Reduced Order Motion Planning for Nonlinear Symmetric Distributed Robotic Systems

M. Brett McMickell　　Bill Goodwine
mmcmicke@nd.edu　　goodwine@controls.ame.nd.edu
Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

***Abstract*— This paper develops a motion planning algorithm which exploits symmetry in distributed systems to reduce complexity and motion planning design time. The motion planning computations are carried out on a reduced order system, then extended to larger-order equivalent systems in such a way that the objectives of the larger system are satisfied and collision avoidance is guaranteed. The algorithm maintains a rigid body formation as a group robots follows a specified trajectory at the beginning and end of the trajectory. At this point, our algorithm is open loop. A simulation of four robots maintaining a square formation is presented to demonstrate the utility of the algorithm.**

I. INTRODUCTION

This paper considers the motion planning problem for symmetric distributed robotic systems which consist of, perhaps many, robots working together to perform a specified task. As the number of robots increases, so does the overall dimension and complexity of the system. There have been many efforts exploring high level planning and coordination between groups of robots [15], [4], [7], [8], [13]; however, none of these attempted to directly exploit any of the symmetry properties that distributed systems may posses. The aim of this work is to consider discrete symmetries to "reduce" the order of complexity of large-scale distributed systems. In this paper, we consider nonlinear robotic systems with equations of motion of the following, general form

$$\dot{x} = \sum_{i=1}^{m} g_i(x) u_i, \tag{1}$$

where the $g_i(x)$ are smooth analytic vector fields and the u_i's are admissible control inputs.

The problem is to find an algorithm which produces a set of inputs that steers a group of robots from a given initial position and orientation to a final position and orientation while maintaining a rigid body formation at the beginning and end of the trajectory. While rigid body formation control for systems of mobile robots clearly has been achieved before [2], [3], [5], [11], [12], [16], [17], the main contribution of this paper is the development of a motion planning algorithm which exploits symmetry of a distributed robotic system to simplify the design process and reduce the time necessary to develop motion plans for large, complex systems. This paper considers motion planning for symmetric distributed robotic systems in which there is no state information communicated between individual robots. Extending these results to an entire equivalence class of systems, as we have done for controllability in [9], will be the subject of a future paper.

The type of symmetry we consider is when certain robots of the overall system can be interchanged without affecting the dynamics of the overall system. The general idea is that a distributed system is comprised of sets of multiple, repeated instances of identical that can be interchanged. Motion planning is designed considering only one of these robots and then is mapped to the other robots in such a way that the total computational burden is far less than if the motion were planned for each robot separately. Furthermore, collision avoidance is guaranteed.

The remainder of this paper is organized as follows. A description of symmetric distributed systems is given in Section II. This is followed by the development of an equivalence relation between vector fields, which leads to the definitions of symmetric systems. A brief summary of piecewise constant motion planning is given in Section III. This leads to the motion planning algorithm for symmetric distributed systems. Section IV presents simulation results demonstrating the utility of the motion planning algorithm on a system of four mobile robots maintaining a square formation.

II. DRIFTLESS SYMMETRIC DISTRIBUTED SYSTEMS

This section outlines background material from the authors' previous work [9], [10] necessary to formulate our motion planning algorithm for symmetric nonlinear distributed systems. In this section, we provide the definitions and notation related to our representation of a driftless distributed system, define a symmetric nonlinear distributed system, and define an equivalence relation between different symmetric nonlinear distributed systems. The equivalence relation naturally leads to an equivalence class of control systems.

A. Driftless nonlinear distributed systems

We will consider smooth analytic driftless systems of the form

$$\begin{aligned} \Sigma: \qquad \dot{x} &= g_{1,1}(x)u_{1,1} + g_{1,2}(x)u_{1,2} + \cdots \\ &+ g_{2,1}(x)u_{2,1} + g_{2,2}(x)u_{2,2} + \cdots \\ &\vdots \\ &+ g_{n,1}(x)u_{n,1} + g_{n,2}(x)u_{n,2} + \cdots, \end{aligned} \tag{2}$$

for all $x \in M$ where M is a smooth manifold, $g_{i,j}$ are smooth analytic vector fields on M.

Since we are considering distributed systems, the system is assumed to be organized into individual robots, corresponding to which are certain vector fields and control inputs. In Equation (2), the first subscript on the g's and u's identifies the robot to which the vector field and control input corresponds, and the second subscript indexes different vector fields and inputs within that subsystem. Let $\tilde{g}_i(x)$, then it represents the *ordered set* of vector fields associated with the ith robot, *i.e.*, $\tilde{g}_i(x) = \{g_{i,1}, g_{i,2}, \ldots\}$.

We assume that M is partitioned into a set of m regular submanifolds, M_i, such that M is the Cartesian product of the M_i, *i.e.*, $M = \prod_{i=1}^{m} M_i$. Each submanifold M_i represents a *subsystem* or *robot* of the distributed system. In this paper, each M_i would represent the configuration space for robot i in the system and $\{u_{i,1}, u_{i,2}, \ldots\}$ would be the control inputs for that robot.

B. Symmetric nonlinear distributed systems

Now we will consider what it means for a nonlinear distributed system to be symmetric. Recall from the introduction that the motivating idea is that there is a subset of individual robots that can be interchanged without changing the dynamics of the overall team of robots. Mathematically this will be represented by the fact that vector fields from various robots will, in some sense, be equivalent. Since the vector fields from different robots are defined on different spaces, we need a definition of equivalence which is more than just requiring that they be 'identical'.

DEFINITION II.1 *Two vector fields, g_1 And g_2 are equivalent, denoted $g_1 \sim g_2$, if there exists a diffeomorphism, $\psi : M \mapsto M$, such that*

$$\psi_* g_1 = g_2,$$

where ψ_* is the push forward of ψ,

$$\psi_* g(x) = T\psi \circ g \circ \psi^{-1}(x),$$

and T is the usual tangent operation (see [1]).

The definition of vector field equivalence applies to general submanifolds without any assumptions regarding the relationship between the coordinate systems defined on different robots; however, often each robot is parameterized identically so that the diffeomorphism, ψ, in Definition II.1 is a simple permutation of states.

Since typically equivalence is determined by a permutation of coordinates, we first review the symmetric group and it's action on a set. Recall that the symmetric group of order $p!$, denoted S_p, is the group of permutations of p objects. A permutation of a set $X = \{1, \ldots, p\}$ is a one-to-one mapping of X onto itself. Such a permutation ρ is written,

$$\rho = \begin{pmatrix} 1 & 2 & \cdots & p \\ k_1 & k_2 & \cdots & k_p \end{pmatrix},$$

which represents that 1 is mapped to k_1, 2 is mapped to k_2, *etc*. The following example further illustrates vector field equivalence.

EXAMPLE II.2 Consider a system of five robots where each robot is parameterized by one state and let g_2 and g_3 be given by,

$$g_2(x) = \begin{bmatrix} x_1 \\ \cos x_2 \\ x_2 + 1 \\ 0 \\ x_2 x_5 \end{bmatrix}, \quad g_3(x) = \begin{bmatrix} x_1 \\ x_3 x_2 \\ \cos x_3 \\ x_3 + 1 \\ 0 \end{bmatrix}$$

and,

$$\psi : M \mapsto M$$

defined by

$$\psi(x_1, x_2, x_3, x_4, x_5) = (x_1, x_5, x_2, x_3, x_4),$$

which corresponds to interchanging robot 2 with node 3. Also, ψ is related to $\rho \in S_4$ where

$$\rho = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 4 & 5 & 2 \end{pmatrix},$$

where the x_1 coordinate is fixed and coordinates $x_2, \ldots, x_5$ are mapped according to ρ. The inverse mapping is

$$\psi^{-1}(x_1, x_2, x_3, x_4, x_5) = (x_1, x_3, x_4, x_5, x_2).$$

Invariance of a system with respect to interchanging robot 2 and 3 requires that

$$\psi_* g_2 = g_3 \qquad \text{and} \qquad \psi_*^{-1} g_3 = g_2.$$

In detail,

$$\begin{aligned}
&\psi_* g_2(x_1,x_2,x_3,x_4,x_5)\\
&= T\psi \circ g_2 \circ \psi^{-1}(x_1,x_2,x_3,x_4,x_5)\\
&= T\psi \circ g_2(x_1,x_3,x_4,x_5,x_2)\\
&= T\psi \circ \begin{bmatrix} x_1 \\ \cos x_3 \\ x_3+1 \\ 0 \\ x_3x_2 \end{bmatrix}\\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \cos x_3 \\ x_3+1 \\ 0 \\ x_3x_2 \end{bmatrix}\\
&= \begin{bmatrix} x_1 \\ x_3x_2 \\ \cos x_3 \\ x_3+1 \\ 0 \end{bmatrix}\\
&= g_3(x).
\end{aligned}$$

A similarly straight-forward computation shows that $\psi_*^{-1} g_3(x) = g_2(x)$.

Given an equivalence relation among vector fields, we now define a symmetric nonlinear distributed system.

DEFINITION II.3 *Let a symmetry orbit,* $\mathbf{O}$*, be subset of* p *robots in* Σ*, let* $\mathbf{F}$ *be the subset of* Σ *containing* $n-p$ *fixed robots, and let* $\rho \in S_p$*. The system* Σ *is a symmetric nonlinear distributed system if*

$$\tilde{g}_i \sim \tilde{g}_{\rho(j)} \qquad \forall i \in \{1,\dots,p\} \text{ and } \forall \rho \in S_p,$$

where $\tilde{g}_i$ *is the ordered set of driftless vector fields corresponding to robot* i *in* $\mathbf{O}$*.*

III. NONLINEAR MOTION PLANNING FOR SYMMETRIC SYSTEMS

The motion planning algorithm developed in this paper is an extension of piecewise constant motion planning algorithm from [6]. A complete description of this motion planning algorithm is beyond the scope of this paper, so only an outline will be provide in this section. This section also provides a method for ensuring that there are no collisions between robots.

A. Piecewise motion planning

Piecewise constant motion planning works exactly for systems whose controllability Lie algebra is nilpotent, *i.e.,* there exists a $k>0$, such that $[g_{i_1},\dots,[g_{i_{p-1}},g_i],\dots] = 0$, $\forall p>k$ and for all vector fields g_i. For systems that are not nilpotent, the method provides only an approximate solution and explicit error bounds on the resulting error are given in [6].

The basic idea of piecewise continuous motion planning is to decompose the desired trajectory into multiple subtrajectories along vector fields which, when evaluated at a point, form a basis for the tangent space of the configuration space. For underactuated systems, the basis will contain motion in the directions of a Lie brackets. Recall, a Lie bracket in coordinates is given by

$$[g_1,g_2] = \frac{\partial g_2}{\partial x_i} g_1(x) - \frac{\partial g_1}{\partial x_i} g_2(x).$$

Motion in a Lie bracket direction can be approximated using the following four segment flow,

$$\phi^t_{[g_1,g_2]}(x_0) = \phi^{\sqrt{t}}_{-g_2} \circ \phi^{\sqrt{t}}_{-g_1} \circ \phi^{\sqrt{t}}_{g_2} \circ \phi^{\sqrt{t}}_{g_1}(x_0) \qquad (3)$$

where $\phi^t_g(x_0)$ represents the flow along the vector field g for time t starting at point x_0. Using basis vector fields, any smooth trajectory can be represented by the *Chen-Fleiss series*,

$$S_t(q) = e^{h_s(t)B_s} e^{h_{s-1}(t)B_{s-1}} \cdots e^{h_2(t)B_2} e^{h_1(t)B_1}, \quad (4)$$

where $[h_1,\dots,h_s]$ are functions know as *Phillip Hall coordinates*, $B_1,\dots,B_s$ are *Phillip Hall basis elements*, and S is the series representation of a given trajectory. Phillip hall basis elements are related to the original system and their Lie brackets and are chosen such that the the basis elements are full rank over the desired trajectory.

The Chen-Fliess series satisfies the following differential equation, referred to as the *formal differential extended system*,

$$\dot{S}(t) = S(t)(B_1 v_1 + \cdots + B_s v_s), \qquad (5)$$

where $v_1,\dots,v_s$ are fictitious inputs. The inputs are referred to as fictitious because they are inputs associated with basis vector fields and may not be available to the actual system.

By differentiating Equation 4 and equating it to Equation 5, we can solve for the $\dot{h}$'s in terms of the fictitious inputs, which results in a ordinary differential equation,

$$\dot{h} = Q(h)v, \quad h(0) = 0, \qquad (6)$$

where Q(h) is a coefficient matrix in terms of the Phillip Hall coordinates. The solution of this differential equation represents the evolution of the system in response to the fictitious inputs.

B. Motion planning for symmetric distributed systems

Motion planning for a nonlinear symmetric system of p robots in $\mathbb{R}^{m\times p}$ using rigid body formation is accomplished as follows. Let each robot consist of n states, where $n \geq m$. First, determine a rotation matrix $R(wt) \in SO(m)$ such that $R(wT)$ produces the desired final orientation of the rigid formation, where w is the rotational velocity of the rigid formation. Next, choose a trajectory, $q(t) \in C^1$, connecting the initial center of the formation to a desired final center for a given $t \in [0,T]$. This can be done using any C^1 function. Note that $q(t) \in \mathbb{R}^m$. Therefore, the trajectory of a robot i in the rigid body is given by,

$$p_i(t) = R(wt)P_i + q(t), \quad (7)$$

where $P \in \mathbb{R}^n$ is the initial starting position of robot i relative to the center of the formation. The rigid formation uniquely determines the position of each robot, but it does not constrain the robot's orientation. Let $r(t) \in C^1$ describe the changing orientation of the robots. The rigid body trajectory for robot i can be written as,

$$\gamma_i(t) = A(wt)\hat{P}_i + \hat{q}(t) + \hat{r}(t),$$

where $\hat{P}$ is the robot i's initial state (position and orientation) with respect to the center of the rigid body, $A(wt)$ is an augmented rotation given by,

$$A(wt) = \begin{bmatrix} \text{R(wt)} & 0 \\ 0 & 1 \end{bmatrix},$$

$\hat{q}(t)$ is an augmented trajectory given by,

$$\hat{q}(t) = \begin{bmatrix} \text{q(t)} \\ 0 \end{bmatrix},$$

and $\hat{r}(t)$ is given by,

$$\hat{r}(t) = \begin{bmatrix} 0 \\ \text{r(t)} \end{bmatrix}.$$

Taking the derivative of the trajectory, we find

$$\dot{\gamma}_i(t) = \dot{A}(wt)\hat{P}_i + \dot{\hat{q}}(t) + \dot{\hat{r}}(t).$$

Select s linearly independent Phillip Hall basis elements, $\{B_1, \ldots, B_s\}$, and determine the corresponding fictitious inputs. To do this, define an ordered matrix $\tilde{C}$ composed of all the linearly independent vector fields,

$$\tilde{C}(\gamma_i(t)) = [g_1(\gamma_i(t)), g_2(\gamma_i(t)), \ldots, g_s(\gamma_i(t))].$$

Recall that some of the g_i's will be Lie brackets between vector fields in the original system. The basis elements were chosen so they have full rank over the entire trajectory. Therefore, $\tilde{C}$ is invertible. The fictitious inputs for robot i are given by,

$$v_i = \tilde{C}^{-1}(\gamma_i(t))\dot{\gamma}_i(t).$$

where $v = [v_1, \ldots, v_n]^T$. The Phillip Hall coordinates corresponding to the fictitious inputs are determined by solving Equation 6 and the initial condition h(0)=0. This gives the control inputs for the extended system. Inputs in the extended system that are associated with motion in a Lie bracket direction are approximated using Equation 3.

The motion plan for developed for this robot can now be extended to other equivalent robots using the following Proposition.

PROPOSITION III.1 *Let Σ be a robotic system containing p robots, such that, $\Sigma : \dot{x} = g_1(x)u_1 + \cdots + g_n(x)u_n$. If Σ is symmetric distributed system, then the Phillip hall coordinates of any two robots, i and j, in the symmetry orbit of the system with a desired trajectory given by Equation 7, are related by,*

$$\dot{h}_i = Q(h)v_i = Q(h)(\psi_* \tilde{C}_j(\psi(\gamma_j(t))))^{-1}\psi_* \dot{\gamma}_j(t). \quad (8)$$

Proof: Consider the trajectory given by Equation 7 for two robots, i and j, which are in the symmetry orbit. Let $\Sigma_i : \dot{x}_i = g_1(x_i)u_1 + \cdots + g_m(x_i)u_n$ and $\Sigma_j : \dot{x}_j = f_1(x_j)u_1 + \cdots + f_m(x_j)u_n$ trajectories of the two system are related by the diffeomorphism, ψ, such that

$$\gamma_i(t) = \psi(\gamma_j(t)) \implies \dot{\gamma}_i(t) = \psi_* \dot{\gamma}_j(t).$$

Let $\tilde{C}_i$ be the ordered matrix of vector fields corresponding with system Σ_i. The ordered matrices of vector fields, $\tilde{C}$, are also related by the diffeomorphism,

$$\begin{aligned} \tilde{C}_i(\gamma_i(t)) &= \psi_* \tilde{C}_j(\psi(\gamma_j(t))) \\ &= [\psi_* f_{k_1}(\psi(\gamma_j(t))), \ldots, \psi_* f_{k_m}(\psi(\gamma_j(t)))], \end{aligned}$$

which implies that $\tilde{C}_i^{-1}(\gamma_i(t)) = (\psi_* \tilde{C}_j)^{-1}(\psi(\gamma_j(t)))$. Therefore, the fictitious inputs, v_i, are also related through the diffeomorphism,

$$\begin{aligned} v_i &= \tilde{C}_i(\gamma_i(t))^{-1}\dot{\gamma}_i(t) \\ &= (\psi_* \tilde{C}_j(\psi(\gamma_j(t)))^{-1}\psi_* \dot{\gamma}_j(t). \end{aligned}$$

Diffeomorphisms are natural with respect to Lie brackets, *i.e*,

$$[\psi_* f, \psi_* g] = \psi_*[f, g],$$

which implies that if system Σ_i is nilpotent of order k, then Σ_j is also nilpotent of order k. Therefore, $Q_i(h) = Q_j(h) = Q(h)$. Therefore,

$$\dot{h}_i = Q(h)v_i = Q(h)(\psi_* \tilde{C}_j(\psi(\gamma_j(t)))^{-1}\psi_* \dot{\gamma}_j(t).$$

■

C. Collision avoidance

The overall approach is to decompose the complete trajectory into subtrajectories that are small enough to ensure there is no collision in the system. Since we are considering small motions, we will consider the system locally in $\mathbb{R}^n$. For the trajectory, γ, given in Equation 7 for $t \in [0, T]$, let $\mathcal{R}_i = \min_{t \in [0,T]} \|\gamma_i(t) - \gamma_j(t)\|$, such that $i \neq j$, *i.e.* the closest any robot gets to robot i while following the trajectory. Also, let $\Delta_i = \|\gamma_i(T) - \gamma_i(0)\|$. Consider a linear trajectory $\Gamma_i(t) = \gamma_i(0) + t(\gamma_i(T) - \gamma_i(0))$ connection the initial position to the final position. Recall, the fictitious inputs are calculated by solving $\dot{\gamma}_i(t) = [g_1(\gamma_i(t)), \ldots, g_s(\gamma_i(t))]v_i$. Applying this to the linear trajectory, $\Gamma_i(t)$, we find

$$\|\dot{\Gamma}_i\| = \|\gamma_i(T) - \gamma_i(0)\| < \|[g_1(\gamma_i(t)), \ldots, g_s(\gamma_i(t))]\|\|v_i\|.$$

From this equation, we find that the fictitious inputs are bounded by a constant, α_i, *i.e.*, $\|v_i\| < \alpha_i\|\dot{\Gamma}_i\| = \alpha_i\Delta_i$.

By construction of the real inputs from the fictitious inputs, $\|u\| < \alpha_i \Delta^{1/k}$ where k is the order of the highest Lie bracket needed to make $\tilde{C}$ full rank. Let $x_{i,\max} = \max_{t\in[0,T]} \|x_i(t) - \gamma_i(0)\|$ denote the flow that is maximally distant from the starting point. Note, this is not necessarily $\gamma_i(T)$. Now, pick a ball, $\mathcal{B}_i$ of radius $\mathcal{R}_i$ centered at the initial point. Let η_i be the maximum norm of all the first order vector fields for all points in the ball $\mathcal{B}_i$. The distance, $\|x_{i,\max} - \gamma_i(0)\|$ is necessarily bounded by the sum of the norms of each individual flow associated with one real control input, $u^l_{i,j}$. That is,

$$\|x_{i,\max} - \gamma_i(0)\| \le \sum_l \sum_j \| \int_0^1 g_l u^l_{i,j} dt\|.$$

We know, $\|u^l_i\| \le \alpha_i \Delta_i^{1/k}$ and $\|g_l(x)\| \le \eta_i$ for all x_i. Therefore,

$$\|x_{i,\max} - \gamma_i(0)\| \le \sum_l \sum_j \eta_i \alpha_i \Delta^{1/k},$$

and since $\Delta_i = \|\gamma_i(T) - \gamma_i(0)\|$, by choosing the desired final point close enough to the stating point, the robots will not collide. Because Δ_i is raised to the power of $1/k$, if k is large, then Δ_i may need to be exceedingly small. This approach is very conservative and the appropriate step length may best be identified experimentally.

IV. EXAMPLE

Consider a group of four simple robotic unicycles each described by [14],

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2 \qquad (9)$$

where u_1 is the linear velocity input and u_2 is the angular velocity input. All robots are identically parameterized, so the diffeomorphism, ψ, is simply a translation mapping.

The robots are initially in a square formation centered about the origin a distance of unity apart. The robots are to follow a linear path, $q(t) = [t, t, 0]^T$ for a time $t \in [0, 1]$ with the orientation of the square rotating by an angle π. The initial and final points are illustrated in Figure 1 (a).

Motion planning is done on one robot and then extended to the other robots. Equation 9 describing the mobile robots is not nilpotent; however, it is nilpotentizable (see [6]). Using inputs,

$$\begin{aligned} u_1 &= \frac{1}{\cos(\theta)} w_1 \qquad (10) \\ u_2 &= \cos^2(\theta) w_2, \end{aligned}$$

the system becomes

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 1 \\ \tan\theta \\ 0 \end{bmatrix} w_1 + \begin{bmatrix} 0 \\ 0 \\ \cos^2\theta \end{bmatrix} w_2,$$

which is nilpotent of order 2. The motion plan for the transformed systems using piecewise constant inputs will be exact. Solving equation 6 for the desired motion, we find that the Phillip Hall coordinates for the ith robot are,

$$\begin{bmatrix} h^k_{1i} \\ h^k_{2i} \\ h^k_{3i} \end{bmatrix} = \begin{bmatrix} \cos(\pi t)P^0_{1i} + \sin(\pi t)P^0_{2i} + t - P^k_{1i} \\ 0 \\ -\sin(\pi t)P^0_{1i} + \cos(\pi t)P^0_{2i} + t - P^k_{2i} \end{bmatrix},$$

where $P^0_1 i$ and $P^0_2 i$ are the initial x and y positions of the ith robot, respectively. The desired motion is greater than the maximum collision bound, so the motion must be divided into segments. Each segment, i, of robot i has different h values denoted h^k_i. Since the state of the robots changes with each step, we denote the starting state at the kth segment of the motion as P^k_i. For example, the initial state of robot 1 is $P^0_1 = [-1, 1, 0]^T$, so the corresponding Phillip Hall coordinates for the initial step are,

$$\begin{bmatrix} h^0_{11} \\ h^0_{21} \\ h^0_{31} \end{bmatrix} = \begin{bmatrix} -\cos(\pi t) + \sin(\pi t) + t + 1 \\ 0 \\ \sin(\pi t) + \cos(\pi t) + t - 1 \end{bmatrix}.$$

After determining the motion plan for a robot j, the motion plan is extended to the other robots using Proposition III.1. The resulting Phillip hall coordinates for the ith robot are,

$$\begin{bmatrix} h^k_{1i} \\ h^k_{2i} \\ h^k_{3i} \end{bmatrix} = \begin{bmatrix} \cos(\pi t)\psi(P^0_{1j}) + \sin(\pi t)\psi(P^0_{2j}) + t - \psi(P^k_{1j}) \\ 0 \\ -\sin(\pi t)\psi(P^0_{1j}) + \cos(\pi t)\psi(P^0_{2j}) + t - \psi(P^k_{2j}) \end{bmatrix}.$$

Figure 1 displays a simulation of the four robots. Figure 1 (a) displays the robots initial and final positions shown as 'o' and 'x', respectively. Figure 1 (b)-(e) show the motion during each of the four subtrajectories necessary to move to final position without a collision. The complete motion plan for the mobile robotic system is given in Figure 1 (f). Note, the maximum step size determined by the collision bound given in Section III-C is conservative. The step size for this example was computed experimentally.

V. CONCLUSIONS AND FUTURE WORK

A motion planning algorithm for nonlinear symmetric systems that exploits the symmetry of a system has been developed. The algorithm is based on piecewise constant inputs [6], which is exact for nilpotent systems. A bound on the maximum step size is provide that ensures the motion is collision free. A simulation of a group of mobile robots was used to demonstrate the utility of this algorithm.

The motion plan developed in this paper is for noninteracting symmetric robots. Future work is directed toward removing this restriction and developing a motion planning algorithm for symmetric distributed systems that can be designed on a “reduced order” equivalent system. We are also exploring a more general formation control which would consider time-varying formations. Furthermore, we are experimentally testing motion control algorithms on a group of small distributed robots.

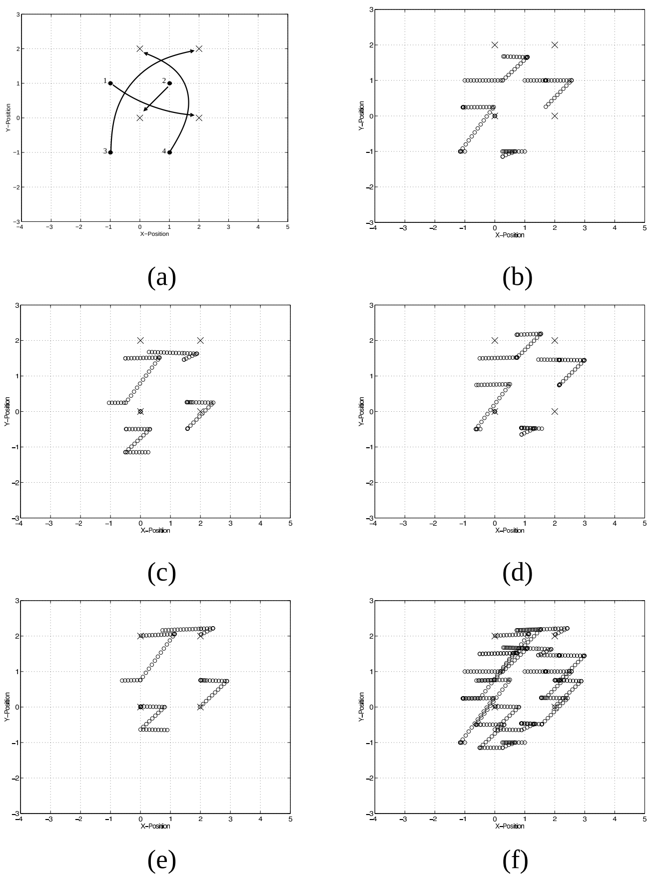

Fig. 1. Plot (a) displays the initial and final configurations shown as “o” and “x”, respectively. Plots (b) - (e) displays the motion plan in four steps to avoid collisions. Plot (f) displays the combined motion plan of all four steps.

VI. ACKNOWLEGEMENT

The partial support of the National Science Foundation grant NSF CCR01-13131 is gratefully acknowledged.

VII. REFERENCES

[1] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. Springer–Verlag, second edition, 1988.

[2] M. Egerstedt and X. Hu. Formation constrained multi-agent control. In *IEEE International Conference on Robotics and Automation*, pages 3961–3966. IEEE, 2001.

[3] J. Alexander Fax and Richard M. Murrray. Information flow and cooperative control of vehicle formations. In *IFAC World Congress*, 2002. to appear.

[4] S. Hirose, R. Damoto, and R. Kawakami. Study of super-mecho-colony (concept and basic experimental setup. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1664–1669, 2000.

[5] Eric Klavins. Automatic synthesis of controllers for distributed assembly and formation forming. In *IEEE International Conference on Robotics and Automation*, pages 3296–3302, May 2002.

[6] G. Lafferriere and Hector J. Sussmann. A differential geometric approach to motion planning. In X. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 235–270. Kluwer, 1993.

[7] Xu Liying, S. Zien-Sabatto, and A. Sekmen. Development of intelligent behaviors for mobile robot. In *Proceedings of the 33rd Southeastern Symposium on System Theory*, pages 383–386, 2001.

[8] R. Logan and S. Theodoropoulos. The distributed simulation of multiagent systems. In *Proceedings of IEEE*, volume 89, pages 174–185, 2001.

[9] M. Brett McMickell and Bill Goodwine. Reduction and nonlinear controllability of symmetric distributed systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1232–1237, 2001.

[10] M. Brett McMickell and Bill Goodwine. Reduction and nonlinear controllability of symmetric distributed systems. *International Journal of Control*, 2002. Submitted for review.

[11] Mark B. Milam, Nicolas Petit, and Richard M. Murray. Constrained trajectory generation for micro-satellite formation flying. In *AIAA Guidance, Navigation, and Control Conference*, 2001.

[12] Reza Olfati-Saber and Richard M. Murrray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, 2002. to appear.

[13] M. Quinn. A comparison of approaches to the evolution of homogenous multi-robot teams. In *Proceedings of the 2001 Conference on Evolutionary Comutations*, volume 1, 2001.

[14] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*, chapter 11. Springer, 1999.

[15] T. Sugar, J.P. Desai, V. Kumar, and J.P. Ostrowski. Coordination of multiple mobile manipulators. In *IEEE International Conference on Robotics and Automation*, pages 3022–3027, 2001.

[16] Ichiro Suzuki and Masafumi Yamashita. Distributed autonymous mobile robts: Formation of geometric patterns. *SIAM J. Comput.*, 28(4):1347–1363, 1999.

[17] Hiroaki Yamaguchi and Joel Burdick. Time-varying feedback control for nonholonomic mobile robots forming group formations. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 4156–4163, 1998.