Feedback Control Methods for Distributed Manipulation Systems that involve Mechanical Contacts

T.D. Murphey and J.W. Burdick
Division of Engineering and Applied Science
California Institute of Technology, Pasadena, CA

Abstract—**This paper introduces feedback control methods for distributed manipulation systems that move objects via rolling and slipping point contacts. Due to the intermittent nature of these mechanical contacts, the governing mechanics of these systems are inherently nonsmooth. We first present a methodology to model these nonsmooth mechanical effects in a manner that is tractable for nonsmooth control analysis. Using these models, we show that when considerations of these nonsmooth effects are taken into account, a class of traditional open loop distributed manipulation control methods cannot stabilize objects near an equilibrium. However, stability can be achieved through the use of feedback, and we present nonsmooth feedback laws with guaranteed stability properties. We then describe an experimental modular distributed manipulation test-bed upon which one can implement a variety of control schemes. Experiments with this test-bed confirm the validity of our control algorithms. Multimedia extensions include videos of these experiments.**

I. Introduction

Distributed manipulators usually consist of an array of similar or identical actuators combined together with a control strategy to create net movement of an object or objects. The goal of many distributed manipulation systems is to allow precise positioning of planar objects from all possible starting configurations. Such "smart conveyors" can be used for separating and precisely positioning parts for the purpose of assembly. Distributed manipulator actuation methods range from air jets [4], [29], [30] and rotating wheels [20], [18], [23] at the macroscale to MEMS and flexible cilia [32], [7] at the microscale.

This paper focuses on the design of distributed manipulation control algorithms for the class of distributed manipulators that create object motion through the use of rolling and slipping mechanical contacts. In such cases, friction forces and the intermittent nature of frictional contact play an important role in the overall system dynamics, leading to non-smooth dynamical system behavior. A key point of this paper is that the nonsmooth properties of these distributed manipulation systems are non-negligible, but are nevertheless tractable and amenable to non-smooth analysis and non-smooth control methods.

In order to properly analyze these non-smooth effects on system behavior, we first introduce in Section II a Power Dissipation Method (PDM) to construct tractable equations of motion for distributed manipulators with intermittent frictional contact. This method formalizes and extends the work of Alexander and Maddocks [1]. The resulting equations take the form of a non-smooth multiple model dynamical system (Definition II.2). We next show in Section III that when these non-smooth mechanical effects are taken into account, the *programmable force field* (PFF) approach to distribution manipulation control is in fact unstable in the object's orientation near the vicinity of equilibrium points. We then show that non-smooth control techniques can be used to stabilize the moving object near an equilibrium (Section IV). Moreover, the programmable force field approach can be blended with our feedback algorithm to provide a globally stable feedback control scheme that efficiently uses sensing and feedback where it is most effective (Section VI). In presenting our analysis, we differentiate between *underactuated* and *fully actuated* distributed manipulators. A separate smooth feedback control algorithm is presented in Section V when full actuation is available. To test our theories, we constructed a modular experimental distributed manipulator test-bed. This system, which is described in Section VII, can emulate a reasonably large class of distributed manipulators that generate motion through rolling and sliding frictional contacts. Experimental results that confirm the validity and effectiveness of our algorithms are presented in Section VIII. The multimedia extensions to this paper present a video of each experiment described in Section VIII.

A. Relation to Prior Work

Methods to design distributed manipulation control systems have been proposed in several other works, including [8], [11], [13], [20]. Particularly relevant to this paper is the work of Luntz et al. [20], where a "modular distributed manipulator system" (MDMS) is introduced. The MDMS consists of a planar array of independently driven wheels that are located at fixed orientations. These wheels manipulate objects placed on top of the array through the influence of mechanical contact between the moving object and the wheel rims. The authors of [20] design and analyze control laws for the purpose of point stabilization of the transported object. Although the work in [20] does consider the fact that the total number of actuators is finite, it compensates for this fact largely through inherent compliance in the system (via backdrivable motors and other mechanisms). They then show to what degree the programmable vector field approach (described shortly) can be applied to a discrete array of actuators. In the cases where the programmable vector field does not provide good enough results they then use feedback to compensate for the discreteness effects. The approach we give here more directly addresses the issue of breaking kinematic contact constraints due to the system being kinematically "overconstrained". This allows us to directly incorporate the effects of sliding frictional contact into our control strategy. Moreover, the experiment we describe later in Section VII has the advantage of having independently driven, independently oriented wheels. As we will see in Section V this allows us to have control laws which are globally smooth.

Aside from the work found in [20], the most common approach to control of a distributed manipulator is based on the notion of *programmable force fields* (PFF) Böhringer et al. [5], [8]. In this methodology, one makes the possibly unrealistic assumption that the array's control capability can be idealized as a continuous distribution of forces across the array surface. In this abstraction, the manipulated object moves under the influence of these surface forces. The distributed forces can be integrated over the body's surface to obtain the object's dynamical response. This process is as follows: assume the part $\mathcal{O}$ can be described by a support characteristic function $\omega(p)$ where $\omega(p)$ is 1 everywhere on the object surface and 0 otherwise. Here p denotes planar Cartesian coordinates in the object's fixed frame. Moreover, let the part be subject to a force field $f(x,y):\mathbb{R}^2 \to \mathbb{R}^2$. Lastly, place the reference frame of $\mathcal{O}$ at the object's center of mass, i.e., $\int_{\mathbb{R}^2}\omega(p)dp = 0$. When the object lies at configuration $q=(x,y,\theta)$ the net force and torque on the object are

$$\begin{aligned} F &= \int_{\mathbb{R}^2} \omega(p) f(A_\theta p + t) dp \\ M &= \int_{\mathbb{R}^2} \omega(p) A_\theta p \times f(A_\theta p + t) dp \end{aligned}$$

with $t=(x,y)^T$ and A the 2x2 rotation matrix of angle θ. The condition for equilibrium is $F=M=0$.

The control design problem reduces to the selection of a continuous force field distribution that will transport the object to a prescribed configuration, and then stabilize it at that configuration. The most basic control law is the "elliptic" field, which generally takes the form $F=\{-\alpha x, -\beta y\}$, where α and β are coefficients to be chosen by the control designer. Under the continuous force idealization, these open loop control laws can stabilize an object to one of several stable equilibria depending on the shape and distribution of mass of the object. Recent work has introduced more general fields that can stabilize more general classes of objects. See, for instance, Böhringer et al. [6], [8], [9] and Kavraki and Sudsang [31].

To use these controls on an actual array, where the manipulation forces will be realistically generated at discrete points, one must adapt the continuous approximation to the given discrete geometry. To assure the performance of the algorithm on the actual system, the contact mechanics should be taken into account at this stage, though this has not often been done in prior work.

The PFF approach has been experimentally shown to work in MEMS-fabricated actuator arrays, where the array elements are "small" and "close" together relative to the size of the object being manipulated [5], [3]. This approach is additionally well suited to distributed air jets, where the aerodynamics effectively "smooth out" the resulting forces on the object. In these cases the strong assumptions underlying the PFF approach are physically realistic. Moreover, it has the advantage of being open loop, and therefore requires no sensing. However, in cases where only a small number of actuators are in contact with the manipulated object (i.e., the continuous actuation approximation is poor) or the coefficient of friction μ is very high, the continuous approximation has been shown experimentally not to work as well [19]. In these cases, the continuous approximation does not adequately incorporate the physics of the actual array and the object/array interface.

These experimental observations led us to explicitly incorporate frictional and discontinuous contact effects into the analysis and control of distributed manipulation (and the related case of overconstrained wheeled vehicles [22] and overconstrained mechanical systems in general [28]). We show that under very simple and general assumptions on the friction model, the PFF approach generically leads to unstable systems when implemented on actual distributed manipulation arrays that have frictional contact (see Section III-B). The instability arises in the object's orientation at the equilibrium configuration. We present a nonsmooth control in Section IV law that locally stabilizes this instability in a provably correct way. We also show that one can combine the programmable force field approach and the previously mentioned local control law to obtain a globally exponentially controllable distributed manipulation control system (Section VI). These results are all proved using the formalism of nonsmooth analysis.

In this paper we present and unify our previous work on non-smooth dynamical models for describing such systems [22], and non-smooth control laws [23], [26] that provably stabilize these systems. The present paper is based on the conference paper [25], but includes stronger stability results than the ones found there as well as additional experimentation.

II. The Power Dissipation Method

Many actual or proposed distributed manipulator implementations rely upon physical contact between the manipulated object and the driving elements. Examples include driving wheels, fingers, cilia, or flaps. To explicitly investigate, incorporate, and control the complex frictional contact phenomena inherent in such systems, one needs to develop general modeling schemes that can capture these phenomena so that the subsequent control design process can take these effects into account. One could resort to a general Lagrangian modeling approach that accounts for the contact effects through Lagrange multipliers. Instead, we present a modeling scheme that captures the salient physical features of intermittent frictional contact in a class of equations that are amenable to non-smooth control analysis.

To realize this goal, we use a "power dissipation method" (PDM) to model the governing dynamics of a distributed manipulation system involving a finite number of frictional contacts. This method is adapted from the work of Alexander and Maddocks [1] on kinematic wheeled vehicles. Our contribution has been to formalize the method and better understand its properties. One can show that this method generically produces unique models [22] that are relatively easy to compute, and to which one can apply non-smooth control system analysis methods. Since the method is a quasi-static modeling method, it produces first-order governing equations, instead of second order equations that are associated with Lagrange's equations. However, in other work (see [28]) we have shown that these equations of motion are nonsmooth *kinematic reductions* of the complete Lagrangian dynamical models. That is, there is a formal relationship between the equations generated by the PDM analysis and those generated by Lagrangian analysis.

We assume that the moving body and actuator elements that contact the object can be modeled as rigid bodies making point contacts that are governed by the Coulomb friction law at each contact point[1]. Let x denote the configuration of the array/object system, consisting of the object's planar location, and the variables that describe the state of each actuator element. Under these assumptions, the relative motion of each point contact between the object and an actuator array element can be written in the form $\omega(x)\dot{x}$. If $\omega(x)\dot{x} = 0$, the contact is not slipping (and the constraint is satisfied), while if $\omega(x)\dot{x} \neq 0$, then $\omega(x)\dot{x}$ describes the slipping velocity at the contact.

In general, the moving object will be in contact with the actuator array at many points. From kinematic considerations, one or more of the contact points must be in a slipping state, thereby dissipating energy. The *power dissipation function* measures the object's total energy dissipation due to contact slippage.

Definition II.1. The *Dissipation Functional* for an n-contact state is defined to be

$$\mathcal{D} = \sum_{i=1}^{n} \alpha_i \mid \omega(x)\dot{x} \mid \tag{1}$$

where $\alpha_i = \mu_i N_i$, with μ_i and N_i being the Coulomb friction coefficient and normal force at the i^{th} contact, which are assumed known[2]. $\Diamond$

Since there will generally not exist a motion where all of the contacts can be simultaneously slipless, we are lead to the following concept for finding the governing motions.

> **Power Dissipation Principle:** With $\dot{x}$ sufficiently small so that quasi-static analysis is appropriate, an object's motion at any given instant is the one that minimizes $\mathcal{D}$.

Assuming that the motion of the actuator array's variables are known, the *power dissipation method* postulates that the object's motion at each instant is the one that instantaneously minimizes power dissipation due to contact slippage. This method is adapted from the work of Alexander and Maddocks [1] on wheeled vehicles. For a greater discussion of the formal characteristics of the PDM, and a discussion of the relationship between the PDM and Lagrangian approaches for such a system, see [21], [28], [22].

When one applies the PDM method to an overconstrained multi-contact distributed manipulation system, the resulting equations take the following form [23].

Definition II.2. A system is a *multiple model driftless affine system (MMDA)* if it can be expressed in the form

$$\dot{x} = f_{\sigma_1}(x)u_1 + f_{\sigma_2}(x)u_2 + \cdots + f_{\sigma_n}(x)u_n \tag{2}$$

where for any x and t, $f_{\sigma_i}(x) \in \{g_{\alpha_i}(x) | \alpha_i \in I_i\}$, with I_i an index set, f_i measurable in (x,t), and the vector fields g_i are analytic in (x,t) for all i. $\Diamond$

[1] Line contacts can be modeled as a set of equivalent point contacts.

[2] If the exact distribution of normal forces is not known, typically an additional step of analysis can yield the distribution or a reasonable approximation of the normal force distribution. However, this approximation is often hindered by nonuniquness and indeterminism.

An MMDA is a driftless affine nonlinear control system where each control vector field may "switch" back and forth between different elements of a finite set. In our case, this switching corresponds to the switching between different contact states between the object and the array surface elements (i.e., different sets of slipping contacts) due to variations in contact geometry, surface friction properties, and normal loading. In [22] it was shown that the PDM generically leads to MMDA systems as in Definition II.2. For analysis of control theoretic issues in MMDA systems, see [24], [27], [21]

A. Equations of Motion

This section applies the power dissipation method to an example of an array of driven wheels that manipulate a planar object. This systems models our experimental apparatus (Fig. 6 in Section VIII). Assign a fixed reference frame in the plane of the moving object. The i^{th} wheel rim contacts the moving object at the point (x_i, y_i) in the fixed frame. The angle between the tangent to the wheel rim at the point of contact and the x-axis of the fixed frame is denoted by θ_i.

Definition II.3. When the tangent to the motion of the contact point has a constant orientation, this contact is said to be *underactuated.* When the orientation of the tangent can be controlled at will, we say that the contact is *fully actuated.* In the first case, each contact represents a single control input, while the fully actuated case represents two independent control inputs.

Assume that the i^{th} wheel has radius R and spins at speed $\frac{1}{R}u_i$. Then the linear velocity of the point of contact along the tangent to the wheel is u_i. Let $SO(2)$ denote the group of all planar rotations and $SE(2)$ denote the group of all planar displacements. Let q be the configuration of the object in $SE(2)$. Let $g_i \in SE(2)$ is the homogeneous representation of the i^{th} actuator node's configuration relative to a fixed reference frame.

$$g_i = \begin{bmatrix} R(\theta_i) & \begin{pmatrix} x_i \\ y_i \end{pmatrix} \\ 0 & 1 \end{bmatrix} \in SE(2) \qquad (3)$$

The relative velocity of each contact point between the wheel and moving object can be expressed as $\Omega(g_i)(\dot{q}, u_i)$ where:

$$\Omega_i(g_i) = \begin{bmatrix} \left[Ad^T_{g_i^{-1}} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right]^T & -1 \\ \left[Ad^T_{g_i^{-1}} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right]^T & 0 \end{bmatrix} \qquad (4)$$

$$= \begin{bmatrix} \cos\theta_i & \sin\theta_i & x_i \sin\theta_i - y_i \cos\theta_i & -1 \\ -\sin\theta_i & \cos\theta_i & x_i \cos\theta_i + y_i \sin\theta_i & 0 \end{bmatrix} (5)$$

where $Ad(\cdot)$ is the adjoint transformation which transforms velocities from one coordinate frame to another, and $R(\cdot) \in SO(2)$. (Note that the terms in the first column are three dimensional row vectors.) I.e., this equation represents the constraint associated with the i^{th} actuator when the contact does not slip.

We allow for the realistic possibility that friction at the contact point is not uniform in all directions of the contact plane. An example of such a smooth distribution of the coefficient of friction is shown in Fig. 1 (see [15], [16] for a discussion of such friction models).

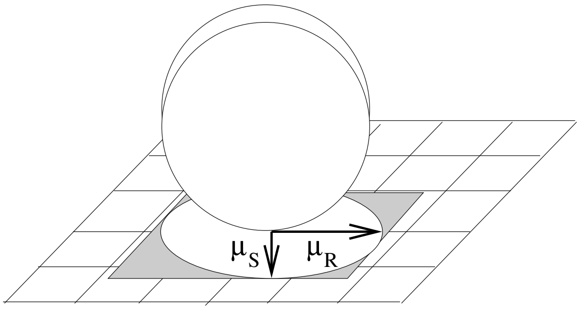

Fig. 1. A wheel with the vector-dependent friction with two friction induced metrics.

While some materials do have friction of this type, such anisotropic friction models are more generally useful as a means to approximately model compliance effects and wheel tread effects in our rigid body modeling framework. For instance, if the driving wheel is an extremely thin disk, then one would expect μ_R (the friction coefficient along the "rim" direction) to be less than μ_S (the friction coefficient along the "side" direction). However, the treads on a tank ensure that μ_R is greater than μ_S. Note that the minimum of the dissipation function will be nonunique in the case of a isotropic (i.e., $\mu_S = \mu_R$) distribution of friction. Note also that in this case, the same indeterminacy arises in the Lagrangian mechanics analysis. The form of μ is important, as it will induce a metric on the velocities of the system. Hence, an elliptical distribution (as seen in white in Fig. 1) will induce a weighted H_2 norm, whereas an absolute value distribution (as seen in gray in Fig. 1) will induce a weighted H_1 norm. We will choose the latter primarily because it is a reasonably good approximation of systems with hysteresis effects due to friction and is easy to work with. However, we should note that that the μ distribution affects the value of $\mathcal{D}$ in different coordinate systems, so any effort to make $\mathcal{D}$ coordinate independent must explicitly take the distribution of μ into account.

We now apply the PDM to this problem. For simplicity, let us introduce for each actuating wheel a reference frame whose origin is fixed at the point of contact, and whose x-axis, denoted x^i_R, points along the tangent to the wheel rim. The y-axis is chosen by the right hand rule, and is denoted by y^i_S. For a set of n actuators, the dissipation functional takes the form:

$$\mathcal{D}(\dot{q}) = \sum_{i=1}^{n} |\mu^i_S N^i \omega^i_S(q)\dot{q}| + |\mu^i_R N^i \omega^i_R(q)\dot{q}| \tag{6}$$

where $\omega^i_S(q)$ is the component of the i^{th} actuator's frictional constraint along the side-slip direction y^i_S, and $\omega^i_R(q)$ denoted the component of this constraint on the rolling direction, x^i_R. The governing equations can then be found by minimizing $\mathcal{D}$ in Eq. (6). To establish necessary conditions for the minimum of $\mathcal{D}$, we would like to take the differential of $\mathcal{D}$. But differentiation is impossible in a classical sense because $\mathcal{D}$ is only Lipschitz continuous. Therefore, we take the nonsmooth differential (in the sense of Clarke [10]). To simplify the calculation of this non-smooth derivative, note that $\mathcal{D}$ takes the form

$$\mathcal{D}(\dot{q}) = \sum_{i=1}^{2n} g(F_i(q,\dot{q}))$$

where $g(y) = |y|$ and $F_i(q,\dot{q}) = \mu^i_X N^i_X \omega^i_X(q)\dot{q}$ for $X = S$ or R. Note that g is Lipschitz and the functions $\{F_i\}$ are differentiable and linear in $\dot{q}$. Therefore (dropping the X subscript) $DF_i(q,\dot{q}) = \mu^i N^i \omega^i(q)$, as expected. Moreover, the Clarke Generalized Differential (CGD) of $g(y) = |y|$ is

$$\frac{\partial g}{\partial \dot{y}} = \begin{cases} 1 & \text{if } y > 0 \\ -1 & \text{if } y < 0 \\ \lambda : |\lambda| \le 1 & \text{if } y = 0. \end{cases} \tag{7}$$

The CGD obeys the chain rule [10], implying that the CGD of $\mathcal{D}$ is:

$$\frac{\partial \mathcal{D}_i}{\partial \dot{q}} = \begin{cases} \mu^i N^i \omega^i(q) & \text{if } \mu^i N^i \omega^i(q)\dot{q} > 0 \\ -\mu^i N^i \omega^i(q) & \text{if } \mu^i N^i \omega^i(q)\dot{q} < 0 \\ \lambda \mu^i N^i \omega^i(q) : |\lambda| \le 1 & \text{if } \mu^i N^i \omega^i(q)\dot{q} = 0 \end{cases} . \tag{8}$$

We focused on the Coulomb friction model in this derivation. If instead we replace $\mu_i N_i$ with a generic configuration-dependent function $h_i(q)$ we can model other types of frictional contact, and obtain an analogous result. We should also comment that in the case of Coulomb friction the minimization of $\mathcal{D}$ is equivalent to finite search over a discrete set of velocities (see [28], [22]).

If one makes the additional assumption that the coefficient of friction is the same at all contacts, $\mu(x,y) = \mu$, then the distribution of the normal forces determines the nature of the dissipation, since the coefficient of friction factors linearly from the dissipation function in this case. This implies that the contact states that dissipate the most energy are those associated with the potential constraints having the largest normal forces N_i. Based on these observations, if the location of the center of mass determines the distribution of normal forces, and if $\mu(x,y)$ is uniform, then the contacts closest to the center of mass do not slip. The non-slip constraints of these contacts define the object's motion. Hence, knowledge of the center of mass's location determines the first two actively satisfied constraints. The third actively satisfied constraint is determined by the friction model.

Let $a_i = (x_i, y_i)$, the location of the i^{th} actuator and let $q_{x,y} = (x,y)$ of the (x,y,θ) configuration of the object. If the coefficient of friction for sideways slip, μ_S, is less than the coefficient of friction in the rolling direction, μ_R, and if the nearest actuator to the center of mass is indexed by i and the second nearest is indexed by j, the equations of motion must satisfy the following conditions.

$$\dot{q} \in \left\{ v \;\middle|\; \begin{array}{l} \|a^i - q_{x,y}\| \le \|a^k - q_{x,y}\| \;\; \forall k \ne i \\ \|a^j - q_{x,y}\| \le \|a^k - q_{x,y}\| \;\; \forall k \ne i,j \\ (\omega^i_R(q), \omega^i_S(q), \omega^j_R(q))v = 0 \end{array} \right\} \tag{9}$$

These conditions can be interpreted as follows. The first condition implies that the i^{th} actuator is closest to the center of mass of the object. The second condition implies that the j^{th} actuator is the second closest actuator to the center of mass. Then, we know that the velocity must satisfy the three constraints (i.e., $(\omega^i_R(q), \omega^i_S(q), \omega^j_R(q))v = 0$ where $(\omega^i_R(q), \omega^i_S(q), \omega^j_R(q))$ is a matrix consisting of the three constraint one forms). One can show by solving for the annihilator of this matrix that these conditions in turn imply that the equations governing the object's motions are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{u_i[s_j((x_i - x_j)c_i + y_i s_i) + c_i c_j y_j] - u_j y_i}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{u_j x_i - u_i[c_i c_j x_i + s_i(x_j s_j + (y_i - y_j)c_j)]}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{u_j - u_i \cos(\theta_i - \theta_j)}{(x_i - x_j)s_j + (y_j - y_i)c_j} \end{bmatrix} \tag{10}$$

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, u_j is the speed of the i^{th} contact, and it should be recalled that (x_i, y_i) is the location of the i^{th} point contact and θ_i is its orientation in the plane. It should be noted that here the index notation should be thought of as mapping (i,j) pairs to equations of motion in some neighborhood (not necessarily small) around the i^{th} and j^{th} actuator. The transition between the equations of motion determined by actuators i and j to equations of motion determined by actuators k and l will in general be determined by the location of center of mass. The state space is consequently divided into different sets of equations of motion whose boundary geometries are dependent upon the center of mass location (an example of this division can be seen in Fig. 3). This model can be expressed as the MMDA system

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = f_1u_1 + f_2u_2 \tag{11}$$

where

$$f_1 \in \begin{bmatrix}\frac{-y_i}{(x_j-x_i)s_j+(y_i-y_j)c_j}\\ \frac{x_i}{(x_j-x_i)s_j+(y_i-y_j)c_j}\\ \frac{1}{(x_i-x_j)s_j+(y_j-y_i)c_j}\end{bmatrix} \tag{12}$$

$$f_2 \in \begin{bmatrix}\frac{s_j((x_i-x_j)c_i+y_is_i)+c_ic_jy_j}{(x_j-x_i)s_j+(y_i-y_j)c_j}\\ \frac{-c_ic_jx_i-s_i(x_js_j-(y_i-y_j)c_j)}{(x_j-x_i)s_j+(y_i-y_j)c_j}\\ \frac{-\cos(\theta_i-\theta_j)}{(x_i-x_j)s_j+(y_j-y_i)c_j}\end{bmatrix} \tag{13}$$

Note that in Eq. (12) the indices i and j still refer to the closest and second closest actuators. Hence, f_1 and f_2 will experience discontinuous changes as the values of i and j change. Moreover, as the system trajectory $q(t)$ crosses a boundary between one region where the equations of motion are determined by actuators i and j to a region where the equations of motion are determined by actuators k and l, f_1 and f_2 must be multivalued on the boundary (hence the use of the inclusion $\in$ symbol). In cases where a boundary is attractive (all trajectories near it converge to it in finite time), the differential inclusion formalism is necessary in order to obtain existence of solutions. This is because on such an attractive boundary solutions cannot escape, and are moreover no longer unique.

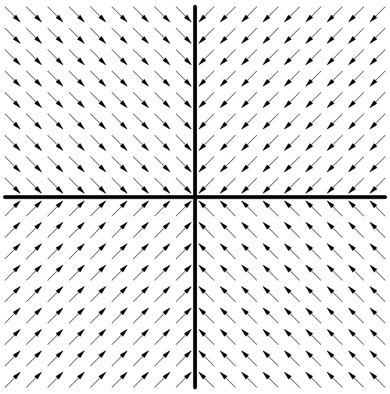

Fig. 2. A discontinuous vector field for which the x and y axes are attractive.

To clarify some of these ideas, consider Fig. 2. Here we see a system that has four possible dynamic states (one for each quadrant). Each quadrant has a constant vector field (as Eq. (10) does when the controls are held constant). Note, however, that every trajectory converges to the origin despite the piecewise constant nature of the dynamics. This example additionally illustrates the need for set-valued analysis–both the x-axis and y-axis are attractive, leading to nonunique trajectories for most initial conditions. This is due to the fact that most trajectories hit one of the axes before they get to the origin, and at that point they join all the other trajectories on that axis as they go to the origin.

Remark II.1. The programmable force field (PFF) method effectively assumes that there are an infinite number of actuators, that all of the actuators slip all the time, and that the physics of contact between the array surface and the object is not important. Hence, the PFF method is more appropriate to the analysis and design of gross motions where accuracy is less important and simplicity of the design problem is appealing. The PDM assumes that there are generally a finite number of contact points, and incorporates a Coulomb friction (or possibly other) contact model into the analysis. However, the PDM can only be well justified for quasi-static systems where the objects move slowly enough that the contact reaction forces dominate the moments of inertia. For distributed manipulation, this will generally be true when the object is near an equilibrium point. Therefore, the PDM is more appropriate to the analysis and control of local, quasi-static motions near the equilibrium. These contrasting features inspire the merging of these two techniques in Section VI.

III. Instability of the Programmable Force Field Method

This section analyzes the stability of the programmable force field technique under the inclusion of the contact models described above. To analyze the stability of these non-smooth multi-model systems, we first introduce a non-smooth extension to the standard Lyapunov stability method.

A. Stability Theorems for Differential Inclusions

The system in Eq. (11) is a differential inclusion of the type found in [12]. That is, due to the switching of the control vector fields in Eq. (2), the governing equations take the form $\dot{x} \in F(t,x)$, where $F(t,x)$ is a (set-valued) multi-function. Our analysis requires the following theorem from [12] (partially extended in [26]), which is the generalization of time varying Lyapunov theory to differential inclusions. We remind the reader that the upper and lower derivatives for a function $V(t,x) \in C^1$ are defined by:

$$\dot{V}^* = \sup_{y\in F(t,x)} (V_t + \nabla V y) \qquad \dot{V}_* = \inf_{y\in F(t,x)} (V_t + \nabla V y) \tag{14}$$

Theorem III.1 ([12]). *Let, in a closed domain* $D(t_0 \leq t < \infty, |x| \leq \epsilon_0)$, *the differential inclusion* $\dot{x} \in F(t,x)$ *satisfy the basic conditions of existence and* $0 \in$

$F(t,0)$. *In this domain, let there exist functions* $V(t,x) \in C^1, V_0(x) \in C$ *for which*

$$V(t,0) = 0, \quad V(t,x) \geq V_0(x) > 0, \quad (0 < |x| < \epsilon_0)$$

Then:
1) If $\dot{V}^* \leq 0$ *in D, the solution* $x(t) = 0$ *of the inclusion* $\dot{x} \in F(t,x)$ *is stable.*
2) If, moreover, there exist functions $V_0(x), V_1(x) \in C$, $W(x) \in C$ *(for* $|x| \leq \epsilon_0$*) and*

$$0 < V_0(x) \leq V(t,x) \leq V_1(x), \quad \dot{V}^* \leq -W(x) < 0,$$

$$(0 < |x| < \epsilon_0), \quad V_1(0) = 0$$

then the solution $x(t) = 0$ *is asymptotically stable.*
3) Moreover, if there exist $k_1, k_2, k_3, c > 0$ *such that*

$$\begin{array}{l} V_0(x) \geq k_1\|x\|^c \\ V_1(x) \leq k_2\|x\|^c \\ W(x) \geq k_3\|x\|^c \end{array}$$

then the solution $x(t) = 0$ *is exponentially stable.*

A simple consequence of this theorem is the following corollary.

Corollary III.2. *If a single Lyapunov function can be used to show that all models of a multiple model system are (stable, asymptotically stable, exponentially stable), then the resulting multiple model system is also (stable, asymptotically stable, exponentially stable).*

B. Instability of 2-Dimensional Arrays Controlled by the PFF

This section shows that distributed manipulators of the type described in the previous section are not asymptotically stabilizable when controlled using the open loop programmable force field method, but are asymptotically stabilizable using feedback. We only give the proof for arrays manipulating objects in $\mathbb{R}^2$, but this proof can be extended to any smooth two dimensional manifold by identifying its charts with $\mathbb{R}^2$.

Theorem III.3. *Consider a 2-dimensional planar array consisting of a finite number of actuators placed on a regular grid. Let the contacts be indexed by* $i, j \in I$*, and let their physics be modeled as in Section II. Assume that the motions of the actuator contacts are governed by an elliptic vector velocity field* $\Psi(x,y) : \mathbb{R}^2 \to \mathbb{R}^4$ *(where* $\mathbb{R}^4$ *is the tangent bundle of* $\mathbb{R}^2$*) defined by* $\Psi(x,y) = (x, y, -\alpha x, -\beta y)$ *for* $\alpha, \beta > 0$*. Then, in some neighborhood of the origin (the desired equilibrium point) the first order governing equations given by the PDM are not asymptotically stable under the influence of the programmable force field (PFF). Moreover, if* $\mu_S < \mu_R$*, then such a system is exponentially stabilizable through the use of feedback.*

Proof. To see that the PFF-controlled system is not asymptotically stable, it is sufficient to realize that $Ad_{g^{-1}}$ in Eq. (4) does not depend on θ whenever the controls are constant. Consequently, the θ dynamics has no actual dependence on θ (i.e., the equations governing the objects orientation take the form $\dot{\theta} = C(\cdot)$, where the function $C(\cdot)$ is independent of θ). Additionally, this implies that the system equations are invariant with respect to initial condition in θ. Thus, object motion in the θ direction is not asymptotically stable. (This is actually true of any static field ψ for the same reason. Hence, no open loop programmable force field can stabilize the orientation under the assumptions of the PDM.)

To see that the system is stabilizable, assume that the governing equations are determined at time t almost always by two constraints at the actuator i with coordinates (x_i, y_i) and input u_i, and one constraint at the actuator j with coordinates (x_j, y_j) and input u_j. Moreover assume that $\mu_S < \mu_R$, thereby ensuring a "rolling" constraint is satisfied rather than a "side-slipping" constraint. Choose a Lyapunov function whose vanishing point is the desired equilibrium: $V(x,y,\theta) = \frac{1}{2}\|q\|^2 = \frac{1}{2}(x^2 + y^2 + \theta^2)$. In every region that is not a boundary between models, one can solve the equation

$$\dot{V} = \frac{\partial V}{\partial q}\dot{q} = -k\|q\|^2 \tag{15}$$

for u_i and u_j. We do so by plugging $\dot{q} = [\dot{x}, \dot{y}, \dot{\theta}]^T$ from Eq. (11) into Eq. (15) and solving for u_i and u_j. Then

$$u_i = \frac{u_i^{num}}{u_i^{den}} \tag{16}$$

where

$$\begin{aligned} u_i^{num} &= u_j\ (\theta - y\,x_i + x\,y_i) + \\ &\quad k\ \left(\theta^2 + x^2 + y^2\right)\ (s(\theta_j)\ (x_i - x_j) + c(\theta_j)\ (-y_i + y_j)) \end{aligned}$$

and

$$\begin{aligned} u_i^{den} &= s(\theta_j)\ (x\,c(\theta_i)\ (x_i - x_j) + s(\theta_i)\ (\theta - y\,x_j + x\,y_i)) \\ &\quad + c(\theta_j)\ (y\,s(\theta_i)\ (-y_i + y_j) + c(\theta_i)\ (\theta - y\,x_i + x\,y_j)) \end{aligned}$$

where u_j can be chosen arbitrarily in the expression for u_i^{num}. Therefore, every model in the multiple model system satisfies the equality in Eq. (15), thus meeting the criterion of Corollary III.2. Therefore, because each model is individually exponentially stable, the origin is exponentially stable for the multiple model system as well. Moreover, the bounding exponent is just the minimum of the individual models. Note that we never mentioned the condition $\mu_S < \mu_R$. This is because the condition is implicit in the equations of motion from Eq. (11) which allow us to solve for u_i in Eq. (16). □

This is a stronger result than the one previously presented by the authors in [25] in that it no longer has any weakly stabilizable modes (modes where only a subset of the trajectories are stabilized), as is validated in the experimental results in Section VIII. Something to note about this result is that the stabilizing control switches discretely as the contact state switches. This means that the control system must know the current contact state. In our current approach we estimate the contact state using the underlying physics (and corresponding assumptions about uniformity of friction and normal forces). However, outside of a carefully built system such assumptions may not be valid, implying the need for online contact state estimation. Fortunately, supervisory control provides a natural framework for both the estimation and control of such a problem. See [2], [17] for an overview of supervisory control and [21] for the application of supervisory control to this problem. We should also comment that the $\mu_S < \mu_R$ assumption only has an obvious physical interpretation for wheel-like contacts, and that the case $\mu_S > \mu_R$ may be stabilizable, but not in as straight forward a fashion.

Remark III.1. The model (11) has the following characteristic. Theorem III.3 implies it is always possible to stabilize the object's center to the desired equilibrium point (which is assumed here to be the origin) while also stabilizing its orientation. However, it is not necessarily always possible to stabilize the object to points within an arbitrarily small neighborhood of the origin. This will depend on the specific configuration of the actuators. In the next section we will present an example where it is only possible to stabilize to points on the manifolds $x = \pm y$. However, these limitations are acceptable, as we seek to stabilize the object to the origin. Moreover, when using the array to transport the object over large distances, these stabilizable manifolds would form a connected graph $G \subset \mathbb{R}^2$. It is unclear if a change in geometry (such as a non-uniform contact distribution) would allow one to stabilize to more points in a neighborhood of the origin.

IV. An Example of Nonsmooth Feedback Algorithms for Local Stabilization at the Origin

The last section showed that when contact mechanics are taken into account, the PFF method cannot stabilize the orientation of a body that is guided by an array with a finite number of frictional contacts. However, Theorem III.3 showed that such systems are stabilizable, and hinted at a general Lyapunov approach for designing stabilizing controllers. This section shows, for the specific array geometry of Fig. 3, how to apply the nonsmooth feedback found in the proof of Theorem III.3 to the stabilization of an object. Experimental validation of this algorithm can be found in Section VIII. While this algorithm is specific to this array geometry, the basic approach can be similarly applied to many other situations. The approach is based on designing control laws for each model in the governing multiple model system. Then, a supervisory controller switches between control law depending upon the current system state. The control laws and switching scheme are chosen to guarantee stability by using the methods described in Sections III and VI. This methodology allows the control design to be relatively simple, even for complex systems.

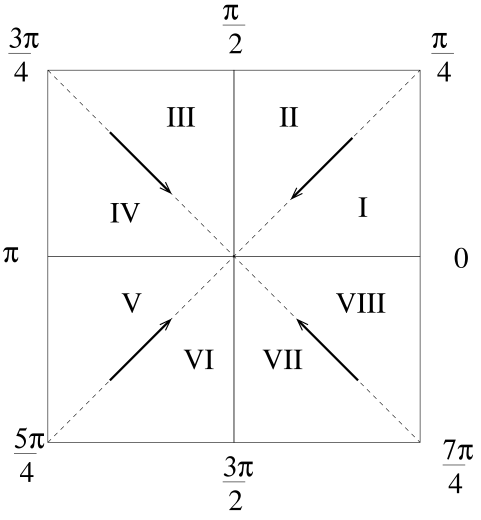

Fig. 3. Geometric description of nonsmooth feedback algorithm

The geometry of our problem is shown in Fig. 3. This system typifies a portion of a distributed manipulator system near a desired equilibrium point that is located at the origin. There are four wheel actuators, located at $(\pm 1, \pm 1)$, driving a planar object. In this section, we consider the underactuated case where the tangents to the wheel rims all point toward the origin, and their orientations are fixed (Section V considers a different approach that is made possible in the case of full actuation).

Let the control inputs be denoted by $u_1, \ldots, u_4$. An analysis of this system using the PDM method shows that the plane in the vicinity of the equilibrium can be divided into 16 distinct regions. In the eight open regions, labeled $\mathbf{I} - \mathbf{VIII}$, one unique contact state holds. The other 8 boundary regions, labeled $0 - 2\pi$ in increments of $\frac{\pi}{4}$ denote the boundaries between different contact states. In our modeling scheme, the dynamics are multi-valued on these boundaries. In each one of the regions $\mathbf{I} - \mathbf{VIII}$ a control law is calculated from the Lyapunov function $k(x^2 + y^2 + \theta^2)$ by solving $\dot{V} = -V$ for u_i, where k is a constant to be chosen[3]. The eight control laws for each quadrant can be found in Table I, and are labeled by their

[3] The value of k will influence the rate of convergence to the origin

their region of applicability. Note that for regions **I-VIII**, $u_3 = u_1$ and $u_4 = u_2$.

In previous work ([23], [26]) additional control laws were designed and deployed on the region boundaries. However, the constructive nature of the proof of Theorem III.3 makes this additional control laws unnecessary. Abstractly, we must know the location of the switching boundaries between the various regions to apply these methods. For situations where the object's mass distribution is well characterized, and the coefficients of friction reasonably well known, this is a reasonable assumption. However, when knowledge of such parameters is uncertain, the boundary geometries can then become uncertain, and the control law performance may degrade. Although we do not present it here, a related approach which uses scale independent hysteresis switching (based on work by [2], [17]) solves this problem [21].

Region	Control Law	Not Slipping
I	u_1 $\frac{-u4\,(\theta+x-y)+k\,\left(\theta^2+x^2+y^2\right)}{x+y}$ u_2 $k\theta$	1S 1R 4R
II	u_1 $\frac{u_2\,(\theta+x-y)+k\,\left(\theta^2+x^2+y^2\right)}{x+y}$ u_2 $-k\theta$	1S 1R 2R
III	u_1 $k\theta$ u_2 $\frac{u_1\,(\theta+x+y)-k\,\left(\theta^2+x^2+y^2\right)}{x-y}$	1R 2R 2S
IV	u_1 $-k\theta$ u_2 $-\frac{u_3\,(\theta+x+y)+k\,\left(\theta^2+x^2+y^2\right)}{x-y}$	2R 2S 3R
V	u_1 $\frac{u_2\,(\theta-x+y)-k\,\left(\theta^2+x^2+y^2\right)}{x+y}$ u_2 $k\theta$	2R 3R 3S
VI	u_1 $-\frac{u_4\,(\theta-x+y)+k\,\left(\theta^2+x^2+y^2\right)}{x+y}$ u_2 $-k\theta$	3R 3S 4R
VII	u_1 $k\theta$ u_2 $\frac{u_3\,(-\theta+x+y)+k\,\left(\theta^2+x^2+y^2\right)}{x-y}$	3R 4R 4S
VIII	u_1 $-k\theta$ u_2 $\frac{u_1\,(\theta-x-y)+k\,\left(\theta^2+x^2+y^2\right)}{x-y}$	4R 4S 1R

TABLE I

LIST OF REGIONS, CONTROL LAWS, AND ACTIVE CONSTRAINTS. THE REGIONS ARE DENOTED AS **I-VIII** AS IN FIG. 3. THE WHEELS ARE LABELED ACCORDING TO THEIR QUADRANT (I.E., WHEEL 1 IS IN QUADRANT 1,). ACTIVE (I.E., NONSLIPPING) CONSTRAINTS IN THE ROLLING DIRECTION ARE LABELED WITH R, WHILE ACTIVE CONSTRAINTS IN THE SIDEWAYS DIRECTIONS ARE LABELED WITH S (I.E., 1R INDICATED THAT WHEEL 1 DOES NOT SLIP IN ITS ROLL DIRECTION.).

Table I shows the control law for each octant as well as indicating which wheels are predicted to be slipping. In each case, the closest wheel to the center of mass must satisfy both the rolling and sideways slip constraint, and the second closest wheel must satisfy either the rolling or the slip constraint. In the case that $\mu_R > \mu_S$, this will always be the rolling constraint. Hence in region **I** the satisfied constraints are wheel 1 not slipping in either direction (1R 1S) and wheel 4 not slipping in the rolling direction (4R). These three constraints then uniquely determine the kinematic equations of motion.

V. Smooth Feedback for Fully Actuated Systems

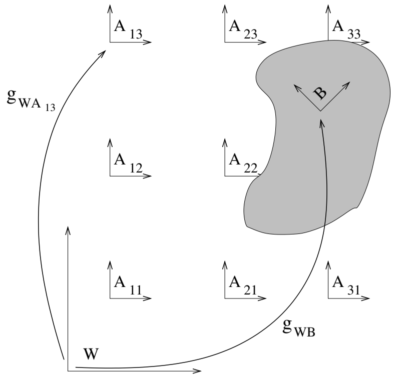

Fig. 4. Rigid body velocities

This section describes a stabilizing *smooth* feedback controller for the special case of fully actuated distributed manipulators. Note that this method provides global stability and exponential convergence to the desired equilibrium. Fig. 4, depicts an array of actuators (this models exactly a nine actuator experiment presented in Section VIII). Let W denote a fixed reference frame, let B denote a frame rigidly attached to the moving object, and let A_{ij} denote an "actuator frame" whose origin is fixed at the point of contact (x_i, y_j) between the actuator and the object. The frames A_{ij} have a fixed orientation with respect to W. Let the rigid body transformation from W to B be denoted by g_{WB} and the rigid body transformation from W to A_{ij} be denoted by $g_{WA_{ij}}$. Recall from Eq. (3) that the g_{ab} are defined by

$$g_{ab} = \begin{bmatrix} R(\theta) & \begin{bmatrix} x_{ab} \\ y_{ab} \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

where $R(\theta) \in SO(2)$ describes the relative orientation of frame b with respect to frame a, and x_{ab} and y_{ab} are the translations going from frame a to frame b. The relative velocity $V_{body} = (\dot{x}_{body}, \dot{y}_{body}, \dot{\theta}_{body})$ of a point above actuator A_{ij} on the body is:

$$Ad_{g_{WA_{ij}}}\dot{q} \tag{17}$$

where in $SE(2)$ the Adjoint operator Ad_g is defined by

$$Ad_g = \left[\begin{array}{cc} R(\theta) & \left(\begin{array}{c} y_{ab} \\ -x_{ab} \end{array} \right) \\ 0 & 1 \end{array} \right].$$

We adopt the following control Lyapunov approach to the control design. Suppose we are given a Lyapunov function on $SE(2)$, denoted by $V(\cdot)$, and define target dynamics of the form:

$$\dot{q} = -\frac{\partial V(q)}{\partial q}. \tag{18}$$

In the idealized situation where the contacts exactly track their commands, this system is trivially exponentially stable (we analyze the robustness of this approach below to disturbances and inaccuracies in the contact control). The velocity $\dot{q}$ is mapped to the actuators in order to obtain a feedback law.

For concreteness, we choose the Lyapunov function: $V(x, y, \theta) = k_1 x^2 + k_2 y^2 + k_3 \theta^2$ for $k_i > 0$.[4] Transforming the velocity into the actuator frame yields $Ad_{g_{WA_{ij}}} \cdot (-\frac{\partial V(q)}{\partial q})$. If the frames at the actuators are parallel to the world frame, then substituting in for $\frac{\partial V(q)}{\partial q}$ yields that the actuator velocities at the contact should be

$$\left[\begin{array}{c} k_3 y_i(\theta - \theta_d) - k_1(x - x_d) \\ -k_3 x_i(\theta - \theta_d) - k_2(y - y_d) \\ -k_3\theta \end{array} \right]$$

where x_d, y_d, θ_d are the desired values and x, y, θ are the state feedback values. Assuming as we did previously that the i^{th} actuator is a wheel of radius R with input $\frac{1}{R}u_i$, we can transform these contact velocities into desired wheel velocities and wheel orientations for this particular example by simply calculating the magnitude and direction of the contact velocity for each actuator:

$$\theta_{ij} = \tan^{-1}\left(\frac{-k_3 x_i(\theta - \theta_d) - k_2(y - y_d)}{k_3 y_i(\theta - \theta_d) - k_1(x - x_d)} \right) \tag{19}$$

$$v_{ij} = \sqrt{\begin{array}{r} (-k_3 x_i(\theta - \theta_d) - k_2(y - y_d))^2 \\ +(k_3 y_i(\theta - \theta_d) - k_1(x - x_d))^2 \end{array}} \tag{20}$$

where θ_{ij} is the orientation of the (i, j) actuator and v_{ij} is the wheel velocity of that actuator. So, given all the actuator locations, one computes Eqs. (19) and (20) for each actuator, and the feedback law is complete. Note that this algorithm scales nicely with the number of actuators, and hence it would be suitable for implementation on real devices such as MEMS arrays. However, the full actuation requirement may not be reasonable on the MEMS scale.

Naively, the system controlled by this feedback law will be stable because of Eq. (18) and the idealization that the actuators follow the desired velocities. However, in practice there will be small perturbations to the actuator orientations and velocities due to disturbances and modeling errors. These small errors can cause unpredictable switching between the contact states. We would like to ensure that the scheme remains stable and robust when these errors are taken into account.

Now we consider the robustness of this feedback law with respect to the multiple model system that arises when the actuators are not all perfectly aligned with the desired motion. This captures the possibility that the actuators do not exactly follow their commanded motion. In this case, some contact slip will occur. Let the controls obtained above be termed the *desired reference* controls u_d^k, and let the actually realized controls be denoted by u^k. The difference between the behavior of the idealized system and the actual system is governed by the following theorem.

Theorem V.1. *There exists a* $\delta > 0$ *and* $T > 0$ *such that if* $|u^k(t) - u_d^k(t)| < \delta \;\; \forall\, k$ *and* $\forall t > T$*, the solutions to the MMDA system given by the PDM are exponentially stable using the controls from Equations (19) and (20).*

Proof: First, we know that for the choice of controls u_d^k we have

$$\dot{V} = \frac{\partial V}{\partial q}\dot{q} < 0.$$

Let $B(\dot{q}, \epsilon) \subset T_q Q$ denote a ball of radius ϵ centered at $\dot{q}$. Therefore, for an $\dot{q}_\epsilon \in B(\dot{q}, \epsilon)$,

$$\dot{V} = \frac{\partial V}{\partial q}\dot{q}_\epsilon < 0 \;\; \forall \;\; \dot{q}_\epsilon \in B(\dot{q}, \epsilon).$$

(This is a simple consequence of the continuity of the expression $\dot{V}$ along a continuous path between $\dot{q}$ and any other element of $T_q Q$.)

A sufficient condition for stability of a multiple model system is that all of the individual models not only be individually stable, but additionally all the models have a commonly valid Lyapunov function (see [26]). We will use this fact to show that for sufficiently small δ all the multiple models will be in $B(\dot{q}, \epsilon)$, thereby ensuring overall stability of the nonsmooth system.

For a given set of inputs u^k there is a corresponding set of kinematic constraints $\omega_i(q)$. Moreover, the PDM implies that the subset of these constraints that satisfy $\omega_i(q)\dot{q} = 0$ will define the governing equations. When $u^k = u_d^k$, we get precisely the desired dynamics. Because these kinematic constraints $\omega_i(q)$ depend continuously on the inputs, for any choice of ϵ' limiting how much we will allow the $\omega_i(q)$ to vary (and hence how much $\dot{q}$ can vary), we can always choose a δ such that $|u^k(t) - u_d^k(t)| < \delta$ accordingly. Therefore, we can always choose δ small

[4] The constants k_i set the rate of convergence to the desired equilibrium.

enough such that $\dot{q} \in B(\dot{q}_d, \epsilon)$. This completes the proof. ■

This theorem implies that even if the actuators start out in a kinematically incompatible state, as long as they converge to within some δ of the desired actuator state, the system will keep its stability properties. Feedback control of the driving mechanism states can easily accomplish this requirement. We should also note that this can easily be extended to exponential stability in a similar fashion.

Experimental results in Section VIII illustrate that this method works well. However, in the case where one does not have full actuation, one must ask if this control law has any analogs. In general the answer is no–there are generally no smooth control laws for systems that are not fully actuated. Hence, the nonsmooth control laws from Section IV are potentially the only way to stabilize distributed systems of the type described here.

VI. A Globally Stable Feedback Algorithm

The last section presented an algorithm for the specialized case of full actuation that fortuitously had the property of global stability. This section blends the PFF method [31], [8] with the local feedback law of Section IV to create a globally stable feedback algorithm that will work for underactuated systems. In this approach, a programmable force field is used to govern the gross motions of the object far away from the equilibrium point, and a local stabilizing feedback law from Section IV is used in the vicinity of the equilibrium configuration. The intuition behind this result is that even though the PFF is unstable in orientation, in order to move a package from one point a in the plane to an equilibrium point b, the package can be allowed to spin freely along its path to b, and one can wait to control the package's orientation after it has come sufficiently close to b. Consider Fig. 5. If a is in the upper right-hand corner, then Theorem III.3 implies that an object starting at a will eventually arrive in the *feedback region* $\mathcal{M}$ (in the middle of Fig. 5) under the programmable force field method— since translational motions (but not rotational motions) are stable even when discrete frictional contacts are taken into account. From a practical point of view this means that as long as one has no performance goals for the orientation θ outside of $\mathcal{M}$, one does not need feedback outside of $\mathcal{M}$. Since sensing and feedback is often expensive to implement, this approach efficiently uses sensing and feedback where it is most effective.

A. A LaSalle Result

To prove stability of the combined control system while taking into account the multiple model aspect of the governing equations, an extension of Lasalle's theorem is required. The basic difference between the classical version of the LaSalle invariance theorem and the one found here is that here we must consider systems governed by differential inclusions. In such systems, the idea of a "flow" does not include uniqueness. That is, rather than having a result for *the* unique flow $\phi(t)$, it must be valid for any flow $\phi(t)$, satisfying $\dot{\phi} \in F(t,x)$, which is in general a *set* of possible trajectories.

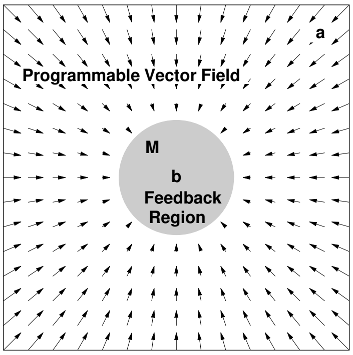

Fig. 5. A LaSalle invariance theorem

Theorem VI.1. *Let $\mathcal{M}$ be the "feedback region," a compact simply connected subset of $\mathbb{R}^2$. Let $V(x)$ be a Lyapunov function on $\mathcal{M}$ and let $F(x,t)$ be a convex set-valued map. Let $\phi_t(x_0)$ denote a flow that satisfies $\dot{x} \in F(x,t)$, starting from x_0. Let $\mathcal{M}$ be a positively invariant compact set under all flows $\phi_t(p)$ satisfying the differential inclusion $\dot{x} \in F(x,t)$ ($\mathcal{M}$ is positively invariant if $\dot{V}^*(x) \le 0$ for all $x \in \mathcal{M}$, where $\dot{V}^*$ is defined in Eq.* (14)*). Let*

$$\begin{aligned} E &= \{x \in \mathcal{M} \mid 0 \in \dot{V}(x)\} \\ N &= \{\bigcup \phi(t,x_0) \mid x_0 \in E \ and \ \phi(t) \in E \ \forall t > 0\} \end{aligned}$$

Then, for all $x \in \mathcal{M}$, $\phi(t,x) \to N$ as $t \to \infty$.

That is, E is the set on which the time derivative of the Lyapunov function is zero, and N is the union of all trajectories that start in E and remain in E for all $t > 0$.

Proof. This proof is roughly patterned on the proof of LaSalle's Invariance Theorem found in [33]. First recall that an ω-limit point of a differential inclusion (or differential equation) and a point $p \in \mathbb{R}^n$ is defined as a point $q \in \mathbb{R}^n$ where for all solutions $\phi_t(p)$ to the differential inclusion $\dot{x} \in F(x,t)$ $\exists\, t_1, \ldots, t_i$ with $i \uparrow \infty$ such that $\phi(t_i) \to q$ as $i \uparrow \infty$. The ω-limit set is the collection of such points, and is denoted $\omega(p)$. Note that this definition does not in any sense require solutions to be unique.

It must be shown that $\dot{V} = 0$ on $\omega(p)$ ($\forall p \in \mathcal{M}$), or equivalently that V is constant on $\omega(p)$. Assume q is an ω-limit point of the differential inclusion, then set $V(q) = c_q$. First we will need the following fact about ω-limit sets.

Lemma VI.2. *$\omega(p)$ is invariant under the flow of F.*

Proof. Let $q \in \omega(p)$ and $q_s = \phi_s(q)$. We first must consider if the map $\phi_s(\cdot)$ exists for all s. First, note that since $\mathcal{M}$ is compact and F is compact and convex, we have existence of $\phi_s(\cdot)$ for $s \in (0, \infty)$ (this is a natural extension of the classical result for ODEs - see [12, pages 77-86]). Now we show that it is true for $s \in (-\infty, 0)$. Using the fact that the limit of any uniformly convergent sequence of solutions to a compact, convex differential inclusion is also a solution (see Lemma 1 in [12, page 76]), we can choose a sequence $\{t_i\}$ with $t_i \to \infty$ as $i \uparrow \infty$ such that $\phi_{t_i}(p) \to q$ as $i \uparrow \infty$ (this is by definition of $\omega(p)$). Then using the fact that $\phi_s(\phi_{t_i}) = \phi_{s+t_i}$(almost everywhere) as one takes the limit $i \uparrow \infty$, we get that $\phi_s(q)$ exists for $s \in (-\infty, 0)$. Again, note that this argument does not require solutions to be unique, and that in general ϕ_s will be multivalued.
Given the existence of the map $\phi_s(\cdot)$ for all s, we can choose a sequence $t_1, \ldots, t_i$ with $i \uparrow \infty$ such that $\phi_{t_i} \to q$ as $t \to \infty$. Then the map $\phi_{t_i+s}(p) = \phi_s(\phi_{t_i}(p))$ converges to q_s as $i \uparrow \infty$. This implies that $q_s \in \omega(p)$ and $\omega(p)$ is therefore invariant. □

Lemma VI.2 implies that $c_q = \inf\{V(\phi_t(x)) | t \geq 0\}$ because $\dot{V}^* \leq 0$ everywhere in $\mathcal{M}$. Thus, $V(\phi_t(q)) = c_q$ (because $q \in \omega(p)$), so $0 \in \dot{V}^*$ on $\omega(p)$. Therefore $\omega(p) \subset E$. Again, because of the above fact that $\omega(p)$ is invariant, $\omega(p) \subset N$, where N is the union of trajectories that start and remain in E. This leads us to the fact that $\phi_t(x) \to N$ as $t \to \infty$, the desired result. □

To apply these results to distributed manipulation, one must only show that a distributed manipulator will satisfy the requirements and assumptions of Theorem VI.1. This will lead to the following Corollary of Theorem VI.1. Assume the distributed manipulation system can be represented by an array of actuators a_{ij} whose contacts are located at coordinates (x_i, y_j). Further assume that the PDM model solution depends only on the center of mass (equivalently, that the coefficient of friction is uniform). For us, $\mathcal{M}$ will be the feedback region of the distributed manipulator, that is, the area in which one has some sort of state feedback available.

Corollary VI.3. *Given a discrete planar array geometry, an elliptic vector velocity field $\Psi(x, y) : \mathbb{R}^2 \to \mathbb{R}^4$ outside of $\mathcal{M} = B_\epsilon \times S_1$ for some $\epsilon > 0$, and a locally stabilizing feedback law (such as the one in Theorem III.3) the solution to the governing equations given by the PDM is globally stable.*

Proof. Assume that the desired equilibrium point is always in $\mathcal{M}$. This implies that since $\mathcal{M} \subset SE(2)$, then $\mathcal{M} = B_\epsilon \times S_1$ where B_ϵ is the ϵ-ball in $\mathbb{R}^2$, and $\mathcal{M}$ is therefore compact. Therefore the first part of Theorem VI.1 is supplied. It is reasonably straightforward to show that $\mathcal{M}$ is positively invariant using an elliptic vector field. Moreover, for a choice of $V = \|q\|^2$, E consists solely of the origin. This implies that the origin is stable. In fact, asymptotically stable, because $\mathcal{M}$ is reached in finite time, and once inside $\mathcal{M}$ the origin is asymptotically stable by Theorem III.3. □

The following corollary indicates that the induced instability of the programmable force field approach can be corrected with a local feedback law, and that moreover the performance can be made exponential.

Corollary VI.4. *Given a discrete planar array geometry, an elliptic vector velocity field $\Psi(x, y) : \mathbb{R}^2 \to \mathbb{R}^4$ outside of $\mathcal{M} = B_\epsilon \times S_1$ for some $\epsilon > 0$, and an exponentially stable local control law, then the resulting system (11) is exponentially stabilizable.*

Proof. First, note that Theorem III.3 already showed local exponential stability. It will be shown that exponential stability can be maintained outside of $\mathcal{M}$. Corollary III.3 shows that the origin is globally stable. All we need to show is that there is an exponential $k_3\|q\|e^{-st}$ which provides an upper bound on $\dot{V}^*$. From Section III-B we know that outside $\mathcal{M}$ the x and y coordinates can be exponentially stabilized, but the θ coordinate is only neutrally stable. Therefore, the maximum value of $\|x, y, \theta\|$ is $d(\partial\mathcal{M}, 0) + \pi^2$, where $d(\cdot, 0)$ is the maximum distance from the origin to the boundary of a set. Setting $k_3 = d(\partial\mathcal{M}, 0) + \pi^2$ it is clear that outside of $\mathcal{M}$ the solutions converge exponentially to $\mathcal{M}$, and inside $\mathcal{M}$ we have already shown that the origin is exponentially stable. Therefore the origin is globally exponentially stable. □

VII. An Experimental Distributed Manipulation System

In order to test our algorithms, as well as those of other researchers, we developed a modular experimental distributed manipulator test-bed. For more details, see [21]. A schematic of the system architecture and a photograph of the system can be seen in Fig. 6. This system has some similarities to other experimental devices (notably [20], [19]). However, it incorporates visual feedback and is more highly articulated by virtue of being able to exert forces in arbitrary directions in the plane even with contact dynamics. Without considering contact dynamics (by using low friction interfaces) this experiment and the experiment in [20], [19]) have nearly identical effects. Our modular design is based on a cell concept. Each cell contains two actuators. One actuator drives a wheel that contacts the moving object, while the other actuator orients the wheel axis, (see Fig. 7). Note that the orienting axis of each cell can be fixed so that our system can simulate underactuated systems. The driving wheel has a four-inch radius, and is made of soft foam rubber to accentuate the friction reaction force. These wheels satisfy the preferred friction distribution rule described in Section II.

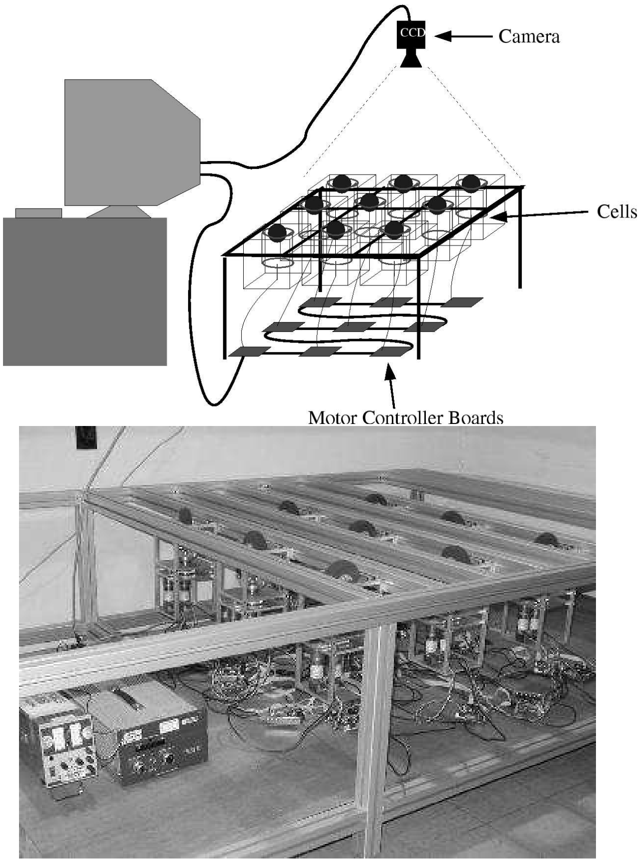

Fig. 6. The FADM System. Left: schematic of system architecture. Right: photograph of the system.

These cells can be easily repositioned into different configurations in the supporting modular table structure so as to simulate different types of systems. As seen in Fig. 6, the Fully Actuated Distributed Manipulation (FADM) system is deployed with a total of nine cells. More cells can be added as needed.

Both actuators of each cell consist of Pittman brushless 12V motors, which are connected to JR-Kerr Pic-Servo-3PH motor controller boards. All 18 motor boards are connected through a daisy chain configuration to a central computer through one of its serial ports. The position of the manipulated object is obtained and tracked visually. A Sony XC-73 monochrome CCD camera with a Cosmicar C60607 6 mm lens is used for the vision system. Images are captured by an Imagenation PXC-200 framegrabber card. For visual position acquisition and tracking, a right triangle is affixed to the moving object.

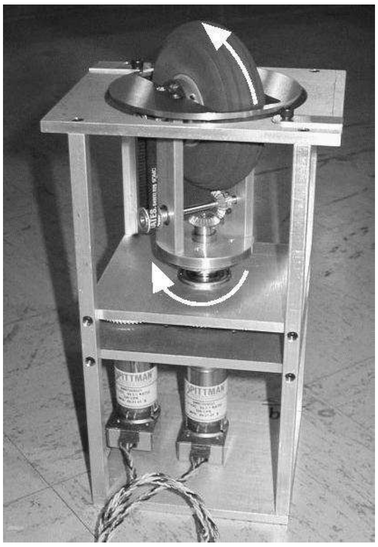

Fig. 7. Photograph of cell. The actuated motions are marked in blue.

Feature tracking software (written in C) developed at Caltech's Computational Vision Laboratory (see [14]) is used to find and track this triangle. Because of the communication delays required to send control signals to all motor controller boards in the daisy chain system, at present only six to seven iterations of the feedback algorithm can be realized per second.

VIII. Experimental Results

Here we summarize experimental results that illustrate the theory reviewed in the previous sections. The goal of each experiment is to stabilize an object from a random initial condition to the final configuration $(x_f, y_f, \theta_f) = (0\text{ m}, 0\text{ m}, 0\text{ rad})$. Each experiment implements a different algorithm that was reviewed above. For these experiments, the manipulated object is a piece of clear plexiglass. The object's transparency allows us to view the actuator movements during manipulation while still ensuring reasonable amounts of friction. To enable visual localization and tracking, a white piece of paper with a black triangle is affixed to the object (see the movie snapshots in Figs. 9,11,13,15. These snapshots also include outlines of the "goal" triangle position as well as outlines of the plexiglass. We should note that these experiments are just example outcomes, and that after many trials the only failures we observed were associated with the vision system losing track of the object. Movies of these experiments can be found in Extensions 1-4. The following paragraphs summarize each experiment and its result.

A. A Programmable Force Field Experiment

To verify the instability of the PFF method in object orientation, this experiment uses a 9 cell arrangement to implement an open loop elliptic vector velocity field

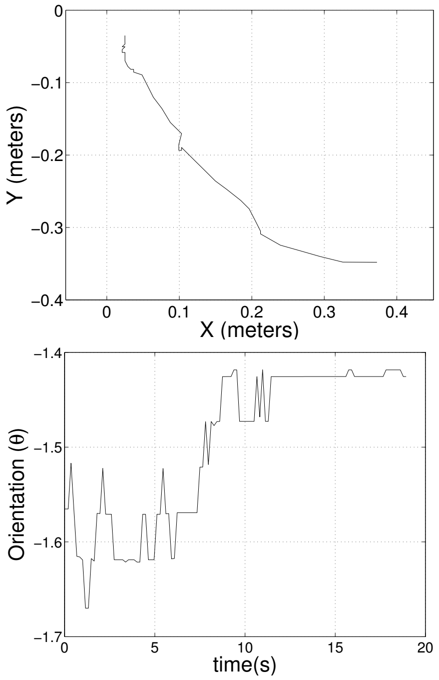

Fig. 8. Open Loop control

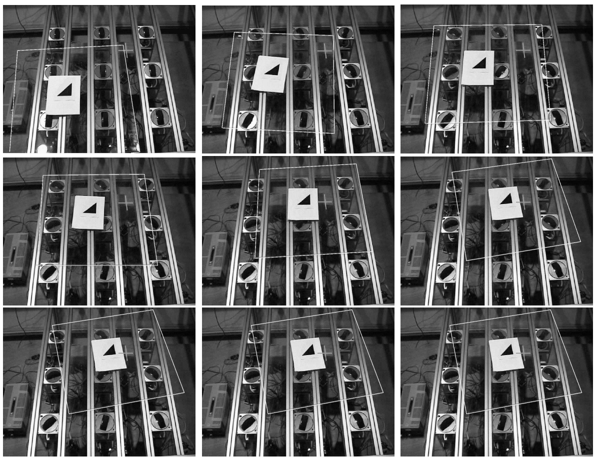

Fig. 9. Open Loop movie snapshots

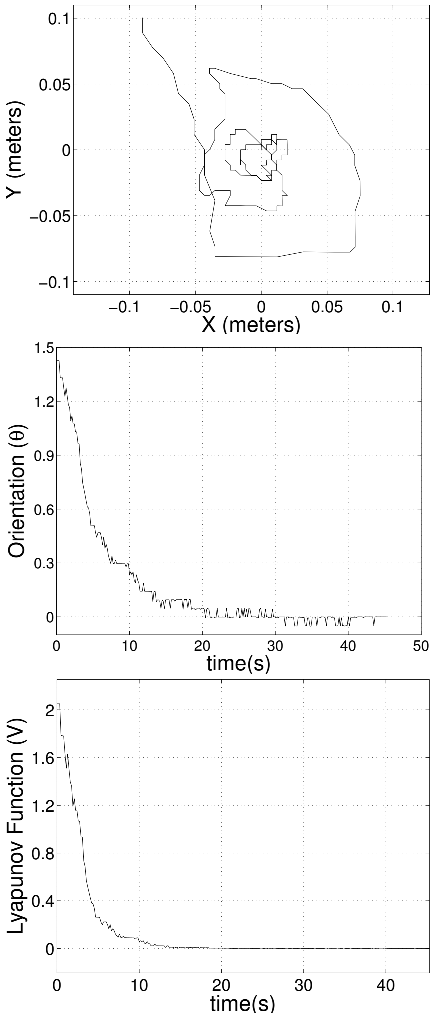

Fig. 10. Under-actuated feedback control

designed so that the long axis of the plexiglass would be aligned with the desired orientation. The x, y, and θ coordinates of the object's actual trajectory is shown in Fig. 8. This experiment used an initial state of approximately $(x_0, y_0, \theta_0) = (0.25\text{ m}, -0.35\text{ m}, -1.5\text{ rad})$, and the final position was approximately $(x_f, y_f, \theta_f) = (0.01\text{ m}, 0.01\text{ m}, -1.5\text{ rad})$. Notice that the moving object did not come anywhere close to realizing the desired final orientation, and in fact was little reoriented despite the fact that we were commanding a reorientation of approximately $\frac{\pi}{2}$ radians. Nevertheless, the PFF method came close to successfully stabilize the x and y translational coordinates of the object's location. In order to achieve this (x, y) stabilization we had most of the wheels slipping at various times. This causes significant stress on both the object being manipulated and the actuators themselves. The methods we present in the previous sections assume that if the actuators are moving at sufficiently slow velocities some, but not necessarily all, actuators are always in contact with the moving body. This requires less energy for a given motion and moreover induces smaller forces on the object and actuators. We should also make clear that this experiment in no way proves that *other* open loop methods may not be successful, but is intended for purposes of comparison to the results that follow. See Extension 1 for a movie of this experiment.

B. Local Nonsmooth Feedback for Underactuated Systems

This experiment duplicates the geometry of Fig. 3 and implements the algorithm of Section IV. The goal is to

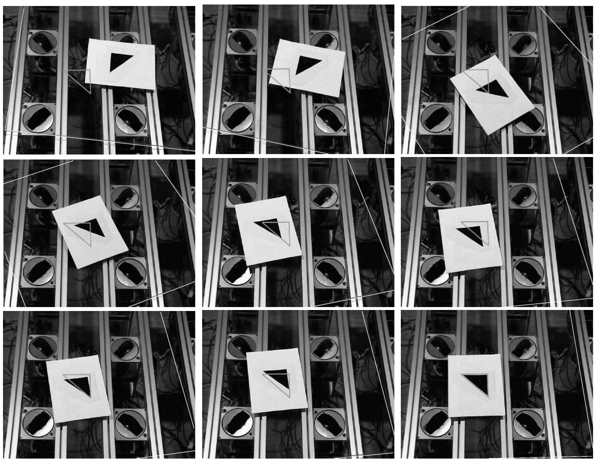

Fig. 11. Under-actuated movie snapshots

stabilize the supported object to the origin under the strict underactuated conditions that the wheel orientations are fixed. The object's actual trajectory is shown in Fig. 10. This experiment used an initial state of approximately $(x_0, y_0, \theta_0) = (-0.9 \text{ m}, .1 \text{ m}, 1.5 \text{ rad})$, and the final position was approximately $(x_f, y_f, \theta_f) = (0.01 \text{ m}, 0.01 \text{ m}, 0.05 \text{ rad})$ (the angle cannot currently be resolved any further at this point because of camera pixelization error). Notice how smoothly the θ variable is driven to the desired value of 0. This is somewhat surprising considering that the θ variable is precisely the unstable mode in the open loop case (see [23]). Conversely, notice the gyrations in the x and y coordinates as the object is stabilized. These peculiar features can be understood from Remark III.1. As the object gets very close to, but is not precisely situated at the origin of $\mathbb{R}^2$, the system becomes difficult to control since for this system not all points in the vicinity of the origin are stabilizable. Consequently, if the object's trajectory approaches too closely to the origin of $\mathbb{R}^2$ before the θ value is very near the desired one, the controls naturally "push" the system away from the origin of $\mathbb{R}^2$ until the orientation is close to the desired value. Moreover, the Lyapunov function V is decreasing along the trajectory, indicating that the control laws are behaving as expected. See Extension 2 for a movie of this experiment.

C. Smooth Feedback for Full Actuation

This experiment implements the methods of Section V on a nine cell configuration. Fig. 12 shows the trajectory realized by the object when using this control law under the assumption of full actuation. This experiment used an initial state of approximately $(x_0, y_0, \theta_0) = (0.3 \text{ m}, -0.4 \text{ m}, 1.8 \text{ rad})$, and the final position was approximately $(x_f, y_f, \theta_f) = (0.01 \text{ m}, 0.01 \text{ m}, 0.05 \text{ rad})$. Notice the smoothness of the trajectory. This experiment

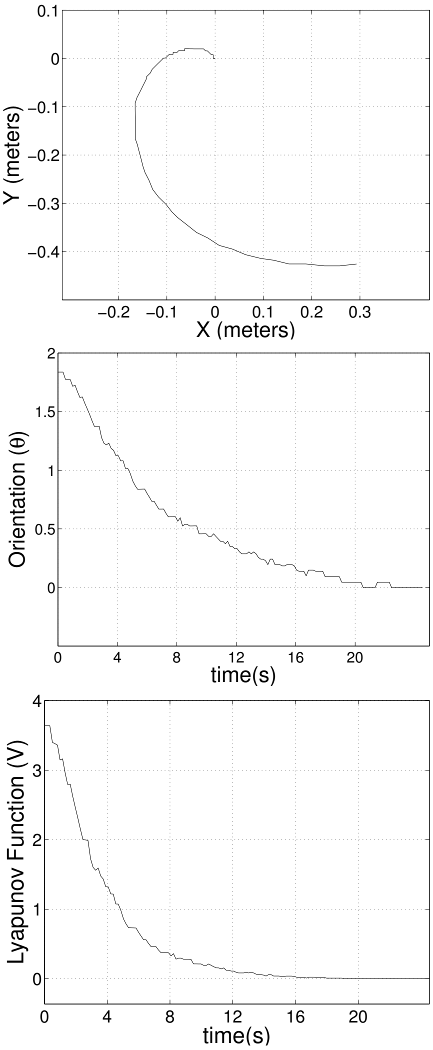

Fig. 12. Smooth feedback control

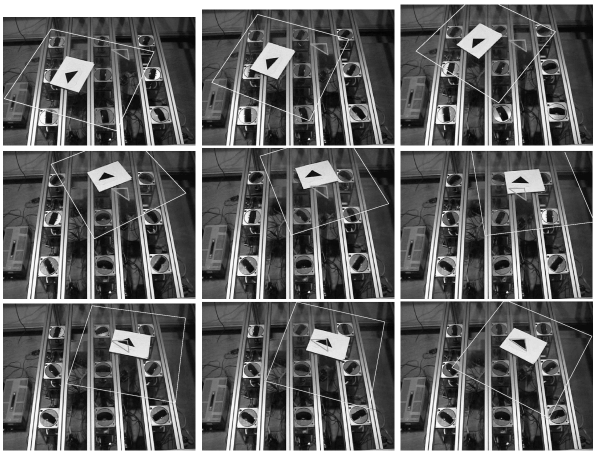

Fig. 13. Smooth feedback movie snapshots

indicates that when a distributed array is fully actuated, the feedback law in Eq. (20) works extremely well. More importantly it is computationally very simple, and the number of computations scales linearly with the number of actuators. The feedback law has good disturbance rejection properties, as can be seen by the fact that the object is stabilized despite the fact that the actuator initial conditions are not compatible with the desired motion, hence verifying the result in Theorem V.1. Also note that although the (x, y) trajectory has "overshoot," the Lyapunov function V is steadily decreasing. See Extension 3 for a movie of this experiment.

D. The Globally Stabilizing Controller of Section VI

This experiment used nine FADM actuators arranged in a regular array (see Fig. 15) to implement the algorithm of Section VI. The feedback region consisting of a 0.25 meter radius circle centered on the goal position. Outside of this region, object motions are governed by a programmable force field. Inside of this region, we allow the wheels to be fully actuated as in Section V and use a locally stabilizing feedback law that takes advantage of the fact that all of the wheels can be individually steered. This leads to exceptional performance in $\mathcal{M}$. This experiment used an initial state of approximately $(x_0, y_0, \theta_0) = (-0.3 \text{ m}, 0.2 \text{ m}, 2.1 \text{ rad})$, and the final position was approximately $(x_f, y_f, \theta_f) = (0.01 \text{ m}, 0.01 \text{ m}, 0.05 \text{ rad})$. See Extension 4 for a movie of this experiment.

IX. SUMMARY

This paper introduced a fundamentally different approach to the control of distributed manipulation than that found in [8], [20], [31]. Much prior work in distributed manipulator control, particularly the use of programmable force fields, has not paid sufficient attention to the effects of friction and intermittent contact. We have shown here these effects lead to nonsmooth dynamical behavior, and are non-negligible. However, we introduced tractable non-smooth analysis and control methods to capture these effects and control them.

While the control algorithms we presented are nonsmooth, they are relatively simple, improving the likelihood that these control laws can be scaled up to high numbers of actuators. Moreover, they were shown to be exponentially stable, which provides bounds on the time it takes an object to reach the goal configuration. In the case of a fully actuated distributed manipulator, a smooth control law ensures good performance which is robust with respect to small errors in actuation. A small disadvantage of these methods is that they require local feedback near the desired equilibrium. The experimental results presented here confirm that these modeling and control methodologies are valid in practice.

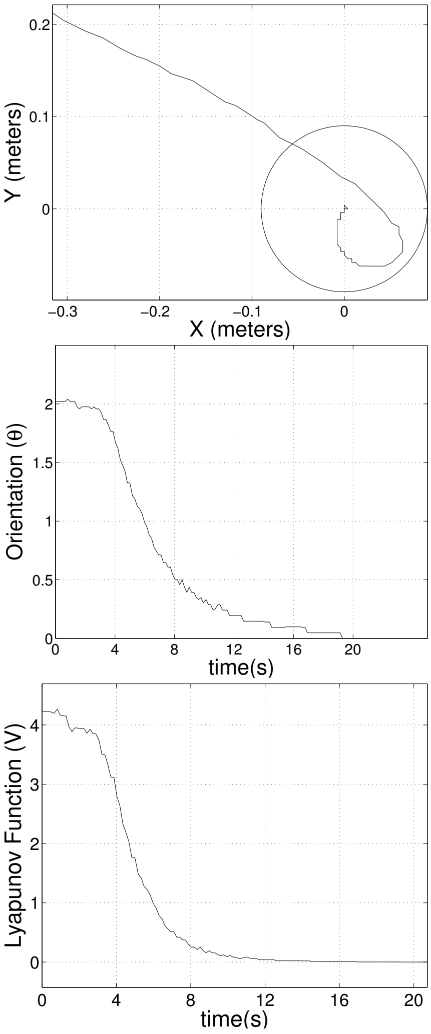

Fig. 14. Combining the programmable force field with local feedback

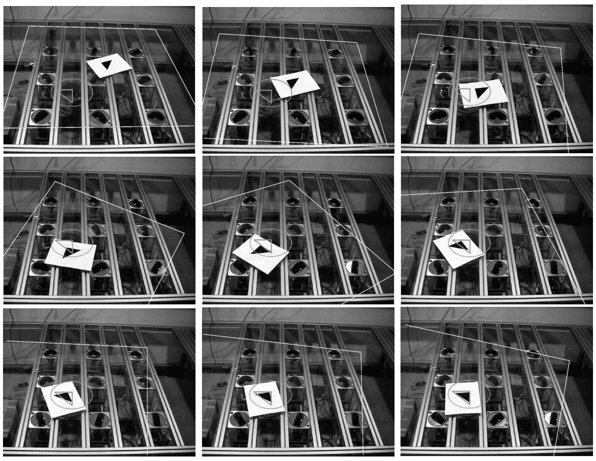

Fig. 15. Global stabilization movie snapshots

A next important step will be to implement the theory demonstrated here on the micro-scale, where friction and limited actuation are very important factors in control. We hope that our methods will in the future be adapted to physical conditions at the MEMS level in cases where the manipulated object is at the same scale as the MEMS actuators. Such an adaptation could then be applied to making micro assembly and other micro tasks more feasible.

Acknowledgements: This work was largely supported by the National Science Foundation through grant NSF9402726 to the Caltech Center for Neuromorphic Systems Engineering. The authors thank thank James Burgess, David Choi, and Andrew Homyk for all their help in constructing and carrying out the experiments describe here.

References

[1] J.C. Alexander and J.H. Maddocks. On the kinematics of wheeled vehicles. *The International Journal of Robotics Research*, 8(5):15–27, October 1989.

[2] B.D.O. Anderson, T.S. Brinsmead, F. De Bruyne, J.P. Hespanha, D. Liberzon, and A.S. Morse. Multiple model adaptive control. I. finite controller coverings. *George Zames Special Issue of the Int. J. of Robust and Nonlinear Control*, 10(11-12):909–929, Sep 2000.

[3] M. Ataka, A. Omodaka, and H. Fujita. A biomimetic micro motion system. In *Transducers - Digest International Conference on Solid State Sensors and Actuators*, pages 38–41, 1993. Pacifico, Yokohama, Japan.

[4] D.K. Biegelsen, A. Berlin, P. Cheung, M.P.J. Fromherz, D. Goldberg, W. Jackson, B. Preas, J. Reich, and L.E. Swartz. Air-jet paper mover: an example of mesoscale mems. In *SPIE Proceedings, Micromachined Devices and Components VI*, pages 122–129, 2000.

[5] K.F. Böhringer, R. G. Brown, B. R. Donald, J.S. Jennings, and D. Rus. Sensorless manipulation using transverse vibrations of a plate. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1989–1996, Nagoya, Japan, 1995.

[6] K.F. Böhringer, B. Donald, and N. MacDonald. Upper and lower bounds for programmable vector fields with applications to mems and vibratory plate parts feeders. In *Algorithms for Robotic Motion and Manipulation*, pages 255–276. A.K. Peters, Ltd., 1997.

[7] K.F. Böhringer, B. R. Donald, and N. C. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, San Diego, California, February 1996.

[8] K.F. Böhringer, B.R. Donald, L.E. Kavraki, and F. Lamiraux. A distributed, universal device for planar parts feeding: unique part orientation in programmable force fields. In *Distributed Manipulation*, pages 1–28. Kluwer, 2000.

[9] K.F. Böhringer, R.G. Donald, N.C. MacDonald, G.T.A. Kovacs, and J.W. Suh. Computational methods for design and control of mems micromanipulator arrays. *IEEE Computer Science and Engineering*, pages 17–29, 1997.

[10] F.H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, 1990.

[11] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4), 1988.

[12] A.F. Filippov. *Differential Equations with Discontinuous Right-Hand Sides*. Kluwer, 1988.

[13] K.Y. Goldberg. Orienting polygonal parts without sensing. *Algorithmica*, 143(2/3/4):201–225, 1993.

[14] Luis Goncalves and Enrico Di Bernardo. Software for motion tracking. Developed at the Caltech Vision Lab, 2000.

[15] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction. Part 1. limit surface and moment friction. *WEAR*, 143:307–330, 1991.

[16] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction. Part 2. dynamics of motion. *WEAR*, 143:331–352, 1991.

[17] J.P. Hespanha, D. Liberzon, A.S. Morse, B.D.O. Anderson, T.S. Brinsmead, and Franky De Bruyne. Multiple model adaptive control, part 2: Switching. *Int. J. of Robust and Nonlinear Control Special Issue on Hybrid Systems in Control*, 11(5):479–496, April 2001.

[18] J. Luntz, W. Messner, and H. Choset. Stick-slip operation of the modular distributed manipulator system. In *Proc. American Control Conference (ACC)*, 1998.

[19] J. Luntz, W. Messner, and H. Choset. *Distributed Manipulation*, chapter Discreteness Issues in Actuator Arrays. Kluwer Academic Publishers, 2000.

[20] J.E. Luntz, W. Messner, and H. Choset. Distributed manipulation using discrete actuator arrays. *Int. J. Robotics Research*, 20(7):553–583, July 2001.

[21] T. D. Murphey. *Control of Multiple Model Systems*. PhD thesis, California Institute of Technology, May 2002.

[22] T. D. Murphey and J. W. Burdick. Issues in controllability and motion planning for overconstrained wheeled vehicles. In *Proc. Int. Conf. Math. Theory of Networks and Systems (MTNS)*, Perpignan, France, 2000.

[23] T. D. Murphey and J. W. Burdick. On the stability and design of distributed systems. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.

[24] T. D. Murphey and J. W. Burdick. A controllability test for multiple model systems. In *Proc. IEEE American Controls Conference (ACC)*, Anchorage, Alaska, 2002.

[25] T. D. Murphey and J. W. Burdick. Feedback control for distributed manipulation. In *Workshop on the Algorithmic Foundations of Robotics*, 2002.

[26] T. D. Murphey and J. W. Burdick. Global exponential stabilizability for distributed manipulation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington D.C., 2002.

[27] T. D. Murphey and J. W. Burdick. Nonsmooth controllability and an example. In *Proc. IEEE Conf. on Decision and Control (CDC)*, Washington D.C., 2002.

[28] T.D. Murphey and J.W. Burdick. The power dissipation method and kinematic reducibility of multiple model robotic systems. *IEEE Transactions on Robotics and Automation*, 2003. Submitted.

[29] K.S.J. Pister, R. Fearing, and R. Howe. A planar air levitated electrostatic actuator system. In *Proceedings IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 67–71, Napa Valley, California, 1990.

[30] D. Reznik, J. Canny, and N. Alldrin. Leaving on a plane jet. In *Int. Conf. on Intelligent Robots and Systems (IROS)*, Maui, Hawaii, October 2001.

[31] A. Sudsang and L. Kavraki. A geometric approach to designing a programmable force field with a unique stable equilibrium for parts in the plane. In *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, 2001.

[32] J. W. Suh, R. B. Darling, K. F. Böhringer, B. R. Donald, H. Baltes, and G. T. A. Kovacs. Cmos integrated organic ciliary array as a general-purpose micromanipulation tool for small objects. *ournal of Microelectromechanical Systems*, 8(4):483–496, December 1999.

[33] S. Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer Verlag, 1990.

Appendix

The multimedia extensions to this article are at: *http://www.ijrr.org*.

Extension	Media Type	Description
Extension 1	Video	This is a video of an experimental implementation of an open loop elliptic velocity field. Recall that the object being manipulated is a clear piece of plexiglass with a white and black piece of paper on top for feedback. As predicted, the x and y coordinates are successfully stabilized and the θ dynamics are not stabilized.
Extension 2	Video	This is a video of an experimental implementation of the under-actuated system. Notice that despite the under-actuation, the plexiglass is stabilized to the origin at the desired orientation.
Extension 3	Video	This is a video of an experimental implementation of the fully actuated system. Here the advantage of full actuation is clear. The trajectory to the desired equilibrium is quite smooth and the wheels are all coordinated without any slipping.
Extension 4	Video	This is a video of an experimental implementation of the globally stabilizing controller that combines the open loop techniques with the closed loop techniques. Outside of the circle shown in the video, the system is running open loop. However, as soon as the center of mass of the plexiglass enters the circle, the closed loop control is turned on and the plexiglass is stabilized to the origin at the correct orientation using the full actuation control laws.