Mechanical Manipulation Using Reduced Models of Uncertainty

Todd D. Murphey

Department of Electrical and Computer Engineering, University of Colorado at Boulder, murphey@colorado.edu

Abstract: In this paper we describe methods applicable to the modeling and control of mechanical manipulation problems, including those that experience uncertain stick/slip phenomena. Manipulation in unstructured environments often includes uncertainty arising from various environmental factors and intrinsic modeling uncertainty. This reality leads to the need for algorithms that are not sensitive to uncertainty, or at least not sensitive to the uncertainty we can neither model nor estimate. The particular contribution of this work is to point out that the use of an abstraction, in this case a kinematic reduction, not only reduces the computational complexity but additionally simplifies the representation of uncertainty in a system. Moreover, this simplified representation may be directly used in a stabilizing control law. The end result of this is two-fold. First, modeling for purposes of control is made more straight-forward by getting rid of some dependencies on low-level mechanics (in particular, the details of friction modeling). Second, the online estimation of the relevant uncertain variables is much more elegant and easily implementable than the online estimation of the full model and its associated uncertainties.

1 Introduction

It is traditional in robotics to view problems of manipulation, motion planning, and control in one of two extreme lights. First, if a system is kinematic (a word which for now we leave not specifically defined), we simplify the system description from a second-order system with forces and inertias to a first-order system that consists of velocities and constraints. Then motion plans and control laws (if necessary) are designed for this kinematic system. It is important to note that in order to *implement* this design based on kinematics, a backstepping algorithm is employed, either explicitly in an “inner-loop-outer-loop” control architecture, or implicitly by purchasing motor controllers (or other appropriate devices) that provide the inner loop control. In the end, the advantages of using kinematic structures include both lessened computational burden (due to the computation in a lower-dimensional space) and increased robustness to some classes of uncertainty (due to robustness properties of the backstepping, inner-loop controller).

If, however, there is some reason that a kinematic analysis is inappropriate, then we often revert to a more complex set of modeling choices. In particular, in multi-point contact many phenomena are introduced, including soft-contact models [2], elaborate models of frictional interfaces [17], and the inclusion of dynamic effects such as inertial terms and generalized forces. Nevertheless, it is not clear that the introduction of these additional modeling techniques helps for the purpose of control, motion planning, etcetera. In fact, it is often the case that this *hurts* our ability to successfully design control strategies. Not only does the introduction of these effects make problems computationally more complex, it also decreases robustness by introducing assumptions that are often not satisfied by the environment or, worse, may only sometimes be satisfied by the environment. Hence, we can be faced with a situation where our modeling assumptions are occasionally correct, but not reliably so.

From a design perspective (as opposed to a simulation perspective), it is thus desirable to, if necessary, introduce elements to a model that provide the full complexity of possible behavior of the system without introducing too much new information (thereby decreasing the applicability of the model). This is related to the idea of abstraction, which originated in the computer science community [10] and was then made formal in a control context in [18] and related works. In this paper, we focus on a particular type of abstraction that formalizes the idea of a system being kinematic, appropriately termed *kinematic reducibility*. However, this is merely the setting for the present work. Our main focus is to discuss what types of representation of uncertainty should be used in the abstracted setting.

Consider the conceptual block diagram in Fig.1. The blocks on the right-hand-side are familiar–these four blocks represent a traditional backstepping algorithm using "virtual inputs." In the case of a kinematic vehicle, we abstract the true dynamics of the vehicle to a kinematic representation where the abstracted inputs are now velocities and the input vector fields are vectors that satisfy the kinematic constraints. Then a backstepper is used that takes these velocities as reference signals for a lower-level controller. *It is useful to point out that doing so assumes that this low-level controller is robust to any uncertainties (coming from terrain, parametric uncertainty, etcetera).* In the context of Fig.1, this means that the reduction to the kinematic system induces a formal reduction of the uncertainty. In this case, there is no representation of uncertainty whatsoever in the abstract description of the system–all the robustness is built into the backstepping algorithm. What we will see is that this same abstraction in multi-point contact systems again reduces the representation of uncertainty, but not to the point that there is no uncertainty at all in the reduced equations. Instead, there is an *abstracted* uncertainty which corresponds to the hybrid, discrete-valued state that represents the contact state (whether any given contact is in contact or out of contact and, if in contact, whether it is slipping or sticking) of the system. It is

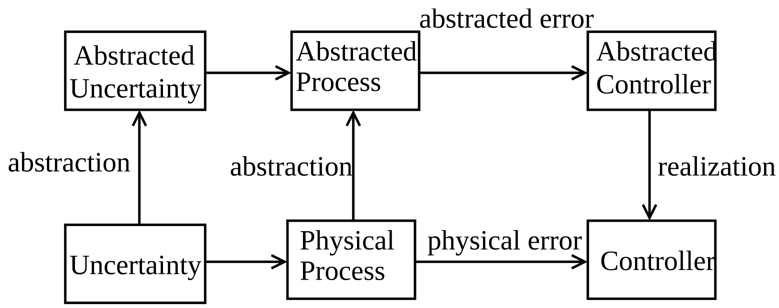

Fig. 1. Conceptual block diagram for reduction of both the physical model and uncertainty

this reduced representation of uncertainty we will use in designing controllers for these systems. The practical advantage of this approach is that the reduced representation of uncertainty makes some aspects of analysis more simple and that it can make estimation less costly both in terms of computation and bandwidth requirements.

Our previous work [15] (first presented at WAFR 2002 [14]) in this area showed that some manipulation surfaces cannot stabilize an object without feedback, and then showed that by using the Power Dissipation Method (PDM) to model the system one can design a stabilizing controller that works surprisingly well experimentally. The weaknesses of this work were primarily that it was unclear *why* the power dissipation method would adequately capture the dynamics, and it was moreover unclear why a feedback controller could be designed in this context. The former issue was cleared up when we showed in [16] that the power dissipation method is actually a class of kinematic reductions, in the sense of the work by Lewis et al [3, 4]. The latter issue, that of understanding why we can do control design using a heuristic modeling technique, has only recently become clear, and this paper is intended to explain why and when controller design may occur in these traditionally heuristic settings.

The key contribution of this paper is that we present a methodology for combining kinematic reductions with stabilizing controllers that only use a reduced representation of uncertainty in their estimators. When possible, this allows one to use a reduced model for computational simplicity while not losing any of the behavior of which the system is capable. We present an example of multiple point manipulation as an example, but point out that the technique of uncertainty abstraction is potentially much more broadly applicable than just what is discussed here. This paper is organized as follows. Section 2 described an example system that motivates the present work. Section 3 discusses modeling of multi-point contact systems using Lagrangian mechanics and the constrained affine connection. Although the use of the constrained affine connection description of the mechanics is absolutely equivalent to the

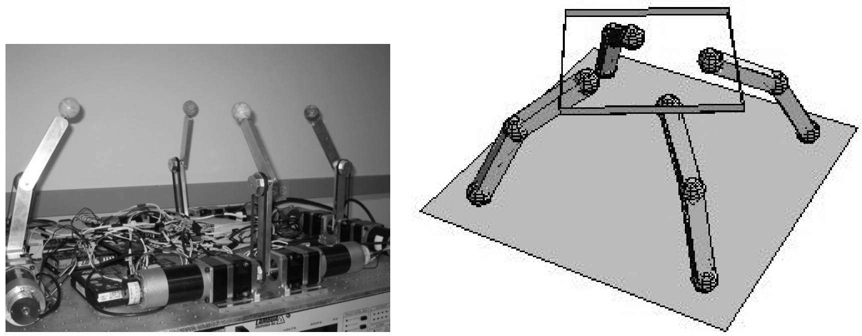

Fig. 2. An 8 degree-of-freedom four "finger" manipulator that manipulates objects it supports. The fingers are constrained, so stick/slip transitions between the actuator end-effectors and the manipulated object must occur when the actuators move. The present paper presents simulations of this device (the picture on the right with a "see-through" box supported and manipulated by four arms is the graphical representation we use in simulations).

more traditional approach using a Lagrangian and generalized coordinates, we use it because it gives a precise statement of a test for kinematic reducibility, as discussed in Section 3.1. Section 4 discusses stability results relevant to these systems, and Section 5 gives, for purpose of illustration, a quick introduction to how one applies these results to the example in Section 2 .

2 Motivation: Multi-point Manipulation

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 [11], 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. Consider the object in Fig. 2. It has eight degrees of freedom, all independently actuated by a DC brushless motor. The motion of the tips of the "fingers" can be constrained to be in a horizontal plane, so it can be used as a manipulation surface. However, the force any given finger exerts is constrained on a line–no finger can exert any "side-ways" force. 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 question is how to control the position and orientation of a supported object without being sensitive to the details of how the frictional stick/slip interactions adversely affect stability. This is addressed in Section 4 after discussing modeling issues in Section 3.

3 Modeling and Analysis of Multiple Point Contact

We assume that the systems we are interested in are finite-dimensional simple mechanical systems (as described for smooth systems in [3]). That is, their equations of motion may be found using a Lagrangian of the form kinetic energy minus potential energy ($L = K.E. - V$) along with a set of constraints on the system of the form $\omega(q)\dot{q} = 0$, where $\omega(q)$ is a matrix representing the configuration (q) dependent constraints. Moreover, there may be external forces acting on the system. If we ignore potential energy (as is appropriate for many planar systems including the one in Section 2), such a system's dynamics may be written down as: $\nabla_{\dot{q}}\dot{q} = u^\alpha Y_\alpha$, where the notation $u^\alpha Y_\alpha$ implies summation over the α. In this expression, ∇ is the constrained affine connection encoding the free kinetic energy and the constraints, in our case the nonslip constraints. Moreover, u represents external forces (not necessarily inputs) and Y represents the associated vector fields on the configuration manifold Q (i.e., $Y \in T_qQ$, the tangent space at $q \in Q$). If we wish to include potential energy, it will show up as a vector field on the right-hand side of the equation.

The systems of interest have two types of external forces–those that correspond to inputs and those that correspond to external disturbances. In the case of multiple point contact, the external disturbance forces generally correspond to reaction forces due to friction when a contact slips. Therefore, it will be useful to write the dynamic equations as: $\nabla_{\dot{q}}\dot{q} = u^\alpha Y_\alpha + d^\beta V_\beta$ so that we can distinguish between the different types of external forces. (Note that if a constraint is satisfied so that the contact is not slipping, there is still a reaction force. In that case the reaction force is incorporated into the definition of of the constrained affine connection ∇ in a manner identical to the constrained Euler-Lagrange equations).

Lastly, because the contact state changes over time (as the contacts transition between stick and slip), the constraints change over time. This implies that ∇ is not a single constrained affine connection, but rather comes from a set of constrained affine connections ∇^σ, each of which represents a different set of stick/slip states of the mechanism. The same holds true for Y^σ and V^σ. Hence, if we index the set of possible stick/slip states by σ, we get second-order equations of motion of the following form:

$$\nabla^\sigma_{\dot{q}}\dot{q} = u^\alpha Y^\sigma_\alpha + d^\beta V^\sigma_\beta \tag{1}$$

where u are input forces and d are external forces. Equation (1) represents the equations of motion for any multiple contact system or overconstrained system that experiences point contact with its environment. (Note that for this equation to make sense, one must assume that the switching signal σ is at least measurable, and often it is assumed that it is piecewise continuous.) Lastly, it is important to point out that the representation $\nabla_{\dot{q}}\dot{q} = u^{\alpha}Y_{\alpha}$ is neither more nor less than the Euler-Lagrange equations [3].

3.1 Kinematic Descriptions of Systems that Slip

We use the affine connection formalism to describe mechanical systems because it is in the context of this formalism that a useful technical connection between 2^{nd}-order mechanical systems and 1^{st}-order kinematic systems has been made (found for smooth systems in [9] and for nonsmooth systems in [16]). In particular, it would be useful to be able to write Eq. (1) in the form:

$$\dot{q} = \overline{u}^{a} X_{a}^{\sigma}, \tag{2}$$

where $\overline{u}$ are *velocity* inputs instead of force inputs. Roughly speaking, a system is kinematic if it can be written as a first order differential equation in q without losing any information about what trajectories the system is capable of producing. More precisely, this kinematic description is only useful if it satisfies two requirements. First, for every solution of the dynamic system in Eq. (1) there must exist a kinematic solution of the form in Eq. (2). In the case of a vehicle, this corresponds to requiring that for every *trajectory* of the vehicle there exists a corresponding *path* that can be obtained from kinematic considerations alone. Secondly, for every kinematic solution there must exist a dynamic solution that is equal to the kinematic solution coupled with its time derivative (so that it lies in TQ). This means that there must exist a dynamic solution for every feasible kinematic path. This way of viewing smooth kinematic systems has been studied extensively, including [9]. Motion planning has been studied using these concepts in [3, 4], but these works were all intended for smooth systems. However, it was shown in [16] that the kinematic reduction of a nonsmooth system of the form in Eq. (1) to one of the form in Eq. (2) is equivalent to the reduction of each smooth model of the multiple model system. The associated algebraic test of kinematic reducibility is that the *symmetric product* between two vector fields Y_i^{σ} and Y_j^{σ} (defined by $\left\langle Y_i^{\sigma} : Y_j^{\sigma} \right\rangle = \nabla^{\sigma}_{Y_i^{\sigma}} Y_j^{\sigma} + \nabla^{\sigma}_{Y_j^{\sigma}} Y_i^{\sigma}$ for given i, j, σ) lie within the distribution of the vector fields and that any reaction forces lie within the span of the input vector fields. That is,

$$\left\langle Y_i^{\sigma} : Y_j^{\sigma} \right\rangle \in span\{Y_i | i = 1, \ldots, m\} \;\; \forall\, i, j, \sigma \tag{3}$$

$$V_{\beta}^{\sigma} \in span\{Y_i | i = 1, \ldots, m\} \;\; \forall\, \beta, \sigma \tag{4}$$

Notice that this need only hold for each σ, so the calculation is a purely algebraic one, despite the fact that our system is nonsmooth. That is, even with the nonunique solutions these systems can have, one may test for each model independently (i.e., holding σ constant) whether a system is kinematic.

3.2 Uncertainty Representations

The main point of this paper comes from noting that the contact state enters solely in the σ dynamics in Eq. (1) and (2). Hence, the uncertainty for the full dynamic system depends on both σ and other uncertainties that drive σ, such as parametric uncertainties, choice of friction model governing the contact interaction, etcetera. However, the uncertainty in the kinematic model *only includes* σ, which means that the abstraction to the kinematic model reduces the representation of the uncertainty to a hybrid, discrete-valued structure (rather than a continuous one like that typically addressed in the robust control community). It is important to note that because of this the only assumption made regarding friction in a kinematic model is that it creates stick/slip effects. In the context of this paper, no other assumptions are necessary. However, when using such an abstraction, one must have confidence that the backstepping algorithm employed is robust with respect to the uncertainties that are left over, in our case model uncertainties and parametric uncertainties. Fortunately, motor controllers are known to be quite robust when following a desired reference velocity. Accordingly, we assume that we can track a desired velocity for the rest of this paper, ignoring transient behavior and coupling. If for some reason asymptotic tracking is not achieved, then an additional layer of analysis will be necessary. This reduced representation of uncertainty is what we will use in designing a stabilizing control for a mechanical manipulation system, and the online estimation of σ will in particular play a significant role in the stability results.

4 Stability Conditions

Now suppose we want to drive a (multiple-model, multi-point contact) mechanical system to a desired state. Then we have, for every choice of σ, a smooth system that must be stabilized (since a perfectly valid choice of σ is to have it be constant for all time). Moreover, because σ is uncertain, it must be thought of as an exogenous disturbance (albeit a discrete-valued one). Now, one could try to create a control law that is stable for all possible signals σ (in fact, one would have to do so if σ is not observable), but this is often impossible from a practical perspective. In fact, in the case of stabilizing the $SE(2)$ configuration of the object in Section 2, it is provably impossible [15]. Therefore, the question becomes one of estimation, the online estimation of the contact state σ (the abstracted uncertain variable) based on available outputs *and* the incorporation of this estimate into the controller. This latter part is important because the classical separation principle found in undergraduate controls textbooks is not valid for nonlinear or nonsmooth systems.

First, we need to know that σ is observable (i.e., different models can be distinguished based on available feedback). Although there are formal methods for determining this (see [19]), we will see in Section 5 that it is occasionally possible to see that σ is observable by inspection. If it is observable, there are generally two methods (and variations thereof) for determining the value of σ at any given time. The first is to directly compare predicted velocities (for every model indexed by σ) to the sensed velocity. This involves differentiating outputs, but may be an acceptable approach if only one derivative of the output is needed. (This is particularly true if the cardinality of values σ can take is small. That is, if the total number of models is small, so the models are relatively easy to distinguish from each other, then even with noisy data we should be able to distinguish them.) If differentiating outputs is not acceptable, then one may alternatively integrate the equations of motion for every model and compare these to the measured output. Either choice is an acceptable choice of estimator from a theoretical perspective because we are only interested in distinguishing different models from each other.

The stability results that are useful for the problems of interest here are from the adaptive control community, particularly multiple model adaptive control [6, 1, 5]. Suppose that we have a family of plants indexed by $p \in \mathbb{P}$, all of which are stabilized by a control law with Lyapunov function V_p. (For the moment, we ignore the design and implementation of these controllers. We will revisit this in Section 5.) Switching between plants is governed by the switching signal σ. In the case of a multiple contact system, σ_e encodes the externally determined *contact state* of the system, that is, which contacts are sticking and which are slipping. Moreover, σ_c encodes the current estimate of σ_e, and particularly tells us which controller is being used at any given time. Ideally, $\sigma_c = \sigma_e$, but there may be latencies that cause this not to be the case. Such systems can be written as:

$$\dot{x} = F_{\sigma(x,t)}(x,t) \qquad \sigma(x,t) \in \mathbb{P} \tag{5}$$

where $\mathbb{P}$ is an index over the set of all admissible plants. We assume that the F_p satisfy the following standard Lyapunov criteria; that there exist for all $p \in \mathbb{P}$ differentiable functions $V_p : \mathbb{R}^n \to \mathbb{R}$, positive constants λ_0, γ and class $\mathcal{K}_\infty$ [8] functions $\alpha, \overline{\alpha}$ satisfying:

$$\dot{V}_p = \frac{\partial V_p}{\partial x} F_q \leq -2\lambda_0 V_p \ for \ p = q, \tag{6}$$

$$\dot{V}_p = \frac{\partial V_p}{\partial x} F_q \leq 2\lambda_{F'} V_p \ for \ p \neq q, \tag{7}$$

$$\alpha(\|x\|) \leq V_p(x) \leq \overline{\alpha}(\|x\|), \tag{8}$$

$$V_p \leq \gamma V_q, \tag{9}$$

for all $x \in \mathbb{R}^n$ and $p, q \in \mathbb{P}$. These are relatively standard requirements for Lyapunov functions [8], except for the condition in Eq.(7) (which requires that whenever the plant and the controller are not matched the resulting instability is bounded by some growth rate $\lambda_{F'}$).

Switching signals σ are assumed to be a piecewise continuous (and therefore measurable) function coming from a family of functions S. We say that

Eq.(5) is *uniformly exponentially stable over* S if there exist positive constants c and λ such that for any $\sigma \in S$ we have

$$\|\Phi_\sigma(t,\tau)\| \leq ce^{-\lambda(t-\tau)} \qquad \forall t \geq \tau \geq 0.$$

Here $\Phi_\sigma(t,\tau)$ denotes the flow (given σ) of Eq.(5). For such a system we say that λ is its *stability margin.*

To characterize and distinguish different families of functions S, we employ the following definitions (from [6]). Given $\sigma \in S$, we define $N_\sigma(t,\tau)$ to be the (integer) number of switches or discontinuities in σ in the interval (t,τ). Given two numbers τ_{AD} and N_0, called the *average dwell time* and *chatter bound* respectively, we say that $S_{\text{ave}}[\tau_{AD}, N_0]$ is the set of all switching signals satisfying $N_\sigma(t,\tau) \leq N_0 + \frac{t-\tau}{\tau_{AD}}$. Lastly, let $S_{ave}[\tau_{AD}, N_0]$ be the set of all switching signals for which $N_\sigma(t,\tau) \leq N_0 + \frac{\tau-t}{\tau_{AD}}$. We will assume for the rest of the present work that switching signals σ_e (the external switching determining the contact state) can be characterized in this way.

Assumption 4.1 *Assume* σ_e *switching satisfies*

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

for some $N_0^e > 0$ *and* τ_{AD}^e.

We can similarly require that the signal σ_c (the switching signal that dictates the current controller) also satisfy dwell-time requirements (i.e., $N_{\sigma_c}(t,\tau) \leq N_0^c + \frac{t-\tau}{\tau_{AD}^c}$) to ensure that the control switching does not destabilize the system.

It is well known that switching between a set of stable linear systems may well yield an unstable system [7]. This means that even in the most moderate case, where estimation of the contact state is perfect (i.e., $\sigma_c = \sigma_e$) and there are no latencies in sensing or actuation, our multiple contact system can in principle be destabilized by switching contact state. Our purpose in this section is to apply some results from the theory of switching systems to understand physically meaningful conditions that will guarantee stability for a multiple model system (even those without a common Lyapunov function). In particular, we will characterize such a condition in terms of the average dwell time as it was described above.

Due to space considerations, proofs of the following theorems are not presented here and the reader is directed to the conference proceeding [13, 12] where these technical results are presented in a theorem/proof format. First, the following result from [6] will be helpful. It states that for a collection of stable plants as Eq.(5) a bound on the average dwell time can be determined such that the hybrid system is stable with any desired stability margin.

Lemma 1 ([6]). *Given a system of the form in (10) such that all the* F_p *satisfy Eqs. (6), (8), and (9) hold, there is a finite constant* τ_{AD}^* *such that Eq.(5) is uniformly exponentially stable over* $S_{ave}[\tau_D, N_0]$ *with stability margin*

$\lambda < \lambda_0$ *for any average dwell time* $\tau_{AD} \geq \tau^*_{AD}$ *and any chatter bound* $0 < N_0$.

In particular, the average dwell time must satisfy $\tau_{AD} > \frac{\log \gamma}{2(\lambda - \lambda_0)}$. Note that if we have a common Lyapunov function, then $\gamma = 1 \Rightarrow \log \gamma = 0 \Rightarrow \tau_{AD} = 0$ satisfies the stability requirements. Hence, common Lyapunov functions are highly desirable, if they can be found. A corollary of this result relevant to the multiple point contact example is Corollary 1 (proven in [13]).

Corollary 1. *If each contact state* σ_e *for a multi-point manipulation system is stabilized with a quadratic Lyapunov function* V_p, *if* $\sigma_e \in S_{ave}[\frac{\log \gamma}{2(\lambda - \lambda_0)}, N_0]$ *for some* N_0, *and if* $\sigma_c = \sigma_e$ *(i.e., the observer is perfect), then Eq. (1) or Eq. (2) (depending on whether the representation used is dynamic or kinematic) is exponentially stable with stability margin* λ.

Note that this result, and the results that follow, are equally applicable to both dynamic and kinematic systems. What does Corollary 1 mean for a multiple model system where there are external signals determining the switching, such as is the case in a multiple contact system? It means that so long as there are no latencies, no errors in estimation, and no noise in the sensors, the multiple model system is stable so long as the external switching signals σ_e are kept sufficiently slow on the average. How slow depends on how the controllers for each plant are designed and, more importantly, how they are related to each other. The closer γ can be kept to 1 (i.e., the closer we are to having a common Lyapunov function), the more quickly σ_e may switch without destabilizing the system.

What happens if there are noise sources, latencies, and time delays causing the controller switching σ_c to not coincide with the environmental switching σ_e? Most of these issues are adequately addressed in [1, 5]. However, if $\sigma_c \neq \sigma_e$, instabilities due to temporary mismatch between controllers and plants can occur. The basic consequence of this is roughly that the longer the mismatch, the slower the external switching must be in order to maintain stability. To address this issue, assume we have equations of motion of the following form:

$$\dot{x} = \begin{cases} F'_q x \text{ on } [t_i, t_i + d_\sigma) \\ F_p x \text{ on } [t_i + d_\sigma, t_{i+1}) \end{cases} \tag{10}$$

where (for each p) $\dot{x} = F_p(x)$ is asymptotically stable and (for each q) $\dot{x} = F'_q$ is potentially unstable but has a bound on the rate of growth $\lambda_{F'}$. Note that our example system in Section 2 satisfies these requirements because all the F_p are stable by design.

It is now useful to state an extension of Thm. 1 (also proven in [13]) to accommodate d_σ. The resulting trade-off is not surprising–the larger d_σ becomes, the more slowly σ_e is allowed to switch. In particular, if we can bound d_σ below by d_* then we find that choosing

$$\tau^e_{AD} > \frac{\frac{\log \gamma}{2} + 2\lambda_{F'} d_\sigma}{(\lambda_0 - \lambda)} \tag{11}$$

results in a stable system, as seen in the following Lemma.

Lemma 2. *Given a system of the form in (10) such that all the F_p satisfy Eqs. (6), (7), (8), and (9), there is a finite constant τ^*_{AD} and a finite constant d^*_σ such that Eq.(10) is uniformly exponentially stable over $\mathcal{S}_{ave}[\tau_{AD}, N_0]$ with stability margin λ, for any average dwell time $\tau_{AD} \geq \tau^*_{AD}$, any chatter bound $0 < N_0$, and any $d_\sigma \leq d^*_\sigma$.*

With this, one may prove the following corollary.

Corollary 2. *If each contact state for a multi-point manipulation system is stabilized with a quadratic Lyapunov function V_p, and if τ^e_{AD} and d_σ satisfy Eq.(11), then for any N_0 the state output is exponentially stable.*

Proposition 2 indicates that if the contact states change slowly enough (i.e., τ^e_{AD} is large) and the estimator is fast enough (i.e., d_σ is small), then the system is stable. Among other things, this means that one does not have to concern oneself with the friction model to establish where switching occurs. Instead, the contact states can change arbitrarily, so long as they do so sufficiently slowly on the average and their effect is observable in the state output.

5 Example

Consider the eight degree of freedom manipulator in Fig. 2. This figure has four point contact actuators (corresponding to the inputs $u_1, \ldots, u_4$) located at $(1,1), (-1,1), (-1,-1), (1,-1)$ respectively (in the simulation), all oriented towards the origin. For each contact there are two independent constraints, a nonslip constraint in each direction tangent to the surface of the contact. Hence, there are $2^{2 \cdot 4} = 2^8 = 256$ possible combinations of stick and slip for the four point contact system. If one uses a symbolic software package such as *Mathematica* to compute the dynamic equations of motion for every possible contact state as in Eq. (1), one can exhaustively verify that all possible models are kinematic, so long as the contact interfaces are *dissipative* when slipping is occurring (i.e., the reaction force is nonzero and in the opposite direction of the slipping). This represents an extremely broad set of frictional interfaces, and the statement is proven in a non-exhaustive manner in [12]. Additionally, all the nontrivial, non-overconstrained kinematics are of one of the four forms in Table 5. There do exist σ_e with trivial kinematics (i.e., actuator velocities do not make the supported object move at all), however, and these correspond to constraints with no actuation. An example of this is a table with wheels that are all razor thin, so that spinning the wheels exerts very little force against an object, but sliding orthogonally to the wheel is very difficult. In such an example, no movement whatsoever occurs, and such a situation must be either be avoided through mechanical design or avoided online, but this is beyond the scope of what we discuss here.

Table 1. The four actuator manipulation surface shown in Fig 2 has all kinematic states, many of which are redundant. This figure shows the four distinct equations of motion that can occur in different contact states. Note that so long as u_1 $(= -u_3)$ and u_2 $(= -u_4)$ are nonzero, the four states can be distinguished from state output. In fact, just observation of θ is sufficient for distinguishing the states. Moreover, measurements of x and y are not helpful because the x and y dynamics are identical in all four models.

Equations of Motion	Control Law
$\dot{q} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} u_2$	$u_1 = \frac{-k\theta\,(\theta+x-y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$ $u_2 = -k\theta$
$\dot{q} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} u_2$	$u_1 = k\theta$ $u_2 = \frac{k\theta\,(\theta+x+y)-k\left(\theta^2+x^2+y^2\right)}{x-y}$
$\dot{q} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} u_2$	$u_1 = \frac{k\theta\,(\theta-x+y)+k\left(\theta^2+x^2+y^2\right)}{x+y}$ $u_2 = k\theta$
$\dot{q} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} u_2$	$u_1 = -k\theta$ $u_2 = \frac{-k\theta\,(-\theta+x+y)-k\left(\theta^2+x^2+y^2\right)}{x-y}$

For each of the four models in Table 5 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. Moreover, by virtue of the design methodology, there is a common Lyapunov function (i.e., $\gamma = 1$ in Eq. (9)). Hence, chattering may occur (particularly near the planar origin), but will not affect stability. Things to note include the following.

1. The system is not smoothly locally controllable (since there are two constant input vector fields and three configuration variables to be controlled). However, all of the states are stabilizable to the the origin of $SE(2)$.
2. Note that these control laws are not only nonlinear, they are not even smooth. In fact, they have discontinuities at the origin.
3. The four models are distinguishable (e.g., based on state output, one can distinguish each model from the next) given nonzero inputs. This will be how we estimate σ_e in the simulations.
4. The four models are, in fact, distinguishable based entirely on θ output (i.e., one may construct a "reduced-order" hybrid observer).

Figure 3 shows a simulation of the four actuator system using $k = 1$. We simulate the kinematic system rather than the dynamic one, but we are currently making a dynamic simulation to explicitly incorporate various modeling choices (particularly of friction) in the simulation. In either case, the control design should be done at the kinematic level to allow for the abstraction of uncertainty we are advocating. We use crossing from one quadrant to another as the way to drive σ_e in the simulation (which is motivated by minimizing

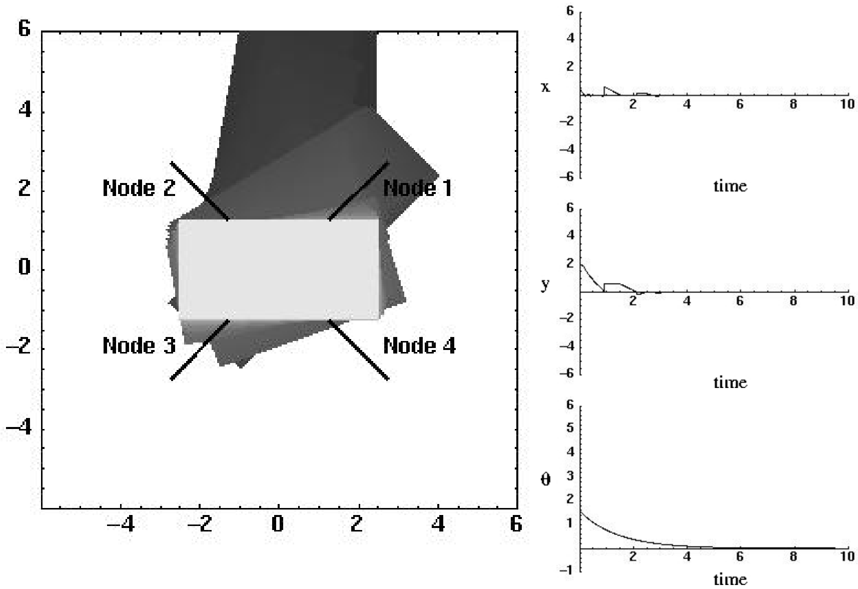

Fig. 3. Simulation of multi-point manipulation when $\sigma_c = \sigma_e$. The rectangle represents the center of the object which is actually in contact with all four of the actuators (Nodes 1-4). The time history progresses from dark rectangles at time 0 to the light rectangles at time 10. The three plots are plots of the X,Y, and θ coordinates against time.

the power dissipation, see [15]), but estimate σ_e online in the simulation. The actuators can only push in one direction for a short amount of space before reaching a kinematic singularity, so they are reset occasionally (this effect shows up in the estimation of σ_e). 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 in the figure by Nodes 1-4. Their range of motion is depicted by a dark line next to Node X. 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 (and did so reliably over many simulation runs not depicted here). Moreover, this trajectory is qualitatively very similar to the trajectories found experimentally in [15].

The simulation is structured as follows. Depending on which quadrant the center of mass of the object is in, σ_e is chosen to be one of the four models in Table 5. Then, all four models are integrated with respect to time while applying an initial control value of $u_1 = 0, u_2 = 0.1$ (it doesn't matter

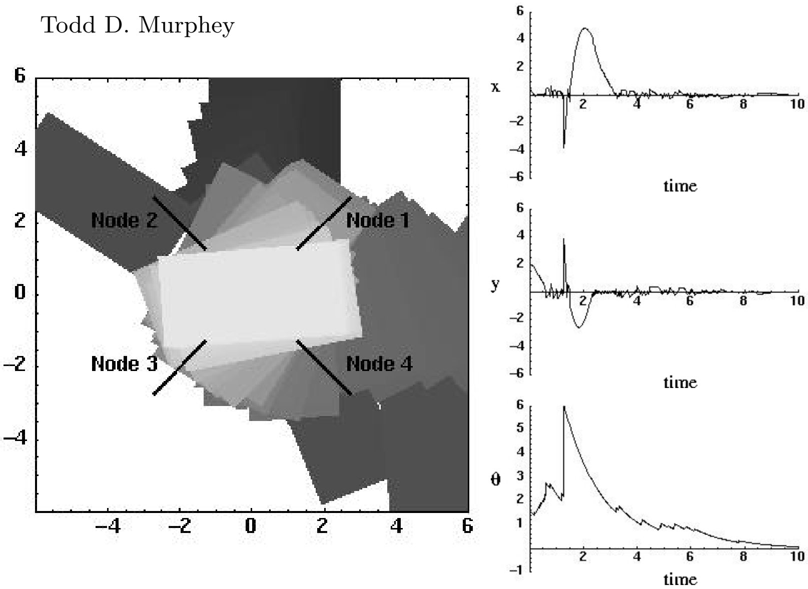

Fig. 4. Simulation of multi-point manipulation when $\sigma_c \neq \sigma_e$ (i.e., the estimated contact state is delayed). The object is only barely stabilized to the origin. (As in Fig. 3, the three plots are plots of the X,Y, and θ coordinates against time.)

what this initial control is, so long as it is nonzero). Based on this, σ_e can be immediately determined by looking at the evolution of θ. We did this rather than directly comparing velocities so that we are not differentiating the output. In any case, this then determines σ_c. Knowing σ_c, we ask the actuator tips to follow the velocities u_i based on the control laws in Table 5. These are implemented with an inverse Jacobian, except for when the actuator tip reaches one of its limits, in which case we reset it to the other end of its range (the dark lines in the figure). We can add noise to the sensed state variables and time delays to the estimated σ_c.

If σ_c is a bad estimate of σ_e, then performance degrades but stability is not lost, as seen in Fig. 4, where a time delay of one tenth of a second is introduced. (Note that the amount of time delay in a kinematic system is scalable by virtue of changing the gain on the controller.) Adding a small amount of noise to the sensed outputs has roughly the same effect as a small time delay, as we would expect. If the time delay for a given gain is made sufficiently large, the system becomes unstable. This indicates, at least in simulation, that the interpretation and application of the stability theorems in Section 4 are appropriate here, and that performance degrades reasonably gracefully as σ_c becomes less and less of a good estimate of σ_e until eventually the system destabilizes.

6 Conclusions

In this paper we have described methods applicable to the modeling and control of mechanical manipulation problems, including those that experience uncertain stick/slip phenomena. The particular contribution of this work is to point out that the use of an abstraction, in this case a kinematic reduction, not only reduces the computational complexity but additionally simplifies the representation of uncertainty in a system. Moreover, this simplified representation may be directly used in a stabilizing control law. The end result of this is two-fold. First, modeling for purposes of control is made more straightforward by getting rid of some dependencies on low-level mechanics (in particular, the details of friction modeling). Second, the online estimation of the relevant uncertain variables is much more elegant and easily implementable than the online estimation of the full model and its associated uncertainties. For instance, online friction system identification is quite complex and is not feasible for many applications. However, the presentation here assumes that all feasible states for the system are indeed kinematically reducible. If they are not, then one must switch back to a full analysis of the uncertainty. Moreover, enumeration of the kinematic states can be computationally challenging because in principle the number of kinematic states can go up exponentially in the number of contacts. We expect to be able to address this by more formally using the discrete symmetry properties that allowed us to reduce to only four states in Table 5.

We stabilize the system using techniques from multiple model adaptive control as developed in [1, 5, 7]. We demonstrate in simulation that this technique works well in the context of a simple example (based on experimental work seen in Fig. 2). Moreover, the model/controller presented in the context of this example does not include any explicit model of friction, making the proposed techniques applicable to cases where an unstructured environment makes it unlikely that one can model frictional interactions accurately. Instead, one moves some of the robustness requirements to the backstepping algorithm employed, hence reducing the uncertainty representation with which the high-level controller must contend.

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 particular, examples where a common Lyapunov does not exist should be examined in depth using the analytical techniques developed here. In the meantime, these results will be implemented both in a dynamic simulation environment we are developing and on a second generation version of the experiment discussed in Section 2 and seen in Fig. 2.

Acknowledgements
The author gratefully acknowledges the support for this work provided by the NSF under CAREER award CMS-0546430.

References

1. B. Anderson, T. Brinsmead, F. D. Bruyne, J. Hespanha, D. Liberzon, and 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):909–929, Sep 2000.
2. A. Bicchi and V. Kumar. Robotic grasping and contact: a review. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 348–353, 2000.
3. F. Bullo and A. Lewis. *Geometric Control of Mechanical Systems*. Number 49 in Texts in Applied Mathematics. Springer-Verlag, 2004.
4. F. Bullo, A. Lewis, and K. Lynch. Controllable kinematic reductions for mechanical systems: Concepts, computational tools, and examples. *Int Symp on Math. Theory of Networks and Systems (MTNS)*, August 2002.
5. J. Hespanha, D. Liberzon, A. Morse, B. Anderson, T. Brinsmead, and 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):479–496, April 2001.
6. J. Hespanha and A. Morse. Stability of switched systems with average dwell-time. Technical report, EE-Systems, University of Southern California, 1999.
7. S. in Systems and Control. *D. Liberzon*. Birkhäuser, Boston, 2003.
8. H. Khalil. *Nonlinear Systems (second edition)*. Prentice Hall, 1996.
9. A. Lewis. When is a mechanical control system kinematic? In *Proc.* 38^{th} IEEE *Conf. on Decision and Control*, pages 1162–1167, Dec. 1999.
10. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. *Formal Methods in Systems Design*, volume 6, chapter Property preserving abstractions for the verification of concurrent systems, pages 1–35. Kluwer, 1995.
11. J. Luntz, W. Messner, and H. Choset. Distributed manipulation using discrete actuator arrays. *Int. J. Robotics Research*, 20(7):553–583, July 2001.
12. T. Murphey. Modeling and control of multiple-contact manipulation without modeling friction. In *Proc. American Controls Conference (ACC)*, 2006.
13. T. D. Murphey. Application of supervisory control methods to uncertain multiple model systems. In *Proc. American Controls Conference (ACC)*, 2005.
14. T. D. Murphey and J. W. Burdick. *Algorithmic Foundations of Robotics V*, chapter Feedback Control for Distributed Manipulation, pages 487–503. Springer Verlag, 2004.
15. T. D. Murphey and J. W. Burdick. Feedback control for distributed manipulation with changing contacts. *International Journal of Robotics Research*, 23(7/8):763–782, July 2004.
16. T. D. Murphey and J. W. Burdick. The power dissipation method and kinematic reducibility of multiple model robotic systems. *IEEE Transactions on Robotics*, Accepted for publication.
17. H. Olsson, K. Astrom, C. C. de Wit, M. Gafvert, and P. Lischinsky. Friction models and friction compensation. *European Journal of Control*, 4(3):176–195, 1998.
18. G. Pappas, G. Laffierier, and S. Sastry. Hierarchically consistent control sytems. *IEEE Trans. Automatic Control*, 45(6):1144–1160, June 2000.
19. D. D. Vecchio and R. Murray. Discrete state estimators for a class of nondeterministic hybrid systems on a lattice. In *Hybrid Systems: Computation and Control*, Philadelphia, Pennsylvania, 2004.