Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Experiments in Nonsmooth Control of Distributed Manipulation

T. D. Murphey, J. W. Burdick, J. Burgess, A. Homyk
Engineering and Applied Science, California Institute of Technology
Mail Code 104-44, Pasadena, CA 91125 USA
{murphey,jwb,burgess,homyk}@robotics.caltech.edu

***Abstract*— This paper describes an experimental modular distributed manipulation system upon which one can implement a variety of control schemes. We have shown elsewhere that when one includes the nonsmooth effects of friction into a model of distributed manipulation, nonsmooth feedback laws must generally be used to control distributed manipulators. We summarize results obtained with this experimental system that confirm the validity of control schemes proposed by the authors in recent papers (see [13, 14]). We describe the control algorithms in some detail and include specifics of the experimental set-up and experimental results.**

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 ranges from air jets, rotating wheels, and electrostatics on the macroscale, to MEMS and flexible scilia at the microscale.

As reviewed below, a number of strategies have been proposed for controlling distributed manipulation systems. In this paper we describe an experimental test-bed that was designed to evaluate and validate such control systems. Our modular system can emulate a reasonably large class of distributed manipulators that generate motion through rolling and sliding frictional contact between the moving object and actuator surfaces. In such cases friction forces and intermittent contact play an important role in the overall system dynamics, leading to non-smooth dynamical system behavior. In previous work we presented non-smooth dynamical models for describing such systems [11], and non-smooth control laws [13, 14] that provably stabilize these systems. We have applied these techniques to our experimental system, and experimental results presented in this paper validite these methods.

Methods to design distributed manipulation control systems have been proposed in several works, including [3, 5, 6]. A common approach is based on the notion of *programmable vector fields* [2, 4]. 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 forces. The control design problem reduces to the selection of a continuous force field distribution that will locally transport the object to a prescribed position, and then stabilize it at that configuration. To implement the control strategy on the real array, one must adapt the continuous vector field control to the real (and discrete) actuator array.

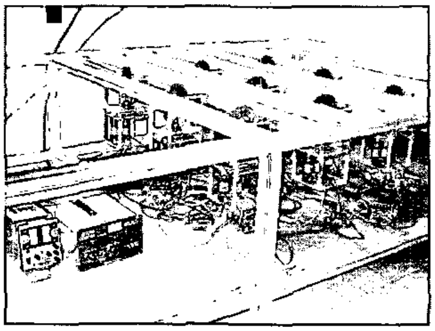

Fig. 1
THE FADM SYSTEM

The programmable vector field 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 [2, 9]. This approach is additionally well suited to distributed air jets, where the aerodynamics effectively "smooth out" the resulting forces on the object. 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 [8]. In these cases, the continuous approximation does not adequately incorporate the physics of the actual array and the object/array interface.

These experimental observations previously 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 [11]). We showed that under very simple and general assumptions on the friction model, the PFF approach leads to unstable systems when implemented on actual distributed manipulation arrays that have frictional contact [13]. The instability arises in the object's orientation at the equilibrium configuration. In Ref. [12] we presented a nonsmooth control law that locally stabilizes this instability in a provably correct way. In [14] we showed how to

0-7803-7736-2/03/$17.00 ©2003 IEEE

combine the programmable vector field approach and the local control law of [12] to obtain a globally exponentially controllable distributed manipulation control system.

In this paper, we briefly review these theories (Section II), present the design of an experimental distributed manipulation test-bed (Section III), develop explicity non-smooth control laws for this system (Section IV), and demonstrate these prior theoretical results on our prototype via detailed experiments (Section V).

II. Background

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 explicity investigate, incorporate, and control the complex frictional contact phenomena inherent in such systems, one needs to develop general modelling schemes that can capture these phenomena. One could resort to a general Lagrangian modelling approach that accounts for the contact effects through Lagrange multipliers. Instead, we sought to develop a general modelling scheme that captures the salient physical features, while also leading to equations that are amenable to control analysis.

To realize this goal, we use a "power dissipation model" (PDM) approach to model the governing dynamics of a distributed manipulation system involving a discrete number of frictional contacts. One can show that this method almost always produces unique models [11] that are relatively easy to compute, and to which one can apply 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.

We assume that the moving body and actuator elements that contact the object can be modelled as rigid bodies making point contacts that are governed by the Coulomb friction law at each contact point. Let q 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 conditions, the relative motion of each contact between the object and an actuator array element can be modeled in the form $\omega(q)\dot{q}$. If $\omega(q)\dot{q} = 0$, the contact is not slipping, while if $\omega(q)\dot{q} \neq 0$, then $\omega(q)\dot{q}$ describes the slipping velocity.

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* or *Friction Functional* for an n-contact state is defined to be

$$\mathcal{D} = \sum_{i=1}^{n} \mu_i N_i \mid \omega(q)\dot{q} \mid \qquad \text{(II.1)}$$

with μ_i and N_i being the Coulomb friction coefficient and normal force at the i^{th} contact, which are assumed known.

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 $\dot{q}$ small, 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* assumes 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 [1] on wheeled vehicles, and more details can be found in [11]. 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 [11].

When one applies the PDM method, the governing equations that result take the following form.

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

$$\dot{x} = f_{\sigma_1}(x)u_1 + f_{\sigma_2}(x)u_2 + \cdots + f_{\sigma_n}(x)u_n \qquad \text{(II.2)}$$

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 and f_i measurable in (x,t) and g_i analytic in (x,t) for all i. $\diamond$

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 [11] it was shown that the PDM generically leads to MMDA systems as in Definition II.2.

A. Equations of Motion

This section sketches the application of the power dissipation method to the example of a planar array of driven wheels. The i_{th} actuator is located at (x_i, y_i), has an orientation with respect to the origin of θ_i, and the velocity input at that actuator is u_i. I.e., the i^{th} wheel is spinning at speed u_i. Moreover, let g_i be the transformation in $SE(2)$ (the special euclidean group) from the origin to a reference frame associated with the i^{th} actuator. The relative velocity of each contact point between the wheel and moving object can be expressed as $\Omega(q)\dot{q}$, where q is the configuration of the object in $SE(2)$ and

$$\Omega_i(q) = \begin{bmatrix} Ad^T_{g_i^{-1}} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \\ 0 \end{pmatrix} u_i \\ Ad^T_{g_i^{-1}} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & 0 \end{bmatrix} \qquad \text{(II.3)}$$

and

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

is the homogeneous representation of the i^{th} actuator node's configuration relative to a fixed reference frame, $Ad(\cdot)$ is the adjoint transformation which transforms velocities from one coordinate frame to another, and $R(\cdot) \in SO(2)$ (that is, R is a member of the special orthogonal group).

We now consider, through an intuitive discussion that can be backed up by analysis, the application of the PDM to this problem.

First note that the minimum of the power dissipation function only occurs when three of the contact constraints are satisfied (i.e. $\Omega_i(q)\dot{q} = 0$ for three choices of the index i). The constraints satisfied are precisely those which would otherwise dissipate the most energy if they were violated. The contact states that dissipate the most energy are those associated with the potential constraints having the largest three normal forces $\alpha_i = N_i \mu_i$. Based on these observations, if the center of mass determines the normal forces (based on assumptions about surface uniformity, etc.), and if $\mu(x, y)$ is uniform, then the object's motion satisfies whichever constraints are closest to its center of mass. That is, the particular region in which the center of mass lies determines the first two actively satisfied constraints. The third actively satisfied constraint is determined by the friction model. The system equations are found by solving for the annihilator of the constraint $\Omega(q)$. If the coefficient of friction for sideways slipping, μ_S, is less than the coefficient of friction for rolling slipping, μ_R, and if the nearest actuator to the center of mass is indexed by i and the second nearest is indexed by j, then the governing equations are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{u_i[s_j((x_i - x_j)c_i + y_i s_i) + c_i c_j y_j] - u_j y_i}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{u_j x_i - u_i[c_i c_j x_i + s_i(x_j s_j + (y_i - y_j)c_j)]}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{u_j - u_i \cos(\theta_i - \theta_j)}{(x_i - x_j)s_j + (y_j - y_i)c_j} \end{bmatrix} \qquad \text{(II.4)}$$

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, etc. 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. This in turn leads to the state space being divided up by transition boundaries between different sets of equations of motion. To write this as an MMDA system, we may rewrite the above system as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f_1 u_i + f_2 u_j \qquad \text{(II.5)}$$

where

$$f_1 \in \begin{bmatrix} \frac{-y_i}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{x_i}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{1}{(x_i - x_j)s_j + (y_j - y_i)c_j} \end{bmatrix} \quad f_2 \in \begin{bmatrix} \frac{s_j((x_i - x_j)c_i + y_i s_i) + c_i c_j y_j}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{-c_i c_j x_i - s_i(x_j s_j - (y_i - y_j)c_j)}{(x_j - x_i)s_j + (y_i - y_j)c_j} \\ \frac{-\cos(\theta_i - \theta_j)}{(x_i - x_j)s_j + (y_j - y_i)c_j} \end{bmatrix}$$

As the 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, we must allow f_1 and f_2 to be multivalued (hence the inclusion $\in$ instead of equality in Equation (II.5)).

B. Review of Relevant Theory

This section briefly describes three previous results that form the basis of the experiments in Section V. An *elliptic vector velocity field* is one of the form $\Psi(x, y) = (-ax, -by)$ (these are common structures in the open loop theory). The first result states that although elliptic vector fields cannot stabilize (II.5), with full state feedback the equations can be stabilized (see [13]).

Theorem II.1. *The system (II.5) is unstable using an elliptic vector velocity field $\Psi(x, y) : \mathbb{R}^2 \to \mathbb{R}^2$, but is locally stabilizable through a discontinuous feedback law.*

We should comment that the instability in Theorem II.1 is only true for this subset of open loop strategies, and is not necessarily true for all open loop strategies. It nevertheless motivates our work in closed loop control, both in this paper and elsewhere. This result only guarantees local stability, however, so a more global theory must be established for purposes of implementation. In this case, it is possible to use elements of the open loop theory of programmable vector fields to do so. In particular, the major goal of [14] was to combine the programmable vector field approach with our local feedback law implied by Theorem II.1. We use a programmable vector field to govern the object's gross motions far away from the equilibrium point, and a locally stabilizing feedback law in the vicinity of the equilibrium configuration. Consider Fig. 2. The intuition behind this approach is that if we can move a package from one point $a \in \mathbb{R}^2$ to an equilibrium point $b \in \mathbb{R}^2$, and if we have feedback in a closed

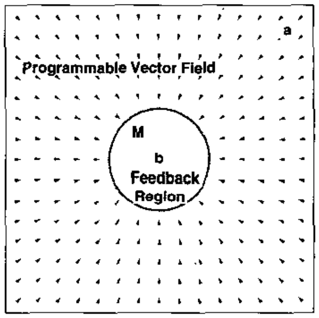

Fig. 2

A LaSalle Invariance Theorem

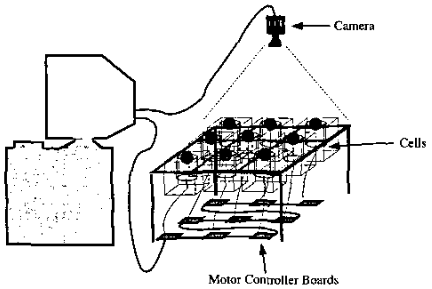

Fig. 3

CARTOON OF EXPERIMENT

neighborhood $\mathcal{M}$ of point b, we can wait to reorient the package until it has entered $\mathcal{M}$.

To effect this blending, we developed a variation of the classical LaSalle Invariance Principle which we then used as the basis for the following theorem (see [14]).

Theorem II.2. *Given a discrete planar array geometry, an elliptic vector velocity field* $\Psi(x,y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ *outside of* $\mathcal{M} = B_\epsilon \times S_1$ *(where* $B_\epsilon \in \mathbb{R}^2$ *is a ball of radius* ϵ*) for some* $\epsilon > 0$*, and a locally stabilizing feedback law (which we know exists by Theorem II.1) the solutions to the governing equations given by the PDM are globally stable.*

III. The Experimental Setup

An experimental prototype has been developed for testing the results in Section II and other proposed algorithms. A photograph of the system can be seen in Fig. 1, and a schematic of the system in Fig. 3. 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 Figure 4). Note that the orienting axis of each cell can be fixed so that the system can simulate simpler systems with a fixed driving wheel orientation. 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 [13].

These cells can be easily repositioned into different configurations in the supporting structure so as to simulate different types of systems. As seen in Fig. 1, the Fully Actuated Distributed Manipulation (FADM) system is deployed with a total of nine cells. More cells can be added as needed.

Both actuators 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 position acquistion and tracking, a rght triangle is affixed to the moving object. Feature tracking software (written in C) developed at Caltech's Computational Vision Laboratory (see [7]) 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, only six to seven iterations per second can be realized at present.

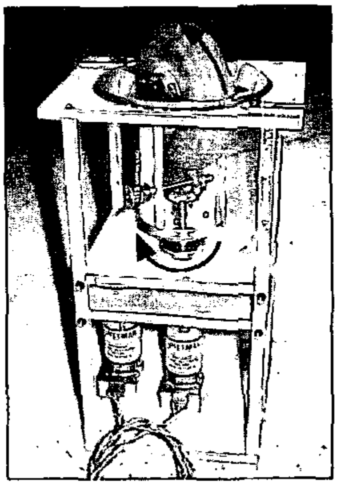

Fig. 4

PHOTOGRAPH OF CELL. ACTUATED MOTIONS ARE MARKED IN BLUE.

IV. Feedback Algorithms

This section presents in detail the algorithm used in the underactuated distributed manipulation experiment found in Section V. For a more rigorous treatment of the design of this control law, see [10]. The discontinous feedback law 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. This methodology allows the control design to be relatively simple, even for relatively complex systems.

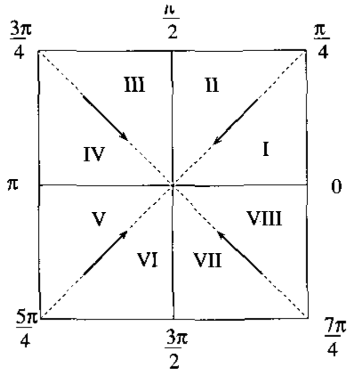

Fig. 5

ALGORITHM FOR NONSMOOTH FEEDBACK

Consider Figure 5, which might represent a portion of a distributed manipulator near a desired equilibrium point. This region has four actuators (corresponding to the inputs $u_1, \ldots, u_4$ and represented in the figure by arrows) located at $(\pm 1, \pm 1)$, all pointed towards the origin. An analysis of this system using the PDM method shows that the region can be divided into 16 distinct regions. In the eight open regions, labeled **I** – **VIII**, one contact state holds. The other 8 boundary regions, labeled $0 - 2\pi$ in increments of $\frac{\pi}{4}$ denote the boundaries between contact states. Depending on which one of these 16 regions the center of mass of the object resides in, the algorithm chooses a different appropriate control law. In each one of these regions a control law calculated from the Lyapunov function $k(x^2+y^2+\theta^2)$ by solving $\dot{V} = -1$ for u_i, where k is some constant to be chosen during implementation. These control laws can be found in Table I. The "Region" Column shows the region of applicability, and the "Control Law" section shows the explicit control law for each actuator. For regions **I-VIII**, $u_3 = u_1$ and $u_4 = u_2$. For regions $\mathbf{0} - \mathbf{7\pi/4}$ whichever control satisfies $u_i = k\theta$, the control of the same parity (i.e. even or odd index) satisfies $u_j = -k\theta$, while the other two controls are equal.

The globally stabilizing control law in Theorem II.2 is actually quite simple. For our purposes, we chose a radius around the origin of .25 meters to be the "feedback" region, $\mathcal{M}$. Outside of $\mathcal{M}$ we defined the vector field ψ to be $\psi = (-x, -y)$ and inside of $\mathcal{M}$ we used feedback to stabilize the object to the desired position and orientation.

V. Experiments

Here we summarize experimental results that illustrate the theory reviewed in the previous sections [1]. The goal of the experiment was to stabilize an object from a random initial condition to the final configuration $(x_f, y_f, \theta_f) = (0\text{ m}, 0\text{ m}, 0\text{ rad})$. For these experiments, the manipulated object is a piece of plexiglass. This choice allows us to view the actuator actions during manipulation while still ensuring reasonable amounts of friction. The plexiglass has a white piece of paper with a black triangle on it (which is identified by the tracking software) as can be seen in the movie snapshots in Figures 9 and 11. We should note that the plexiglass covers the majority of the viewable area. These snapshots also include outlines of the "goal" triangle position. The following paragraphs summarize each experimental effort and result.

A. Programmable Vector Field

This experiment uses our 9 cell experiment to implement an open loop elliptic vector velocity field. The main point to notice here is that the object did not reorient vey much despite the fact that we were trying to get it to reorient by approximately π radians. Nevertheless, the open loop method did successfully stabilize the x and y coordinates of the object's location. 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 result that follow.

Region	Control Law	
I	u_1	$\frac{-u4\,(\theta+x-y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$k\theta$
II	u_1	$\frac{u_2\,(\theta+x-y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$-k\theta$
III	u_1	$k\theta$
	u_2	$\frac{u_1\,(\theta+x+y)-k\left(\theta^2+x^2+y^2\right)}{x-y}$
IV	u_1	$-k\theta$
	u_2	$-\frac{u_3\,(\theta+x+y)+k\left(\theta^2+x^2+y^2\right)}{x-y}$
V	u_1	$\frac{u_2\,(\theta-x+y)-k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$k\theta$
VI	u_1	$-\frac{u_4\,(\theta-x+y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$-k\theta$
VII	u_1	$k\theta$
	u_2	$\frac{u_3\,(-\theta+x+y)+k\left(\theta^2+x^2+y^2\right)}{x-y}$
VIII	u_1	$-k\theta$
	u_2	$\frac{u_1\,(\theta-x-y)+k\left(\theta^2+x^2+y^2\right)}{x-y}$
0	u_1	$k\theta$
	u_2	$\frac{-u_1\,((-1+\delta)\,\theta+x+y)+k\left(\theta^2+x^2+y^2\right)}{\delta\,\theta+x-y}$
$\pi/4$	u_1	$\frac{-(-u_2+\delta(u_2+u_4))\,(\theta+x-y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$-k\theta$
$\pi/2$	u_1	$\frac{u_2\,(\delta\,\theta+x-y)+k\left(\theta^2+x^2+y^2\right)}{(1-\delta)\,\theta+x+y}$
	u_2	$-k\theta$
$3\pi/4$	u_1	$k\theta$
	u_2	$\frac{(-u_3+\delta\,(u_1+u_3))\,(\theta+x+y)-k\left(\theta^2+x^2+y^2\right)}{x-y}$
π	u_1	$k\theta$
	u_2	$-\frac{u_3\,(\delta\,\theta+x+y)+k\left(\theta^2+x^2+y^2\right)}{(-1+\delta)\,\theta+x-y}$
$5\pi/4$	u_1	$\frac{(-u_4+\delta\,(u_2+u_4))\,(\theta-x+y)-k\left(\theta^2+x^2+y^2\right)}{x+y}$
	u_2	$-k\theta$
$3\pi/2$	u_1	$-\frac{u_4\,(\delta\,\theta-x+y)+k\left(\theta^2+x^2+y^2\right)}{(-1+\delta)\,\theta+x+y}$
	u_2	$-k\theta$
$7\pi/4$	u_1	$k\theta$
	u_2	$\frac{-(-u_1+\delta(u_1+u_3))(\theta-x-y)+k\left(\theta^2+x^2+y^2\right)}{x-y}$

TABLE I

LIST OF CONTROL LAWS

B. Local Nonsmooth Feedback

This experiment duplicates the geometry of Fig. 5. The goal is to stabilize the supported object to the origin under the strict conditions that the wheel orientations are fixed. The nonsmooth feedback law used in this experiment was precisely that described in the previous section. Something to notice in this experiment is how smoothly the θ variable is driven to 0. This is somewhat surprising considering that the θ variable is precisely the unstable mode in the open loop case (see [13]). However, this is precisely why the experimental results have rather large oscillations as the trajectory is stabilized. As the object gets very close to the *origin of* $\mathbb{R}^2$, *the system becomes difficult to control*, so if the trajectory gets too close to the origin of $\mathbb{R}^2$ before the

[1] Movies of these and other experiments can be found at *http://www.cds.caltech.edu/~murphey/experiment/.*

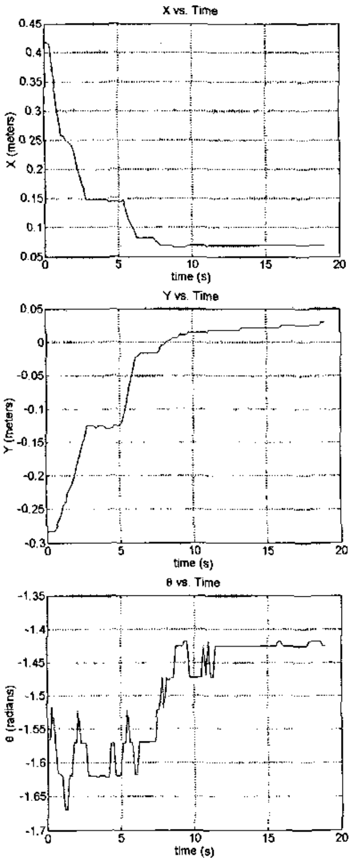

Fig. 6

OPEN LOOP CONTROL

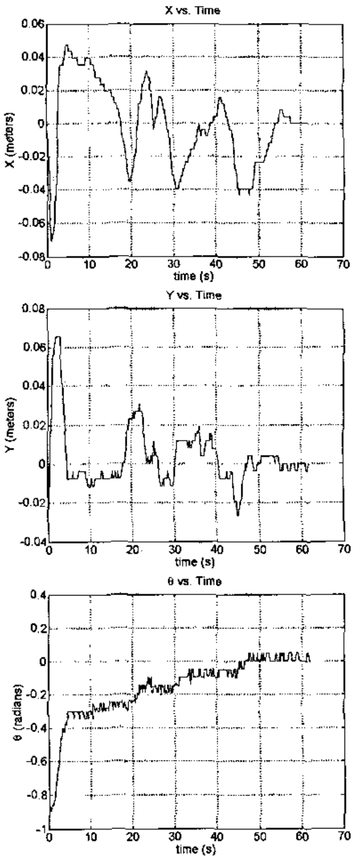

Fig. 8

UNDER-ACTUATED FEEDBACK CONTROL

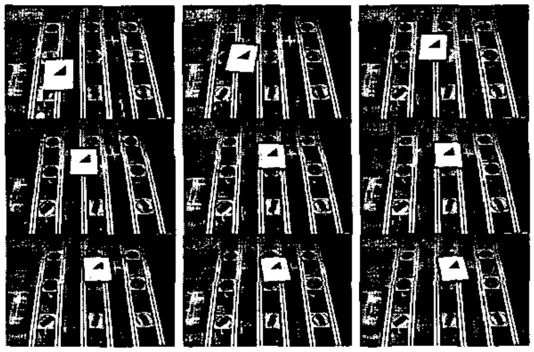

Fig. 7

OPEN LOOP MOVIE SNAPSHOTS

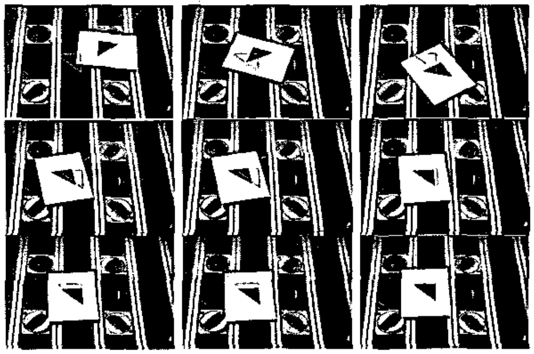

Fig. 9

UNDER-ACTUATED MOVIE SNAPSHOTS

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.

C. Global Invariance

This experiment used nine FADM actuators arranged in a regular array (see Fig. 11). We defined a 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 vector field. Inside of this region, we 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 $\mathcal{M}$. See the companion paper [15] for details of this locally stabilizing control law.

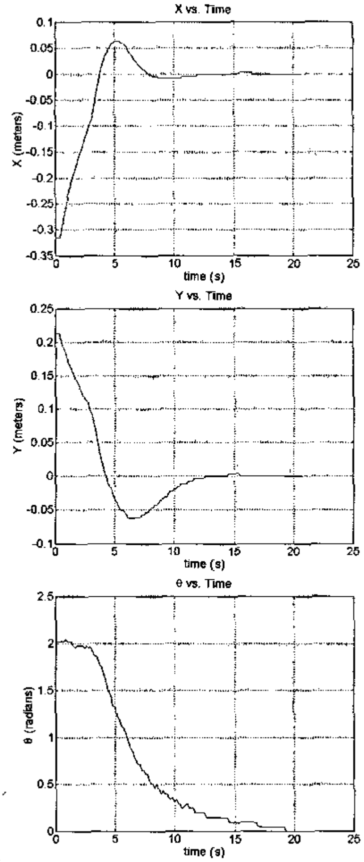

Fig. 10

COMBINING THE PROGRAMMABLE VECTOR FIELD WITH LOCAL FEEDBACK

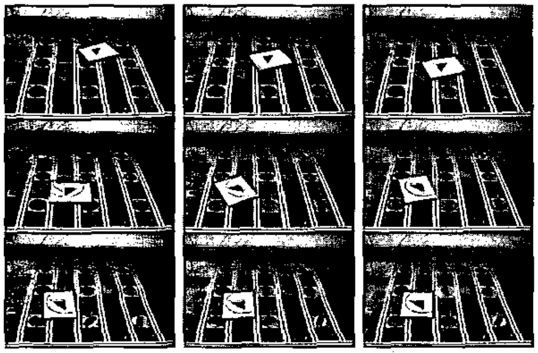

Fig. 11

GLOBAL STABILIZATION MOVIE SNAPSHOTS

VI. Summary

The experimental results presented here confirm that the modeling and control methodologies in [13, 14] are valid. This indicates several important facts. First, it means that the effects of friction and intermittent contact are important to understanding many distributed manipulation problems. Second, the control algorithms relevant to this class of systems are typically nonsmooth, but still relatively simple. Basic Lyapunov techniques adapted to the nonsmooth setting proved to be more than sufficient for designing and implementing controllers for our distributed manipulation system. 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. In the future, these methods will be adapted to account for more realistic conditions at the MEMS level, making micro assembly and other micro tasks more feasible.

Acknowledgements: We would like to thank David Choi for his help in building the experiment. We are grateful to the National Science Foundation (grant NSF9402726) for helping to support this research through its Engineering Research Center (ERC) program.

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] 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.

[3] 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.

[4] M. Coutinho and P. Will. The intelligent motion surface: a hardware/software tool for the assembly of meso-scale devices. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1997. Albuquerque, New Mexico.

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

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

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

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

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

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

[11] 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.

[12] T. D. Murphey and J. W. Burdick. Global stability for distributed systems with changing contact states. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Hawaii, 2001.

[13] 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.

[14] 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.

[15] T. D. Murphey and J. W. Burdick. Smooth feedback control algorithms for distributed manipulation. In *Submitted to IEEE Int. Conf. on Robotics and Automation*, 2003.