

Smooth Feedback Control Algorithms for Distributed Manipulators

T.D. Murphey, J.W. Burdick

Engineering and Applied Science, California Institute of Technology

Mail Code 104-44, Pasadena, CA 91125 USA

{murphey, jwb}@robotics.caltech.edu

Abstract—This paper introduces a smooth control algorithm for controlling fully actuated distributed manipulation systems that operate by frictional contact. The control law scales linearly with the number of actuators and is simple to implement. Moreover, we prove that control law has desirable robustness properties in the presence of the nonsmooth mechanics inherent in distributed manipulation systems that rely upon frictional contact. This algorithm has been implemented on an experimental distributed manipulation test-bed, whose structure is briefly reviewed. The experimental results confirm the validity and performance of the algorithm.

I. INTRODUCTION

A distributed manipulation system typically consists of a large number of similar or identical actuators arranged in a planar array, combined with a control strategy to create net movement of an object or objects placed on the array. Many distributed manipulation systems are designed to allow precise positioning of planar objects from arbitrary initial configurations. With this capability, a distributed manipulator is a “smart conveyor” that can be used for separating parts and precisely positioning them for the purpose of assembly operations. Distributed manipulation systems offer potential for micro-assembly using MEMS technology [20]. Distributed manipulator actuation methods ranges from air jets and wheels on the macroscale, to microelectromechanical systems (MEMS) and flexible cilia at the microscale.

These machines are typically relatively easy to build, but the systematic control of such devices can be quite difficult, leading to a recent increase in the study of distributed manipulation control systems [2, 8, 9, 10, 13, 19, 20]. Distributed manipulation systems are in a sense massively *over-actuated*. They can potentially have thousands of inputs (though only a subset of the actuators typically influence the moving object at any given instant), while the output consists of the states of the manipulated object. For good general references on distributed manipulation, see [4, 6, 11].

This paper addresses the issue of control system design for a class of planar distributed manipulation systems whose physical operation involves rolling and sliding contacts between the moving object and the actuator surfaces. As reviewed in Section II, the governing equations of such systems are generally non-smooth [14]. In general, one needs to use discontinuous control laws to stabilize such systems. In prior work [16, 17, 15], we introduced non-smooth control laws for stabilizing this class of distributed

manipulation systems. In this paper we show that while the governing equations can often be nonsmooth, in the special case of full actuation (whose definition is given shortly) there exists a *smooth* solution to the control design problem. Moreover, we prove that this solution is robust and stable in the presence of the non-smooth governing mechanics. We have implemented this algorithm on an experimental system that is described more fully in Section IV (and shown schematically in Figure 1. The excellent performance of the experimental implementation confirms our theoretical predictions.

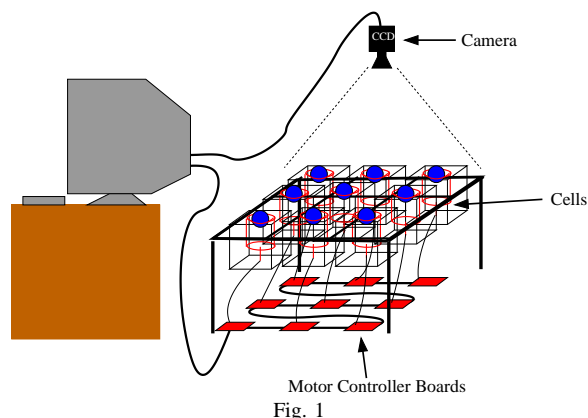


Fig. 1

A DISTRIBUTED MANIPULATION EXPERIMENT

This paper has the following structure. Section II reviews relevant background on distributed manipulation mechanics and control. Section III considers the case of “full actuation”—when all the actuators can be steered and driven. It gives an extremely simple, scale-able algorithm that is provably globally exponentially stabilizing, thus showing that it is highly desirable to have a fully actuated distributed manipulator.

Section IV summarizes experimental results obtained when this algorithm is implemented on the prototype system that is summarized in Section sec-our-experiment. A companion paper [18] describes this prototype test-bed in more detail, and summarizes experiments that validate our previously proposed non-smooth algorithms [15, 17] for the non-fully-actuated case.

II. BACKGROUND

The commonly used *programmable force field* (PFF) approach [3, 7] for distributed manipulator control is based on a continuous “force field” abstraction which

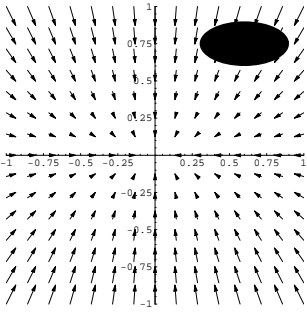


Fig. 2
PROGRAMMABLE FORCE FIELD

assumes that at each point on the manipulation surface one can specify the manipulation force at that point (see Figure II for an example of a continuous sink and an elliptical object). The dynamics of the moving object are obtained by integrating the continuous force field to get a total force on the part. To use the controls on an actual array, where the manipulation forces will be generated at discrete points, one must adapt the continuous approximation to the geometry of a given discrete array. For a good reference, see [5].

A. Nonsmooth modeling

When only a small number of actuators are in contact with the object being manipulated (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 (see Luntz et al. [12]). 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 investigate the nonsmooth properties that arise due to frictional stick/slip phenomena at the interface between the actuation surfaces and the manipulated object [16]. 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 contacts. In Ref. [15, 17] we presented local and global nonsmooth control algorithms that fix this instability problem. These controls are particularly relevant to the case where the distributed manipulation system is not fully actuated. In the case of Figure 1, this situation corresponds to not allowing the actuating wheels to be steered. The implementation of the result presented here is very simple by comparison, and moreover leads to very easy implementation, but is only appropriate to fully actuated systems (i.e., when the actuating wheels in the example of Fig. 1 are steerable).

We now give an overview of the modeling methodology we use. These results are needed as part of the proof of robustness and stability in Section III. The rationale behind this modeling technique is discussed in [14, 16]. Let q denote the configuration of the distributed manipulator system. The configuration q includes the object's position

and orientation, as well as the variables that describe the state of each actuator. The relative motion of each contact between the part 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 (i.e., that contact is a nonholonomic constraint), while if $\omega(q)\dot{q} \neq 0$, then $\omega(q)\dot{q}$ describes the slipping velocity. In general, the moving part 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 part'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

$$\mathcal{D} = \sum_{i=1}^n \mu_i N_i \left| \omega_i(q)\dot{q} \right| \quad (\text{II.1})$$

where μ_i and N_i are 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 (although our current interest will be those cases when such a condition exists), we are lead to the following concept for finding the governing equations.

Power Dissipation Principle: With \dot{q} small, a part's motion at any given instant is the one that minimizes \mathcal{D} .

The *power dissipation method* assumes that the part'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. In [14], we showed that the power dissipation approach generically leads to multiple model systems defined next.

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

$$\Sigma : \quad \dot{q} = f_1(q)u_1 + f_2(q)u_2 + \cdots + f_m(q)u_m \quad (\text{II.2})$$

where for any q and t , $f_i \in \{g_{\alpha_i} | \alpha_i \in I_i\}$, with I_i an index set, g_{α_i} analytic in (q, t) for all α_i , and the controls $u_i \in \mathbb{R}$ piecewise constant and bounded for all i . Moreover, letting σ_i denote the “switching signals” associated with f_i (which will be referred to as “MMDA maps”),

$$\begin{aligned} \sigma_i : \quad Q \times \mathbb{R} &\longrightarrow \mathbb{N} \\ (q, t) &\longrightarrow \alpha_i \end{aligned}$$

then the σ_i are measurable in (q, t) .

An MMDA system is a driftless affine nonlinear control system where each control vector fields may “switch” back and forth between different elements of a finite set. The σ_i which regulate this switching may not be known, so we have no guarantees about the nature of the switching except that it is, by assumption, measurable. In the case of distributed manipulation, this switching corresponds to the switching among different contact states (i.e., different sets

of slipping contacts) due to variations in contact geometry and surface friction properties. Practically, the switching in contact states can not be predicted in advance, and it is very difficult to measure in practice.

In the case of an n actuator array, we will have $2n$ potential constraints $\omega^i(q)$. The minimum of \mathcal{D} is precisely the choice of \dot{q} that annihilates the three constraints with the most dissipation. These, of course, will change over time because μ_i and N_i depend implicitly on the configuration, $q \in Q$. For our purposes, all we need to know is that an n actuator system will have $\binom{2n}{3}$ possible models. Note also that each of these models depends smoothly on the inputs u^k . This will be important in Theorem III.1 when we prove that the proposed control design is robust with respect to switching between models due to actuator uncertainties.

III. MAIN RESULT

This section describes in detail how a globally stabilizing *smooth* controller can be constructed. This approach requires that the distributed manipulator be fully actuated - i.e., at each actuator location the actuator can be oriented in any direction, and produce a velocity in that direction of arbitrary magnitude. The former is the important part while the latter can be made more realistic by using saturation functions.

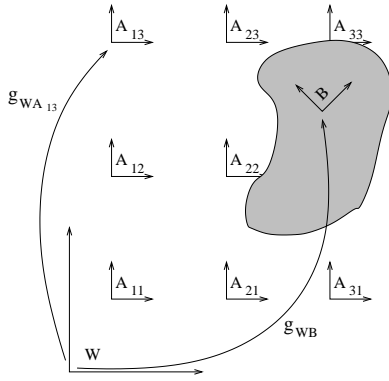


Fig. 3

RIGID BODY VELOCITIES

Consider Figure 3, which depicts an abstraction of a 9 cell experimental system (Section IV). 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” fixed at the point of contact between the actuator located at (x_i, y_j) and the object. This last frame has 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 the W to A_{ij} be denoted by $g_{WA_{ij}}$. Recall 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:

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

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

$$Ad_g = \begin{bmatrix} R(\theta) & \begin{pmatrix} y_{ab} \\ -x_{ab} \end{pmatrix} \\ 0 & 1 \end{bmatrix}.$$

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:

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

This system is trivially exponentially stable. The velocity \dot{q} is mapped to the actuators in order to obtain a feedback law.

We continue with a particular choice of Lyapunov function: $V(x, y, \theta) = k_1 x^2 + k_2 y^2 + k_3 \theta^2$ for $k_i > 0$. The Adjoint operator mapping velocities from W to the A_{ij} when the actuator frames are oriented parallel to the world frame is:

$$Ad_{g_{WA_{ij}}} = \begin{bmatrix} Id & \begin{pmatrix} y_j \\ -x_i \end{pmatrix} \\ 0 & 1 \end{bmatrix}$$

where Id is the identity. Transforming the velocity into the actuator frame yields $Ad_{g_{WA_{ij}}} \cdot (-\frac{\partial V(q)}{\partial q})$. Substituting in for $\frac{\partial V(q)}{\partial q}$, the actuator velocities should be

$$\begin{bmatrix} k_3 y_i (\theta - \theta_d) - k_1 (x - x_d) \\ -k_3 x_i (\theta - \theta_d) - k_2 (y - y_d) \\ -k_3 \theta \end{bmatrix}$$

where x_d, y_d, θ_d are the desired values and x, y, θ are the state feedback values. To transform this into wheel velocities and wheel orientations for the particular example found here, calculate the magnitude and direction of the (x, y) velocity. This gives for each actuator:

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

and

$$v_{ij} = \sqrt{\frac{(-k_3 x_i (\theta - \theta_d) - k_2 (y - y_d))^2}{(k_3 y_i (\theta - \theta_d) - k_1 (x - x_d))^2}} \quad (\text{III.2})$$

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 Equations (III.1) and (III.2) for each actuator, and the feedback law is complete.

Now we consider the robustness of this feedback law with respect to the multiple model system that arises if the actuators are not all perfectly aligned. In particular, if

we consider the controls obtained above to be the *desired* controls u_d^k and what we actually obtain are u^k , then we get slightly different dynamics. This brings us to the following theorem.

Theorem III.1. *There exists $\delta > 0$ such that if for $t > T$ we have $|u^k(t) - u_d^k(t)| < \delta \forall k$ for some T , then the solutions to the MMDA system predicted by the PDM are exponentially stable using the controls from Equations (III.1) and (III.2).*

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

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

Therefore, in a sufficiently small neighborhood of $\dot{q} \in T_q Q$ (denoted by $B(\dot{q}, \epsilon) \subset T_q Q$) we have

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

(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 $T_q Q$.)

Now, 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 satisfy the same Lyapunov function (see [17]). We will use this fact here, and 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 we know that we have a corresponding set of kinematic constraints $\omega_i(q)$, and that the PDM implies that a subset of these satisfying $\omega_i(q)\dot{q} = 0$ will define the actual kinematics. In the case of $u^k = u_d^k$, 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 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. We should also note that this can easily be extended to exponential stability in a similar fashion.

Experimental results in Section IV illustrate that this method works extremely 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, see [17] for details.

IV. EXPERIMENTAL RESULTS

An experimental apparatus has been developed for testing the results in this paper and in previous work by the authors. A photograph can be seen in Figure 4. The design is a modular one based on a basic cell design. Each

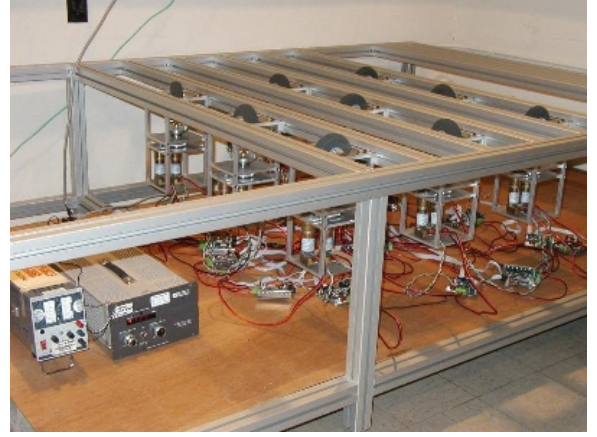


Fig. 4
FRONT OF THE FADM SYSTEM

cell contains two actuators. One actuator orients the wheel axis, while the other actuator drives the wheel rotation. These cells can be repositioned easily into different configurations. The Fully Actuated Distributed Manipulation (FADM) system shown in Figure 4 is configured with a total of nine cells—though more can be easily added. For more details on the experimental setup, please refer to the companion paper [18].

We include an illustrative experimental¹ result (found in Figure 5). The goal of the experiment was point stabilization from a random initial condition to the origin (of $SE(2)$). We put the controlled object down in a random initial configuration (in this case $(x_0, y_0, \theta_0) = (0.4 \text{ m}, -0.4 \text{ m}, 2.6 \text{ rad})$) and initiated the experiment with the actuators all in the same initial conditions of $\theta = 0$ (relative to the world coordinate axes). The goal then was to stabilize the object to a final position of $(x_f, y_f, \theta_f) = (0 \text{ m}, 0 \text{ m}, 0 \text{ rad})$. The actually achieved final position was $(x_f, y_f, \theta_f) = (0.01 \text{ m}, 0.01 \text{ m}, 0.05 \text{ rad})$. The figure panels depict the x , y , and θ trajectories as functions of time and a plot of the (x, y) trajectory in the plane for a rectangular plexiglass object.

Notice that the translational stability of the origin is maintained, while the rotational dynamics are stabilized to $\theta = 0$ due to the feedback law. The important point to notice is the smoothness of the trajectory. This experiment indicates that when a distributed array is fully actuated, the feedback law in Equation (III.2) 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 III.1.

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

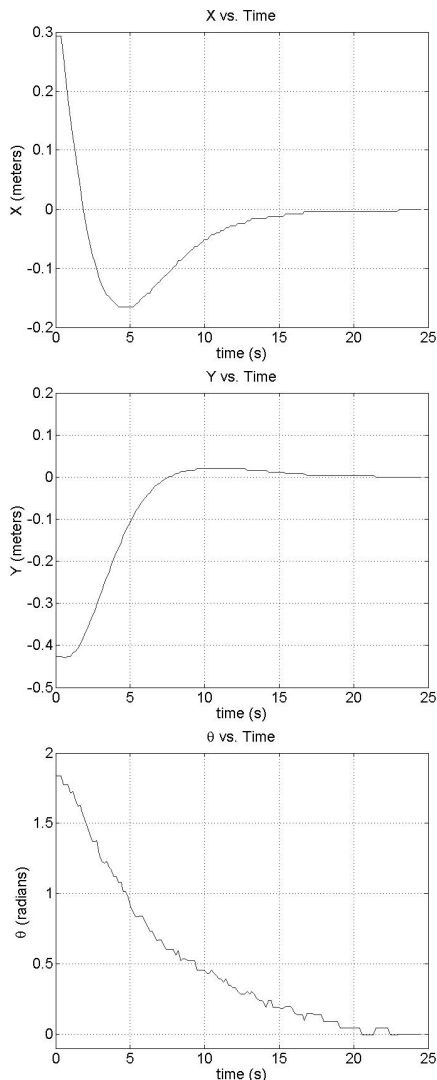


Fig. 5

GRAPHS OF THE x , y , AND θ DYNAMICS OF THE EXPERIMENTAL RESULTS

V. SUMMARY

The very simple algorithm presented in this paper provides a different closed-loop approach to the control of distributed manipulation systems. Prior work, particularly the programmable vector field approach, has assumed that all the distributed system's actuators move sufficiently fast that they slip all of the time. This causes significant stress on both the object being manipulated and the actuators themselves. In the case of fully actuated systems, it is possible to ensure that *all* of the relevant actuators contact the moving body without slip. This requires less energy for a given motion and moreover induces smaller forces on the object and actuators. This approach also raises many additional questions. For instance, for a given problem with nonsmooth mechanics, what is an analytical test to guarantee that a smooth solution (like the one presented here) exists? Is such a solution unique? Answers to these questions are part of ongoing research.

Acknowledgements: This work was partially supported by a grant from the National Science Foundation (grant NSF9402726) 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. Bhattacharya and R.D. James. A theory of thin films of martensitic materials with applications to microactuators. *J. Mech. Phys. Solids*, 47:531–576, 1999.
- [3] 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.
- [4] K.F. Böhringer and H. Choset, editors. *Distributed Manipulation*. Kluwer, 2000.
- [5] 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.
- [6] D. Rus B.R. Donald, J. Jennings. *Algorithmic Foundations of Robotics (WAFR)*, chapter Information invariants for distributed manipulation, pages 431–459. A.K. Peters, Ltd, Wellesley, MA, 1995.
- [7] 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.
- [8] H. Fujita. Group work of microactuators. In *International Advanced Robot Program Workshop on Micromachine Technologies and Systems*, pages 24–31, Tokyo, Japan, 1993.
- [9] P. Krutlevitch, A.P. Lee, P.B. Ramsey, J. Trevino J. Hamilton, and M.A. Northrup. Thin film shape memory alloy microactuators. *Journal of Micro Electro Mechanical Systems*, 5(270), 1996.
- [10] A.P. Lee, C.F. McConaghy, P. Krutlevitch, G.E. Sommargren, and J. Trevino. Electrostatic comb drive for vertical actuation. In *Proceedings of Micromachined Devices and Components, SPIE 1997 Symposium on Micromachining and Microfabrication*, volume 109, Austin, Texas, 1997.
- [11] J. Luntz, W. Messner, and H. Choset. Velocity field design for parcel manipulation on the modular distributed manipulation system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.
- [12] J. Luntz, W. Messner, and H. Choset. *Distributed Manipulation*, chapter Discreteness Issues in Actuator Arrays. Kluwer Academic Publishers, 2000.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] T. D. Murphey and J. W. Burdick. Experiments in nonsmooth control of distributed manipulation. In *Submitted to IEEE Int. Conf. on Robotics and Automation*, 2003.
- [19] K.P. Seward, P. Krutlevitch, H.D. Ackler, and P. Ramsey. A new mechanical characterization method for microactuators applied to shape memory films. In *10th International Conference on Solid State Sensors and Actuators, Transducers '99*, Sendai, Japan, 1999.
- [20] 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. *Journal of Microelectromechanical Systems*, 8(4):483–496, December 1999.