F}^{soc}(\mathbf{x}, r_j) = \alpha e^{-d(\mathbf{x}, r_j)/\beta}\mathbf{n}(\mathbf{x}, \mathbf{r}_j) \tag{3}$$

where $d(\mathbf{x}, r_j)$ is the shortest distance between $\mathbf{x}$ and r_j, and $\mathbf{n}(\mathbf{x}, \mathbf{r}_j)$ is the normal vector between $\mathbf{x}$ and the nearest point on r_j. A similar social potential can be defined from Eq. 1.

Using Eq. 3, the social force between r_i and r_j, can then be defined as

$$\mathbf{F}^{soc}(r_i, r_j) = \int_{A(r_i)} \mathbf{F}^{soc}(\mathbf{p}, r_j)\, dA \tag{4}$$

where $A(r_i)$ is the region of r_i and $\mathbf{p}$ is a point on the area of r_i in 2D or a point in the volume of r_i in 3D. And, the associated torque on the body is

$$\tau^{soc}(r_i, r_j) = \int_{A(r_i)} (\mathbf{n}(\mathbf{p}, r_j) \times \mathbf{F}^{soc}(\mathbf{p}, r_j))\, dA \tag{5}$$

In effect, the translation component of the total force on r_i will be the resulting social force over the entire region of the robot, and the total torque will similarly be the torque induced by each of these forces at each point on the body.

Like the formulations in Eq. 1 and Eq. 2, the constants of the social force can be adjusted for varying strengths, ranges of effect, and for repulsion or attraction. For a repulsive social force, as the distance between these points decreases the resulting force and torque will increase causing those points on the bodies to separate. Then, the total social force and torque on a single robot would be the summation of the resulting values of Eq. 4 and 5.

1) Discretization: While the general social potential field is formulated over a continuous space, it is not simple to evaluate this integral in implementation. Instead, in our implementation we approximate the integral. Given a uniform sampling of the robot r_i's region, $\mathcal{P}^i = \{\mathbf{p}_1^i, \mathbf{p}_2^i, \ldots, \mathbf{p}_K^i\}$

$$\mathbf{F}^{soc}(r_i, r_j) = \frac{1}{K}\sum_{\mathbf{p}\in\mathcal{P}^i} \mathbf{F}^{soc}(\mathbf{p}, r_j) \tag{6}$$

and

$$\tau^{soc}(r_i, r_j) = \frac{1}{K}\sum_{\mathbf{p}\in\mathcal{P}^i} (\mathbf{n}(\mathbf{p}, r_j) \times \mathbf{F}^{soc}(\mathbf{p}, r_j)) \tag{7}$$

It should be noted that this can lead to a very large number samples, which could become computationally expensive. By using Helbing's formulation of the social force, the force parameters provide several ways of approximating the summations. First, note that the parameter β controls the rate at which the $\mathbf{F}^{soc}$ declines. Thus, when the distance between points and robots is greater than $c\beta$, for relatively large c, the value of the force is negligible and does not contribute greatly to the social force. In our implementation, we set c such that $\alpha e^{-c} < 0.001$. We can use this value as a distance bound and quickly cull away the negligible point-robot pairs. (See Fig. 5).

2) Excitation: One well known drawback of potential field based methods is the possibility of getting stuck in a local minima. This is also the case for the forces generated by our approach. To help overcome a local minima, we define an *excitation* factor, E. Briefly, while an agent has not reached its goal and its velocity is below a predefined threshold, v_0, it's excitation level increases. As excitation rises, the attractive force in the direction of its goal grows as well.

We define excitation to evolve over time as

$$\frac{dE}{dt} = -\frac{E}{T} + (1 - \frac{||\mathbf{v}||}{v_0})$$

where T is a damping factor (we use $T = 2$). Essentially, the first term smooths the excitation so that it does not grow too large too quickly, while the latter term increases or decreases the level of excitation depending on how fast a robot is moving in relation to v_0. Then, given E, we can redefine α in Eq. 2 by

$$\alpha(E) = \alpha * (1.0 + E)$$

In practice, this approach has performed well and we have not observed many situations where a complete deadlock has occurred due to local minima.

B. Additional Coordination Constraints

The social potential forces provide a strong basis for coordination between several robots. A variety of behaviors including grouping, following, guarding, moving toward a local goal, and more can be directly implemented through manipulation of the social potential fields. However, there are a number of situations which are less suitable for use with these fields.

1) Distance Constraints: There are several situations in which group of robots need to be able to form and maintain a formation. While strong attractive social fields could be used to help keep robots in close proximity, they lose strength rapidly when robots get further apart. Therefore, we add a distance constraint which works to maintain a distance between points on a pair of robots. Depending on the strength of the constraint, other robots may or may not be able to easily affect a formation built from these constraints. Briefly, we can define a distance constraint between points $\mathbf{x}_i$ and $\mathbf{x}_j$ on different robots by the function:

$$C_{dist}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_j - \mathbf{x}_i|| - r$$

where r is the desired distance between the points. While this constraint could be solve exactly, we relax the constraint so that the formation can also adapt to necessary changes in the environment or by other robots. Instead, we use a spring-like force:

$$\mathbf{f}_{dist}(\mathbf{x}_i, \mathbf{x}_j) = -\left(k_s(|\mathbf{x}| - r) + k_d \frac{\dot{\mathbf{x}} \cdot \mathbf{x}}{|\mathbf{x}|}\right) \frac{\mathbf{x}}{|\mathbf{x}|}$$

where k_s is a spring constant, k_d is a spring damping constant, and $\mathbf{x} = \mathbf{x}_j - \mathbf{x}_i$. As before, the related torque on robot r_i is:

$$\tau^i_{dist}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{d}_{\mathbf{x}_i,O}) \times \mathbf{f}_{dist}(\mathbf{x}_i, \mathbf{x}_j)$$

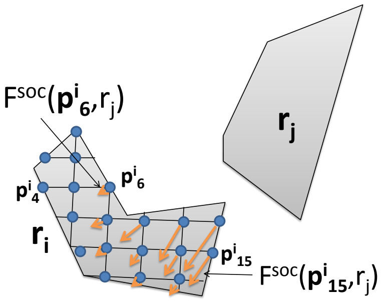

Fig. 5. **Social Force Discretization**: Since the general social force integral is difficult to compute, we discretize the domain. This figure shows the discretized social force computation on r_i from r_j. Robot r_i has been uniformly sampled into a set of points, shown as small blue dots. For each sample, $\mathbf{p}^i_n$ within a fixed distance from r_j, $\mathbf{F}^{\mathbf{soc}}(\mathbf{p}^i_n, r_j)$ will be computed. By the force law definition, points closer to r_j such as $\mathbf{p}^i_{15}$ will receive larger forces than points farther away, such as $\mathbf{p}^i_6$. For points too far away, such as $\mathbf{p}^i_4$, no force is computed.

where $\mathbf{d}_{\mathbf{x}_i,O}$ is the vector from the origin of the inertial frame O to $\mathbf{x}_i$. The net result of this force can be seen in Fig. 1, since the guard robots use a distance constraint to maintain their formation.

V. Results

We have completed a preliminary implementation of this approach and tested its performance and scalability on a workstation computer with 2.66 GHz Intel X5355 Xeon CPUs and 3 GB of main memory. To understand the effectiveness of the approach, we have applied it to a variety of local coordination benchmarks.

A. Benchmarks

We have tested our approach on three different benchmarks, each designed to test different aspects of the coordination technique.

- **Antipodal**: In this benchmark, a fixed number of random robots are placed around the circumference of a circle. Each robot must move across the circle and toward the opposing orientation. This benchmark uses social potential fields between all bodies as well as an attractive force to have the robots cross the circle to its antipodal position. Planning took 125 s to complete the task for 16 robots with an average step time of 8ms. See Fig 4.
- **Letters**: In the letters benchmark, 33 randomly placed robots must navigate to form the letters **ICRA'09**. Like the Antipodal demo, only social repulsive and attractive forces are used. The benchmark took 273 s to complete, with an average step time of 5 ms. See Fig 7.
- **Guarding and Escorting**: In this benchmark, an important robot is trying to navigate to a position across the environment. Several guard robots form a formation around the important robot and then escort it to its goal while trying to remain in formation. This benchmark has 35 aggressive robots, and uses formation constraints as well as social repulsive and attractive potentials. Planning completed in 133 s with an average step time of 42 ms. See Fig 1.

Demo	Num Robots	**Forces (s)**	**Update (s)**	Avg Step Time (s)	Total Time (s)
Antipodal	16	0.0078	0.00001	0.00781	125.7
Antipodal	32	0.0231	0.00003	0.02313	358.7
Antipodal	48	0.0303	0.00004	0.03034	637.2
Antipodal	64	0.0464	0.00005	0.04645	881.1
Antipodal	80	0.0290	0.00007	0.02907	939.3
Antipodal	100	0.0300	0.00008	0.03008	969.4
Antipodal	200	0.0752	0.00017	0.07537	3748.8
Guarding	39	0.0419	0.00003	0.04193	132.9
Letters	33	0.0053	0.00002	0.00532	273.8

TABLE I

Planning times for benchmarks: THIS TABLE SHOWS THE TIMING FOR EACH OF OUR BENCHMARKS. FORCES (FORCE COMPUTATION, CONSTRAINT UPDATES), UPDATE (MOTION EQUATION INTEGRATION), AND STEP TIME ARE AVERAGES OVER THE ENTIRE PLANNING RUN. AND, TOTAL TIME IS THE AMOUNT OF TIME IT TOOK TO FIND A PLANNING SOLUTION.

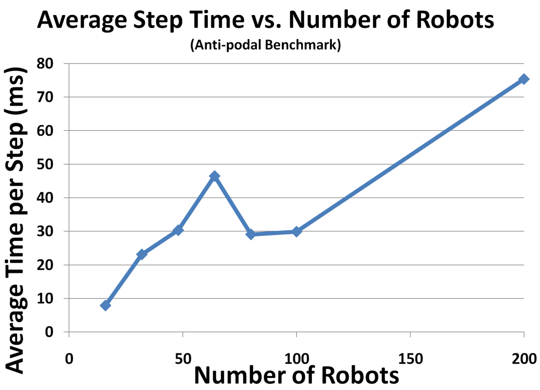

Fig. 6. **Antipodal benchmark timings**: To test scalability, the number of robots was increased in the antipodal benchmark.

The preliminary results of our approach are very promising. In a mostly unoptimized implementation, we were able to generate collision-free, coordinated motion plans in a relatively short amount of time. The included video shows these results for the Antipodal, Letters, and Guarding scenarios. It should be noted that there is no global planner present, so robots will simply attempt to take the straight-line path to goal even if it is less free than other paths. Even though this results in a great deal of traffic, they are frequently able to still reach their goals. Table I highlights the run-time performance for our benchmarks. We note that in all cases, as the number of robots scales from 16 to 200, the average step time remains at a few to tens of milliseconds (ms) which shows that it scales well with the number of robots. For the Antipodal benchmark, there was a drop in the average time as we went beyond 64 robots. This can be attributed to the fact that the size of the circle was increased to accommodate the robots. When the robots are further apart, fewer interactions occur. Fig. 6 confirms the scalability with the number of robots, as it shows the average step time as the number of robots increases. The total planning times can be misleading, since these are also dependent on how far each robot must travel. The average step time is a better indicator of the actual performance.

B. Analysis

There are numerous factors which play a role in the theoretical run-time performance of our approach. The most dominant feature would be the number of robots, N, and their discretization. Each sample on a robot r_i could potentially visit every other sample on every other robot and obstacle at each time step. Thus, in the worst case, our time step complexity approaches $O(M^2)$, for $M > N$ samples. However, by exploiting the fall-off distance parameter in Eq. 2, this is typically not the case. Given the maximum search distance of $c\beta$ (as defined in Sec. IV-A.1), each sample visits a finite, bounded number of other samples within that region. This reduces our run-time complexity to $O(M)$, but it is important to note that while bounded, the maximum number of samples a single sample could visit could still be a large number. In our tests, we did see cases where too fine of a sampling severly degraded performance but these are usually eliminated by the previously mentioned optimizations.

Like other simulations with several parameters, the values of these parameters played a role in the success of the approach. However, since many of the parameters in our approach have real-world interpretations (distance, force in Newtons, etc), we found that giving them realistic values often resulted in successful and stable planning.

C. Comparison and Limitations

In this section, we provide a qualitative comparison of our work with the large body of work on multi-robot motion planning and note some limitations of our approach.

1) Comparisons: Like other decentralized methods, our work relaxes completeness for efficiency. Thus, compared to centralized approaches such as Probabilistic Roadmap Methods (PRM) [9], Rapidly-exploring Random Trees [10], or other variations on these methods, our approach scales much better in the number of robots, but does not offer completeness or convergence. Furthermore, additional coordination constraints can greatly restrict the search space, making it more difficult for centralized approaches to find suitable paths. As a result, these methods become impractical for systems with a large number of robots, such as in the scenarios we are attempting to solve.

Of the decentralized methods, we can group them into several categories; incremental planning, decoupled planning and coordination, dynamic replanning, path modification, and potential fields.

Incremental planning methods first determine a priority for each robot. Then, robots plan in the priority order such that they avoid collisions with robots who have already found a path. Variations on this interleave centralized planning with incremental to improve reliability [2]. Since each robot is completely independent other robots, certain coordination constraints such as distance constraints are more difficult to satisfy than in our approach. In contrast, the social potential model in our approach accommodates close proximity between robots and scales well, as the number of robots increases. However, since incremental planning uses a planner for each individual robot, it supports a wider range of robots than is currently supported by our approach and can navigate in environments where a global path is required.

Decoupled planning and coordination is a variation on incremental planning and decomposes multi-robot planning into two phases. In the first phase, paths are planned without consideration of other robots. Then, a coordination phase adjusts the robot's velocity along their path so that no collisions occur [13], [14], [17]. Similarly, graph representations such as coordination graphs, help to combine planned paths to avoid collisions [18]. This approach shares many of the same qualities as incremental planning. Particularly, stiffer coordination constraints increase the difficulty and lower the chance of success as compared to our approach.

Dynamic replanning methods wait until the current path becomes invalidated, and then uses any remaining valid information to determine a new plan [27], [28], [29], [19]. Some replanning methods have been popular choices for higher-DOF multi-robot problems. The key drawback of replanning methods is that the cost of replanning can be high, particularly for a large group of robots or with additional coordination constraints such as distance constraints. Thus, they will not always scale as well as potential field methods. Path modification approaches [22], [30], [31] attempt to bend or deform the path around moving obstacles, rather than allow them to become invalidated. However, in general these methods cannot handle global changes in connectivity which commonly occur in multiple robot problems.

Various planners also make different assumptions for sensing or exchange of information. Communication is a popular choice, and was successfully used for a replanning approach coupled with decentralized planning of multiple vehicles [19]. Several communication configurations assume that sufficiently precise position and trajectory information can be passed to robots within a local region such that the position of nearby robots is known at any point in time as long as the robot follows that trajectory. This works well and provides essentially the same information as needed by SPF. But, on the other hand, sensing of position information like in SPF can also account for unexpected changes to the paths of other robots or other inaccuracies in sensing.

As discussed earlier, potential field methods [21], [23], [5], [24] rapidly update a navigation function in order to adjust to moving obstacles and robots. Constraint-based Motion Planning (CBMP) [3] also falls into this category with iterative methods to satisfy additional constraints. We propose a generalization of the social potential field, which has assisted in generating collision-free motion in crowded settings while offering real-time performance.

2) Limitations: There are some limitations to our work. There is no guarantee that a robot will find a path even if one exists. However, this has not often been the case in our experiments. While the excitation helps robots escape a local minimum, it is not guaranteed to always work. For instance, a relatively small and light robot will not be able to move past several larger, heavier robots. Since we do not consider points beyond our distance criteria, a fast moving robot or obstacle might collide with a robot before it could react. While dynamic constraints are inherently simulated, they are not explicity handled and thus it is possible for a robot to reach an inevitable collision state. However, it should be noted that by either bounding relative velocities or by adjusting the social force parameters to give a robot sufficient time to react, these cases can frequently be avoided.

VI. Conclusion

We presented a potential field based method for local coordination of multiple robots. For local avoidance and navigation, social potential fields were generalized for convex and non-convex polyhedra and adapted to fit within a physics-based motion planning framework. Overall, the results are promising based on our preliminary implementation. We are able to plan for many robots, several of which are often in tight spaces, in a relatively short amount of time.

There are several additional directions for future work along with the items mentioned in the limitations. It would be useful to integrate this method with a global planner to allow it to function in larger and more complex environments. A similar approach may be possible for holonomic vehicles as well as for articulated robots. For a more realistic simulations, sensing and communication can be considered. Better models of discretization can help to reduce the number of samples per robot while also preserving accuracy. Since the approach is parallelizble, better implementations are likely to improve performance.

VII. Acknowledgements

We would like to thank the reviewers for their comments and help in improving this work and this document.

References

[1] G. Sanchez and J. Latombe, "Using a prm planner to compare centralized and decoupled planning for multi-robot systems," in *ICRA*, 2002, pp. 2112–2119.

[2] M. Saha and P. .Isto, "Multi-robot motion planning by incremental coordination," in *Proc. of IROS*, 2006, pp. 5960–5963.

[3] M. Garber and M. Lin, "Constraint-based motion planning using voronoi diagrams," *Proc. Fifth International Workshop on Algorithmic Foundations of Robotics*, 2002.

[4] R. Gayle, S. Redon, A. Sud, M. C. Lin, and D. Manocha, "Efficient motion planning of highly articulated chains using physics-based sampling," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[5] J. H. Reif and H. Wang, "Social potential fields: A distribued behavioral control for autonomous robots," *Robotics and Autonomous Systems*, 1999.

[6] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics and design solutions: Experiments, simulations and design solutions," *Transportation Science*, vol. 39, no. 1, pp. 1–24, 2005.

[7] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[8] S. M. LaValle, *Planning Algorithms*. Cambridge University Press (also available at http://msl.cs.uiuc.edu/planning/), 2006.

[9] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, pp. 12(4):566–580, 1996.

[10] S. M. LaValle and J. J. Kuffner, “Rapidly-exploring random trees: Progress and prospects,” *Robotics: The Algorithmic Perspective (Proc. of the 4th Int’l Workshop on the Algorithmic Foundations of Robotics*, 2000.

[11] B. Aronov, M. de Berg, A. F. van der stappen, P. Svestka, and J. Vleugels, “Motion planning for multiple roobts,” *Discrete and Computational Geometry*, 1999.

[12] T.-Y. Li and H.-C. Chou, “Motion planning for a crowd of robots,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[13] K. Kant and S. Zucker, “Towards efficient trajectory planning: The path-velocity decomposition,” *Int. Journal of Robotics Research*, vol. 5, no. 3, pp. 72–89, 1986.

[14] T. Simeon, S. Leroy, and J. Laumond, “Path coordination for multiple mobile robots: a geometric algorithm,” *Proc. of IJCAI*, 1999.

[15] P. O’Donnell and T. Lozano-Perez, “Deadlock-free and collision-free coordination of two robot manipulators,” *Proc. IEEE International Conference on Robotics and Automation*, pp. 484–489, 1989.

[16] S. LaValle and S. Hutchinson, “Optimal motion planning for multiple robots having independent goals,” *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, pp. 912–925, 1998.

[17] J. Peng and S. Akella, “Coordinating multiple robots with kinodynamic constraints along specified paths,” *International Journal on Robotics Research*, 2005.

[18] Y. Li, K. Gupta, and S. Payandeh, “Motion planning of multiple agents in virtual environments using coordination graphs,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[19] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman, “Multi-robot search and rescue: A potential field based approach,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[20] J. van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[21] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1985.

[22] S. Quinlan and O. Khatib, “Elastic bands: Connecting path planning and control,” *Proc. of IEEE Conf. on Robotics and Automation*, 1993.

[23] C. W. Warren, “Multiple path coordination using artificial potential fields,” *Proc. of IEEE Conf. on Robotics and Automation*, pp. 500–505, 1990.

[24] S. S. Ge and Y. J. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous Robots*, 2002.

[25] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and M. Norman, “Multi-robot search and rescue: A potential field based approach,” *,Autonomous Robots and Agents Series: Studies in Computational Intelligence*, 2007.

[26] O. Khatib, “Real-time obstable avoidance for manipulators and mobile robots,” *IJRR*, vol. 5, no. 1, pp. 90–98, 1986.

[27] A. Stentz, “The focussed D* algorithm for real-time replanning,” *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.

[28] M. Kallmann and M. Mataric, “Motion planning using dynamic roadmaps,” *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, April 2004.

[29] M. Zucker, J. Kuffner, and M. Branicky, “Multipartite rrts for rapid replanning in dynamic environments,” *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007.

[30] O. Brock and O. Khatib, “Elastic strips: A framework for motion generation in human environments,” *Int. Journal of Robotics Research*, vol. 18, no. 6, pp. 1031–1052, 2002.

[31] Y. Yang and O. Brock, “Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation,” *Proceedings of Robotics: Science and Systems*, August 2006.

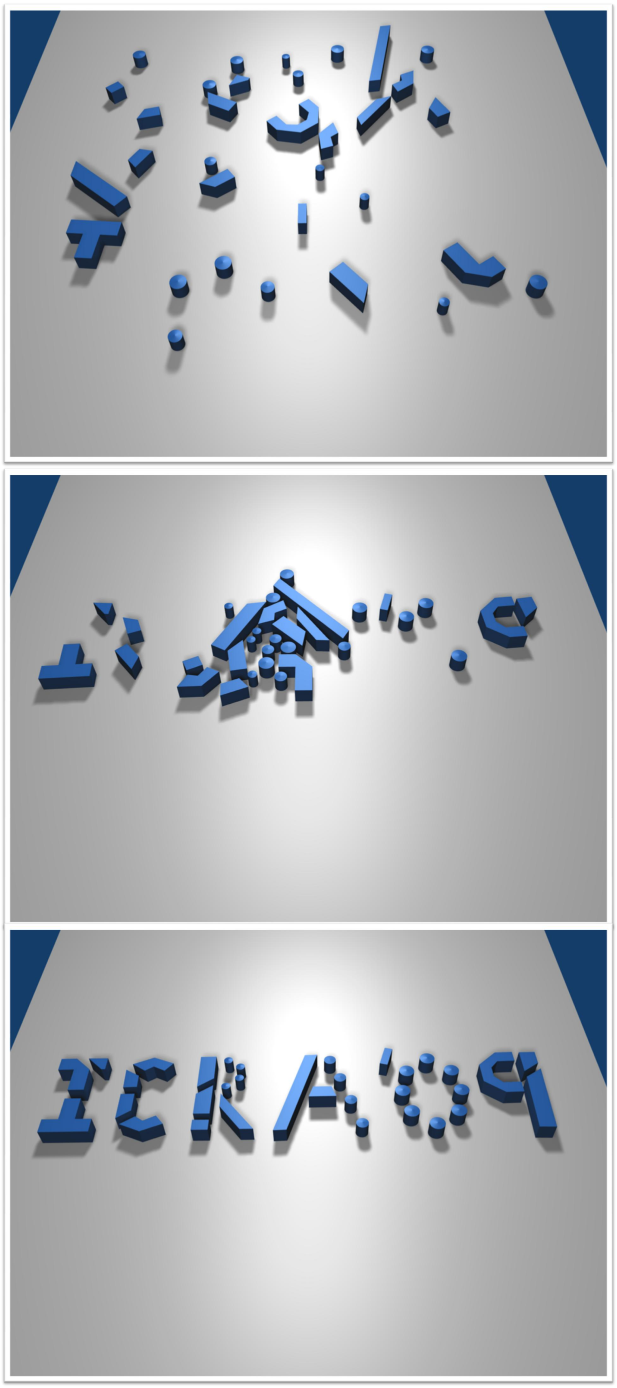

Fig. 7. **Letters**: The letters benchmark consists of 33 convex and nonconvex robots in an assembly-like situation. (a) The letters start off at random positions and orientations. (b) Repulsive and attractive forces allow the robots to move toward their goals but to also avoid collisions with each other. (c) The robots arrive at their final location.