On Observing Contact States in Overconstrained Manipulation

Todd D. Murphey

Mechanical Engineering
Northwestern University
Evanston, IL 60208

Abstract. Estimation of contact state is important to any multi-point interaction that involves frictional stick/slip phenomena. In particular, when there are more kinematic constraints than there are degrees of freedom, some contact interfaces must slip, leading to the need for contact state estimation. Fortunately, supervisory control techniques from adaptive control can be applied to this problem with relatively little modification. We discuss this approach in terms of a distributed manipulation experiment developed to explore overconstrained manipulation. In this context, we show in a simulated model that on-line contact state estimation dramatically improves performance over methods that estimate contact states off-line.

1 Introduction

A manipulation system consisting of many points of contact typically exhibits stick/slip phenomenon due to the point contacts moving in kinematically incompatible manners. We call this manner of manipulation *overconstrained manipulation* because not all of the constraints can be satisfied. Naturally, uncertainty due to overconstraint can sometimes be mitigated by having backdrivable actuators, soft contacts, and by other mechanical means, but these approaches avoid the difficulties associated with stick/slip phenomenon at the expense of losing information about the state of the mechanism. This, in turn, leads either to degraded performance or to requiring additional sensors. This paper is concerned with systems that have multiple points of contact, all of which are frictional and adequately described by either constraint forces (when there is no slipping at the point contact) or by the slipping reaction force. Prototypes of this situation include distributed manipulation systems, such as those found in [1,2], as discussed in Section 2.

An important question in these systems is that of contact state estimation [3]. That is, for n kinematic constraints associated with contact interfaces, estimating at any given time which constraints are satisfied (the contact is "sticking") and which are not (the contact is "slipping"). How can one determine the stick/slip state for each point contact? Without a sensor at each point contact, the output signal must be used in some way to determine this. Moreover, the computational complexity of the solution must be considered as well, since for an n contact system there are 2^n possible stick/slip combinations. The main contribution of this paper is to show that methods from adaptive control called supervisory control [4–6] provide a reasonable solution to this problem, although the computational complexity of the solution indicates that better algorithms are needed for n large, such as in the case of MEMS manipulation. There are some reasonably simple adaptations of this technique to increase its efficiency. For instance, in [7] it was shown that many contact systems of interest can be formally reduced to kinematic systems despite stick/slip phenomena. In this case, the number of possible stick/slip states reduces to n rather than 2^n, making the adaptive control technique used here more viable. However, these modifications are largely superficial, and will need to be improved upon. Nevertheless, for n reasonable, the technique presented here works quite well in simulation, as discussed in Sections 3 and 4.

This paper is organized as follows. Section 2 discusses distributed manipulation in more detail, and discusses the experimental implementation used before in [1]. Section 3 describes the algorithm developed in [1] and gives an example simulation for this experimental system when the contact states are assumed to be known perfectly. We then illustrate how variations in the contact state can, not surprisingly, degrade the performance of the algorithm. Section 4 gives the necessary background for understanding a supervisory control system, such as that described in [4,5]. We do not provide any of the technical proofs ensuring that these methods can be applied to our systems of interest, and refer the reader to [8]. Instead, we focus on what the main result means for multi-point contact problems, both in terms of advantages and shortcomings. We also discuss the implementation of this adaptive control method to the distributed manipulation example and show in simulation that the original algorithm performance is indeed recovered even when the contact states are not known a priori.

2 Motivation–Distributed Manipulation

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 macro-scale, to MEMS and flexible cilia at the micro-scale.

Methods to design distributed manipulation control systems have been proposed in several works, including [9–14]. However, in cases where only a small number of actuators are in contact with the manipulated object or the coefficient of friction μ is very high, continuous approximations of these systems have been shown experimentally not to work well [1,2]. In these cases, the physics of the actual array and the object/array interface must be incorporated into the control design process. In particular, the discontinuous nature of the equations of motion must be addressed.

The work in [1] describes 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. The control question is twofold in its theoretical interest. First, unlike many other control problems currently being studied, distributed manipulation problems are typified by being massively overactuated. A planar distributed manipulation problem will typically only have three outputs (x, y, θ), but it may potentially have thousands of inputs. Therefore, control schemes must scale with the number of actuators in order to be able to implement them on real devices such as MEMS arrays. Second, there is the question of physical modeling. Partly because of the aforementioned overactuation, nonsmooth effects become commonplace in distributed manipulation due to intermittent contact, friction, and kinematically incompatible constraints. When these are the dominant concerns, they must be incorporated into the modeling and therefore into the control design as well. Control laws appropriate to these systems have been successfully designed, as discussed in Section 3.

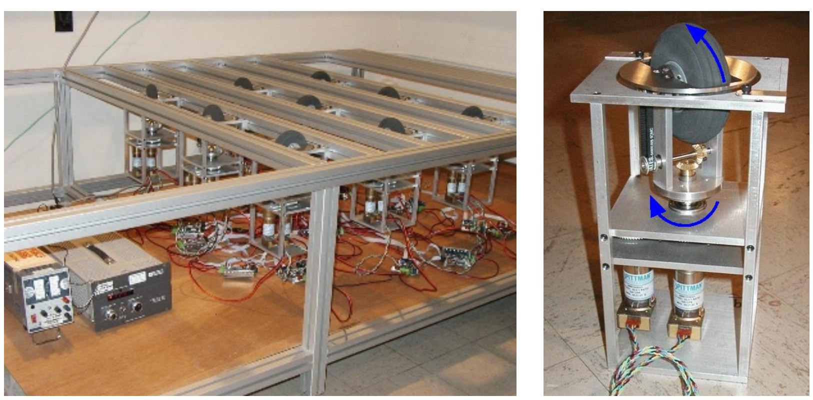

Fig. 1. The Caltech Distributed Manipulation System. (a) Front View (b) Module

A photograph of the apparatus can be seen in Figure 1. The design is a modular one based on a basic cell design. Each cell contains two actuators. One actuator orients the wheel axis, while the other actuator drives the wheel rotation (see Figure 1(b)). These cells can easily be repositioned within the supporting structure to form different configurations. The system shown in Figure 1(a) is configured with a total of nine cells—though more can be easily added. The position and orientation of the manipulated object is obtained and tracked visually. To enable visual tracking, a right triangle is affixed to the moving object. For more details on the experimental setup, please refer to [1].

When experiments were performed using this device in [1], the contact state was estimated in an open loop manner. That is, based on physical principles (described momentarily), a static condition was chosen under which the contact states would change. Somewhat surprisingly, this worked in the implementation, but only because the device was in a controlled environment and it was well characterized. In most cases this will not be the case, so we necessarily must address the problem of contact state estimation. Even in a nine cell system, this is a nontrivial task. Each wheel has two constraints (a rolling constraint and a no-sideways-slip constraint) leading to a grand total of $2^{18} \sim 10^5$ possible contact states. Although not formally addressed here, there exist techniques to reduce the number of possible states, including kinematic reduction[7] and coordination[15].

3 Modeling and Analysis

To explicitly investigate, incorporate, and control the complex frictional contact phenomena inherent in overconstrained manipulation, one needs to develop general modeling schemes that can capture these phenomena without being intractable from a control perspective. One could resort to a general Lagrangian modeling approach that accounts for the contact effects through Lagrange multipliers. Instead, we sought to develop a general modeling 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 Method" (PDM) approach to model the governing dynamics of an overconstrained mechanical system involving a discrete number of frictional contacts. One can show that this method almost always produces unique models [7,16] that are relatively easy to compute, are formally related to the Lagrangian mechanics, and to which one can apply control system analysis methods. This method produces first-order governing equations, instead of second order equations that are associated with Lagrange's equations.

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. 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 written 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 1. The *Dissipation* or *Friction Functional* for an n-contact state is defined to be

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

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

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 $\mathcal{D}$ due to contact slippage. This method is adapted from the work of [17] 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 [7,8].

When one applies the PDM method, the governing equations that result take the form of a multiple model system.

Definition 2. A control system Σ evolving on a smooth n-dimensional manifold, Q, is said to be a *multiple model driftless affine system (MMDA)* if it can be expressed in the form

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

where $q \in Q$. For any q and t, the vector field f_i assumes a value in a finite set of vector fields: $f_i \in \{g_{\alpha_i} | \alpha_i \in I_i\}$, with I_i an index set. The vector fields g_{α_i} are assumed to be analytic in (q, t) for all α_i, and the controls $u_i \in \mathbb{R}$ are piecewise constant and bounded for all i. Moreover, letting σ_i denote the "switching signals" associated with f_i

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

the σ_i are measurable in (q, t).

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,

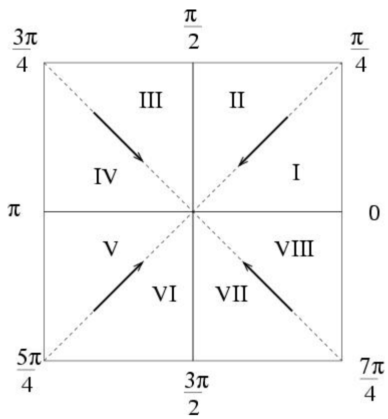

Fig. 2. A distributed manipulator with four actuators

surface friction properties, and normal loading. In [7] it was shown that the PDM generically leads to MMDA systems as in Definition 2 and is formally equivalent to a kinematic reduction of the Lagrangian formulation of the equations of motion.

The work in [1] showed that the PDM implies that the governing equations for a distributed manipulation system are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f_1 u_1 + f_2 u_2 \tag{3}$$

where

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

where $c_i = \cos(\theta_i)$, $s_i = \sin(\theta_i)$, etc. The input u_1 is the input to the closest actuator to the center of mass, and the input u_2 is the input to the second closest actuator to the center of mass. 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.

Consider Figure 2, which might represent a portion of a distributed manipulator near a desired equilibrium point. This region has four actuators

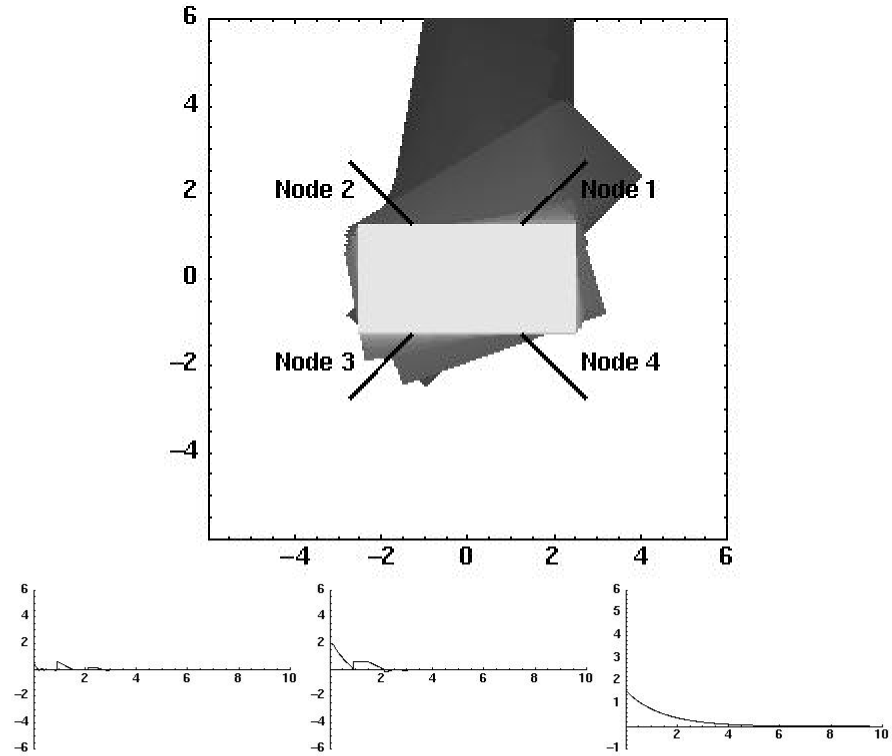

Fig. 3. Simulation of distributed manipulation when the contact state is known perfectly. The rectangle represents the center of the object which is actually in contact with all four of the actuators (Nodes 1-4). The time history progresses from dark triangles at time 0 to the light triangle at time 10. The bottom three plots are plots of the X,Y, and θ coordinates against time.

(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 8 distinct regions, labeled $\mathbf{I} - \mathbf{VIII}$, where one contact state holds. These are separated by 8 boundaries, labeled $0 - 2\pi$ in increments of $\frac{\pi}{4}$. 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 some constant to be chosen during implementation. Therefore, there are eight control laws, each defined in a separate octant. These control laws can be found in [1].

If these estimated boundaries $0 - 2\pi$ are accurate, then the control laws perform quite well. Figure 3 shows a simulation of the four actuator system. The object is indicated by a rectangle, but the reader should note that although the rectangle is illustrated as being small, the actual body it represents is in contact with all four actuators at all times, which are denoted

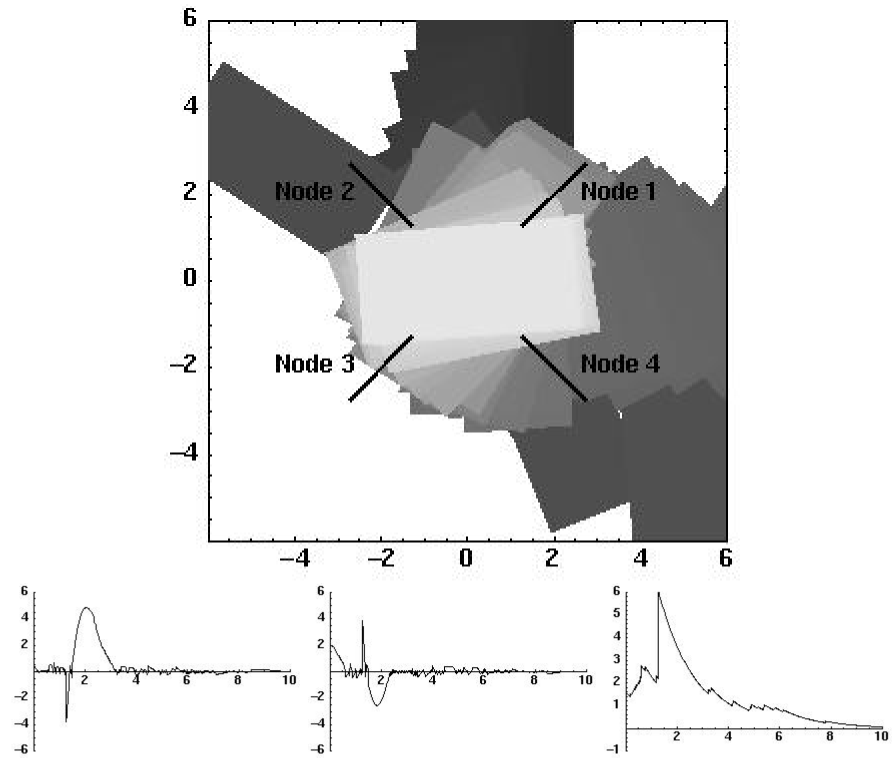

Fig. 4. Simulation of distributed manipulation when the contact state is estimated in some open loop manner but is incorrect. The object is only barely stabilized to the origin due to the contact state being varying from the nominal value. (As in Fig. 3, the bottom three plots are plots of the X,Y, and θ coordinates against time.)

in the figure by Nodes 1-4. The initial condition is $\{x_0, y_0, \theta_0\} = \{.5, 2, \frac{\pi}{2}\}$, and progress in time is denoted by the lightening of the object. The three plots beneath the XY plot are X, Y, and θ versus time, respectively. This, and the other simulations, were all done in *Mathematica*, using Euler integration in order to avoid numerical singularities when crossing contact state boundaries. In Fig. 3, the object is stabilized to $(0, 0, 0)$ with no difficulty.

In the simulations the constraints are enforced separately from the control law, allowing the control to switch at different times from the constraints. In particular, if the boundary that determines the physical contact state is allowed to vary while the control laws only change at the estimated boundaries $0 - 2\pi$, then the performance degrades substantially. Starting the object at an initial condition of $\{x_0, y_0, \theta_0\} = \{.5, 2, \frac{\pi}{2}\}$, Fig. 4 shows this degradation in comparison to Fig. 3, although the system is still stable. In the case of Fig. 4, the controller is assuming that the contact state changes when the center of mass of the object crosses the line $x = 0$, whereas the contact state is actually changing when the line $x = -0.3y$ is crossed. This is precisely the difficulty fixed by estimating the contact state on-line, as shown in Section 4.

4 Hybrid Observability and Supervisory Control

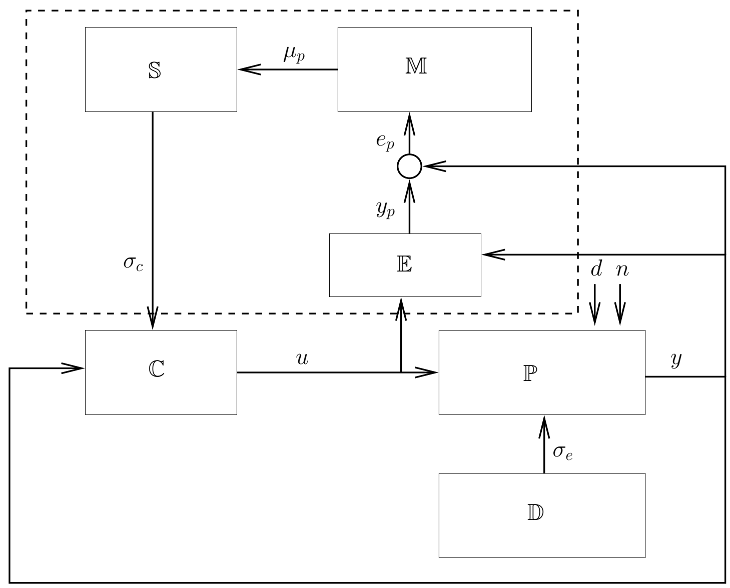

Fig. 5. A supervisory control system

Efforts in the adaptive control community have already created a framework appropriate to addressing the problem of estimating contact state. In particular, *supervisory control* (as in [4,5] and elsewhere) is an effective technique to use when a system is a linear multiple model system. Fortunately, our system, when reduced to a kinematic system using the power dissipation method, is a first order system with constant vector fields. (In fact, not only is it linear, it does not even have drift.) Hence, it is a particularly trivial multiple model system. With very little modification (which is not explored in-depth here but covered in [8]), this supervisory framework easily answers how to estimate the current contact state based on the output of the system. Moreover, it would be quite straight forward to adapt this to a second order system in a case where the kinematic reduction used in Section 3 (using the power dissipation method) is not possible.

The basic idea in supervisory control is that if there is a family (finite or possibly countably infinite) of plants P_σ indexed by σ representing the dynamics, then one can choose controllers appropriate to each P_σ and orchestrate a "switching" between these controllers such that the resulting system is stable. Traditionally, this is a technique where σ is constant but unknown, but it has been shown that given certain conditions on the type of environmental "disturbance" switching allowed to occur in an MMDA system, many of the results in [4,5] still apply. In particular, a result in [8] shows that for a family of plants P_σ and associated controllers $\mathcal{C}_\sigma$ the multiple model system is stable even if the environment causes the system to switch between plants, provided it does so sufficiently slowly on the average, which will be discussed shortly.

Consider the block diagram representation of a supervisory control system found in Fig. 5. Denote the set of possible admissible plants by $\mathbb{P}$. Each model in $\mathbb{P}$ represents a contact state of the overconstrained system. Assume that associated with each plant P_σ coming from $\mathbb{P}$ there is a known stabilizing controller C_σ. Denote the set of these controllers by $\mathbb{C}$. To determine which model in $\mathbb{P}$ most closely "matches" the *actual* model, the input-output relationships for all the plants in $\mathbb{P}$ will need to be estimated. Hence, the need for the *estimator*, denoted by $\mathbb{E}$, which will generate errors between the predicted output for each plant and the actual output of the multiple model system. These errors will then be fed into the *monitoring signal generator*, denoted by $\mathbb{M}$, which will provide monotone increasing signals μ_p. The monitoring signal will be fed into the *switching logic*, denoted by $\mathbb{S}$, which will then determine by means of a *switching signal*, σ_c, which controller to use to control the system output. Call the triple $(\mathbb{S}, \mathbb{M}, \mathbb{E})$ the *supervisor*. Details on the specifics of the supervisor can be found in [8]. Although the choice of supervisor made here (for the purposes of the simulations) is a *scale-independent hierarchical supervisor*[4,5], there are many variations in the literature on this idea. Additionally, there is an environmental signal generator $\mathbb{D}$ creating σ_e. $\mathbb{D}$ represents the externally driven switches in contact state that we would like to estimate. Lastly, denote by $N_{\sigma_e}(t_0, t)$ the number of switches σ_e experiences during time $[t_0, t)$.

For our purposes it is sufficient to note that the supervisory control system, as described, is stable so long as 1) each controller C_σ stabilizes its associated plant P_σ, 2) the estimator tracks the contact state well, and 3) the supervisor switches fast enough to ensure convergence without switching so fast as to induce instability. This last requirement is formalized in the following assumption.

Assumption 1 *Assume σ_e switching is "slow on the average," i.e.,*

$$N_{\sigma_e}(t, \tau) \leq N_0 + \frac{t - \tau}{\tau_{AD}}$$

where $N_0 > 0$ is called the "chatter bound" and τ_{AD} is called the "average dwell time."

With Assumption 1 in place, it is possible to prove the following proposition.

Proposition 1. *For the distributed manipulation system described in Section 3, for any contact state boundaries and for any average dwell time τ_{AD} there exists a supervisor such that the resulting system is exponentially stable to the origin.*

Proposition 1 indicates that if the contact states change slowly enough and feedback is fast enough, then the system can be controlled by estimating the contact state on-line. This means that one does not have to concern oneself with the friction model to establish where switching occurs, as was done previously in Section 3 in order to derive a control law. Instead, the contact states can change arbitrarily, so long as they do so sufficiently slowly on the average. It would be useful to know if a combination of physical geometry and controller choice can guarantee that this property holds, but for now we leave it as a standing assumption.

Now apply this supervisory approach to the four actuator array from Section 3. Replace the boundary $x = 0$ with the boundary $x = -0.3y$, and allow the estimator $\mathbb{E}$ to estimate the contact state and the supervisor $\mathbb{S}$ to orchestrate the controller. In this case (found in Fig. 6) the performance is considerably better than that found in Fig. 4 and resembles the performance found in Fig. 3. However, there are several important characteristics missing from this simulation. First, there is no noise in the output of the system, and it would be useful to know what the sensitivity of this nonsmooth system is to such output noise. Secondly, there is no time delay, which will almost certainly play a substantial role in the dynamics near the origin.

5 Conclusions

In this paper we have introduced the problem of estimating contact states for overconstrained systems and have offered a solution that is based on adaptive control techniques developed in [4,5] and elaborated on in [8]. The problem of contact state estimation is clearly important for systems in which stick/slip phenomena play a dominant role. Indeed, for the distributed manipulation experiment described here, manipulation tasks are actually impossible without the constant trade-off between sticking and slipping. Ultimately, the analytical techniques presented here should be extended to the more geometric setting of grasping and manipulation in the presence of gravitational forces. In the meantime, these results will be implemented on a version of the experiment discussed in Section 2.

Despite the validity of the estimation techniques presented here, more work must be done to make these methods more computationally efficient. In the supervisory control approach, every model must be integrated forward in time. In the case of the four wheel manipulator there are 8 constraints,

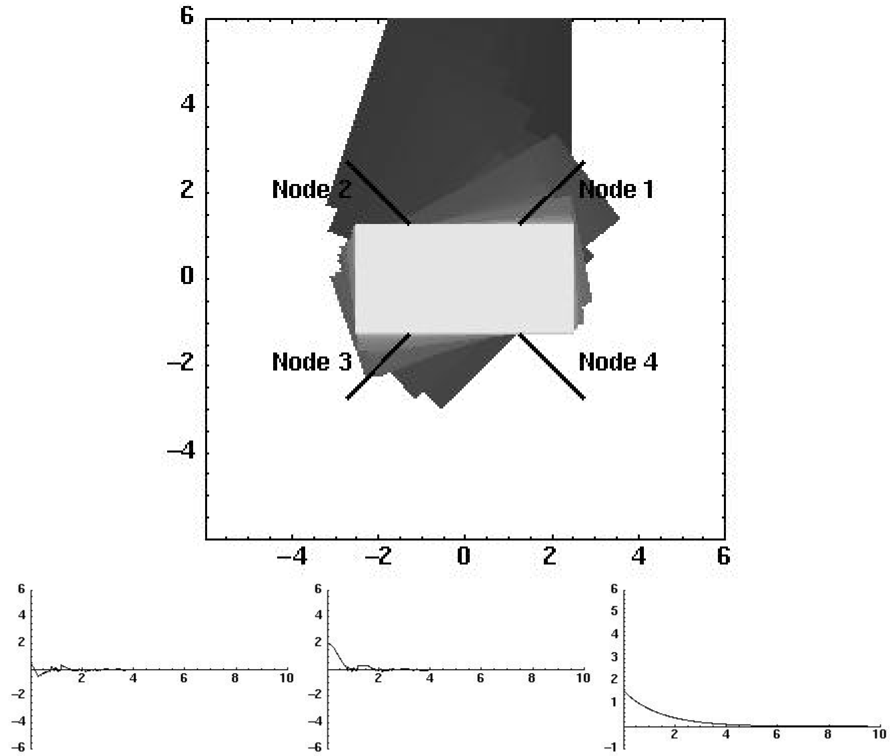

Fig. 6. Simulation of distributed manipulation when the contact state is estimated on-line, using the supervisory control methodology. Here the performance is much closer to that seen in Fig. 3, the case where our knowledge of the state is perfect. (As in Fig. 3, the bottom three plots are plots of the X,Y, and θ coordinates against time.)

leading to 2^8 possible dynamic equations of motion. Utilizing the kinematic reduction found in [7], these can be reduced to 8 total states, a tractable number for the supervisor. However, if the number of actuators is large, or if the system does not satisfy the conditions to be kinematically reducible (as is likely the case in most grasping problems), then the traditional supervisory approach will surely fail. A potential solution to this is to serially test models rather than testing them in parallel, provided they satisfy some sort of a priori known hierarchy. This method will require the additional characteristic of scaling the gains on the controller down whenever the current model chosen by the supervisor is not predicting the output well. Proving stability of such techniques is non-trivial and is the focus of ongoing study.

References

1. T. D. Murphey, J. W. Burdick, Feedback control for distributed manipulation with changing contacts, Accepted for publication in the International Journal of Robotics Research .
2. J. Luntz, W. Messner, H. Choset, Distributed manipulation using discrete actuator arrays, Int. J. Robotics Research 20 (7) (2001) 553–583.
3. R. H. T. Debus, P. Dupont, Contact state estimation using multiple model estimation and hidden markov models, in: The Eighth Int. Symp. on Experimental Robotics, Ischia, Italy, 2002.
4. B. Anderson, T. Brinsmead, F. D. Bruyne, J. Hespanha, D. Liberzon, A. 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) (2000) 909–929.
5. J. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. Brinsmead, F. D. Bruyne, Multiple model adaptive control, part 2: Switching, Int. J. of Robust and Nonlinear Control Special Issue on Hybrid Systems in Control 11 (5) (2001) 479–496.
6. J. Hespanha, A. Morse, Stability of switched systems with average dwell-time, Tech. rep., EE-Systems, University of Southern California (1999).
7. T. D. Murphey, J. W. Burdick, The power dissipation method and kinematic reducibility of multiple model robotic systems, Transactions on Robotics and Automation Submitted.
8. T. D. Murphey, Control of multiple model systems, Ph.D. thesis, California Institute of Technology (May 2002).
9. K. F. Böhringer, B. R. Donald, L. E. Kavraki, F. Lamiraux, A distributed, universal device for planar parts feeding: unique part orientation in programmable force fields, in: Distributed Manipulation, Kluwer, 2000, pp. 1–28.
10. M. Erdmann, M. Mason, An exploration of sensorless manipulation, IEEE Journal of Robotics and Automation 4 (4).
11. K. Goldberg, Orienting polygonal parts without sensing, Algorithmica 143 (2/3/4) (1993) 201–225.
12. K. F. Böhringer, R. G. Brown, B. R. Donald, J. S. Jennings, D. Rus, Sensorless manipulation using transverse vibrations of a plate, in: Proc. IEEE Int. Conf. on Robotics and Automation, Nagoya, Japan, 1995, pp. 1989–1996.
13. M. Coutinho, 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.
14. H. F. M. Ataka, A. Omodaka, A biomimetic micro motion system, in: Transducers - Digest International Conference on Solid State Sensors and Actuators, 1993, pp. 38–41, pacifico, Yokohama, Japan.
15. T. D. Murphey, J. W. Burdick, Nonsmooth controllability and an example, in: Proc. IEEE Conf. on Decision and Control (CDC), Washington D.C., 2002.
16. T. D. Murphey, 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.
17. J. C. Alexander, J. H. Maddocks, On the kinematics of wheeled vehicles, The International Journal of Robotics Research 8 (5) (1989) 15–27.