

# Global Exponential Stabilizability for Distributed Manipulation Systems

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 considers the global exponential stability of planar distributed manipulation control schemes. The “programmable vector field” approach is a commonly proposed method for distributed manipulation control. In [13] it was shown that when one takes into account the discreteness of actuator arrays and the mechanics of actuator/object contact, the controls designed by the programmable vector field approach can be unstable at the desired equilibrium configuration. We show here how a discontinuous feedback law that locally stabilizes the manipulated object at the equilibrium can be combined with the programmable vector field approach to control the object’s motions. We prove that the combined system is globally exponentially stabilizable even in the presence of changes in contact state. Simulations illustrate the results.

## I. Introduction and Previous Work

A *distributed manipulation system* consists of a roughly planar array of actuators that can re-position an object by the movements of its array elements. In the future, arrays of this type could be useful for assembly operations where parts must be robustly transported and precisely positioned. This paper considers the global behavior and convergence of distributed actuator control systems. We present a method for globally exponentially stabilizing a part on a planar actuator array, and prove the global stability of this method. Our method takes into account the discreteness of the array’s actuators and reasonable models of object/array contact.

Methods to design distributed manipulation control systems have been proposed in several works, including [4] and [6]. A common approach is based on the notion of *programmable vector fields*. This method was first pioneered in [2], and is well summarized in [3]. In this methodology, one makes the possibly unrealistic assumption that the array’s actuation and control capability can be idealized as a continuous distribution of forces across the array surface. In this abstraction, the manipulated object moves under the influence of these continuously distributed forces. The control design problem reduces to the selection of a continuous force field distribution that will locally transport the object to a prescribed position, and then stabilize it at that configuration. The basic control strategy is to choose a sequence of force fields which move the part from one equilibrium to

another equilibrium. To implement the control strategy on the real array, one must adapt the continuous vector field control to the real (and discrete) actuator array.

This approach has been experimentally demonstrated in MEMS-fabricated actuator arrays, where the array elements are “small” and “close” together relative to the object being manipulated ([3]). However, when the actuators are far apart (i.e., the continuous actuation approximation is poor) or the contact mechanics can not be neglected, the continuous approximation is known not to work as well (see [13]). In these cases, the continuous approximation does not adequately incorporate the discrete nature of the actual array and the contact mechanics of the object/array interface. In [13] we showed that when one uses the programmable vector field method, these effects will lead to instability in the orientation ( $\theta$ ) component of the moving object’s  $SE(2)$  location. We then introduced a local feedback law that stabilizes the object at the equilibrium.

While [13] considered the local stabilizability and control of the moving object near its equilibrium configuration, [12] considered the global stabilizability and control synthesis problem. In particular, we presented a way of blending the philosophies of [3] and of [13] to produce a globally stabilizing control which only requires local feedback near an equilibrium point. The requirement that feedback need only be used near the equilibrium leads to economical implementations. Here we expand that philosophy to include exponential convergence to the origin. By enforcing exponential convergence, we guarantee that our combined control system will “quickly” take the manipulated object to the desired equilibrium configuration.

Section II reviews the basic background required for our analysis, and summarizes the previous results and modeling methodologies that we build upon. Section III presents the paper’s main theorems, along with a discussion of how it applies to the case of distributed manipulation. Section IV presents illustrative simulation results.

## II. Background

Here we review the necessary aspects of the programmable vector field approach (see e.g., [3]), a *power dissipation method* (PDM) for constructing models of discrete actuator arrays, and some necessary results from the theory of nonsmooth control vector fields.

## A. Programmable Force Fields

The programmable vector field approach to distributed manipulator control is based on a continuous “force field” abstraction which assumes that one can specify the manipulation force at each point on the manipulation surface. The moving object’s motions are obtained by integrating the force field to obtain the net force on the part as follows. Assume the part  $w$  can be described by a support characteristic function  $w(x, y)$  where  $w(x, y)$  is 1 everywhere on the object surface and 0 otherwise. Moreover, let the part be subject to a force field  $f(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ . Lastly, let  $w$ ’s reference frame be affixed at the center of mass, i.e.

$$\int_{\mathbb{R}^2} \omega(p) dp = 0.$$

When the object lies at configuration  $q = (x, y, \theta)$  the net force and torque on the object are:

$$F = \int_{\mathbb{R}^2} \omega(p) f(A_\theta p + t) dp \quad (1)$$

$$M = \int_{\mathbb{R}^2} \omega(p) A_\theta p \times f(A_\theta p + t) dp \quad (2)$$

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

The most basic control law uses a “squeeze” field, which generally takes the form  $F = \{-\alpha x, -\beta y\}$  where  $\alpha$  and  $\beta$  are coefficients to be chosen by the control designer. These open loop controls can stabilize an object to one of two stable equilibria. Using this idea as a basis, significant work has been done to produce unique stable equilibria unique stable equilibrium (e.g., [14]).

To use these controls on an actual array, where the manipulation forces will be generated at discrete points, one must adapt the continuous approximation to the given discrete geometry. However, when the actuators are discrete, and the contact mechanics are nonnegligible, the use of the controls derived from the continuous approximation is known not to work as well, because the object’s motions vary dramatically depending on the contact state. In [13] we showed that rotational instability will result in the programmable force field approach due to the switching of contact states that arises from the contact mechanics and the discreteness of the actuator array.

**Remark:** Although the inputs for the programmable force field are forces and the inputs we are going to consider are vector field inputs (velocities), these two are in reality often the same because the forces are typically generated by the friction at a slipping contact point. That is,  $F = -\mu N v$  where  $\mu$  is the coefficient of friction,  $N$  is the normal force, and  $v$  is the slipping velocity. The input classes are equivalent in this case.

## B. The Power Dissipation Model

To analyze control system performance, we seek models for the governing equations of distributed actuation sys-

tems that faithfully capture the system’s essential physics, and that are tractable and amenable to control and motion planning analysis. In pursuit of this goal, we use a “power dissipation model” (PDM) approach to model the governing dynamics of a discrete actuator/object system. This method produces unique models that are relatively easy to obtain, and to which one can apply control system analysis methods. Since the method is a quasi-static modeling method, it produces first-order governing equations, instead of second order equations that are associated with Lagrange’s equations. The primary disadvantage is that the method only applies to quasi-static systems. In [13] we argue that this assumption is quite good for our problems of interest, and for distributed manipulation arrays in general. Here we present an overview of the modeling methodology, and refer the reader to [13] and [11] for more details.

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 array element. Let us assume that the motion of the actuator array’s variables are known. We treat the object and the array element contact as a rigid body contact system (though approximate compliance effects can be easily incorporated into this paradigm). We assume point contacts between the object and the array elements (complex contacts such as line contacts are modeled as a set of point contacts in this approach), and that the object’s contact with the manipulating surface is governed by the Coulomb friction law at each contact point.

It is not hard to show that the relative motion of each contact between the object and an actuator array element can be modeled in the form  $\omega(q)\dot{q}$ . If  $\omega(q)\dot{q} = 0$ , the contact is not slipping (i.e., it is nonholonomic), 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, there by 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

$$\mathcal{D} = \sum_{i=1}^n \alpha_i |\omega(q)\dot{q}| \quad (3)$$

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

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

**Power Dissipation Principle:** With  $\dot{q}$  small, an object’s motion at any given instant is the one that minimizes  $\mathcal{D}$ .

The *power dissipation method* assumes that the object’s motion at each instant is the one that instantaneously min-

minimizes power dissipation due to contact slippage. This method is adapted from the work of [1] on wheeled vehicles. Here we briefly summarize some of the PDM’s formal characteristics, which were first developed in the context of wheeled vehicles ([11]). In particular, we show that the power dissipation approach generically leads to multi-model (or hybrid) systems. For the omitted proofs, a greater discussion of these results, and a discussion of the relationship between the PDM and Lagrangian approaches for such a system, see [11].

The next definition describes the kind of system one obtains by using the PDM approach.

*Definition 2:* A system is a *multi-model driftless affine system (MMDA)*<sup>1</sup> if it can be expressed in the form

$$\dot{x} = f_{\sigma_1}(x)u_1 + f_{\sigma_2}(x)u_2 + \dots + f_{\sigma_n}(x)u_n \quad (4)$$

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

An MMDA is a driftless affine nonlinear control system where each control vector fields may “switch” back and forth between different elements of a finite set. In our case, this switching corresponds to the switching between different contact states between the object and the array surface elements (i.e., different sets of slipping contacts) due to variations in contact geometry, surface friction properties, and normal loading. In [11] it was shown that the PDM generically leads to MMDA systems as in Definition 2.

**Remark:** Here we should comment on the relationship between the philosophies of the PDM and programmable force field approaches. The programmable force field method effectively assumes that there are an infinite number of actuators, that all of the actuators are slipping all the time, and that the physics of contact between the array surface and the object is not that important. Hence, the programmable force field method is more appropriate to the analysis of gross motions where accuracy is less important and its simplicity of analysis and design is appealing. The PDM assumes that there is a finite number of discrete contacts governed by a Coulomb friction contact model. However, it is formally only applicable to quasi-static motions. As we argue in [13], the quasi-static assumption is generally quite good for distributed actuator arrays, and is a particularly good assumption near the object’s equilibrium state.

### C. Modeling the Equilibrium Point of a Programmable Vector Field

The PDM modeling approach is applicable to a wide variety of interfaces between the moving object and the array

<sup>1</sup>By abuse of notation, in [11, 13] we referred to these systems as “switched driftless affine systems”. The nomenclature used here adheres to the most commonly used nomenclature in the control system literature

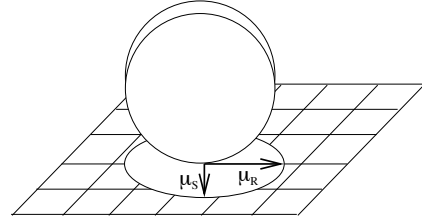


Fig. 1. A wheel with anisotropic friction

actuators: rotating wheels, moving fingertips, sliding surfaces, etc. We only assume that local object/actuator contact is significantly influenced by Coulomb friction. We allow for a smooth, but non-uniform, distribution of the coefficient of friction in all directions of the contact plane, like that seen in Figure 1 (see [7] for a discussion of such friction models). While some materials do have friction of this type, such anisotropic friction models are more generally useful as a means to approximately model compliance effects and wheel tread effects. For instance, if the wheel shown above was as thin as a saw blade, we would expect  $\mu_R$  (the friction coefficient along the “rim” direction) to be less than  $\mu_S$  (the friction coefficient along the “side” direction), and in doing so we would be modeling the non-ideal point contact in terms of a variable coefficient of friction. However, the treads on a tank ensure that  $\mu_R$  is greater than  $\mu_S$ . Note that the minimum of the dissipation function will only be non-unique when the ellipse reduces to a circle (i.e.  $\mu_S = \mu_R$ ). Moreover, the same indeterminacy shows up in the analogous Lagrangian analysis.

### D. Equations of Motion

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

$$\Omega_i(q) = \begin{bmatrix} Ad_{g_i}^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} \cos \theta_i \\ \sin \theta_i \\ 0 \end{pmatrix} u_i \\ Ad_{g_i}^T \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} & 0 \end{bmatrix} \quad (5)$$

and

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

is the homogeneous representation of the  $i^{th}$  actuator node location and orientation relative to a fixed reference frame,

$Ad(\cdot)$  is the adjoint transformation which transforms velocities from one coordinate frame to another, and  $R(\cdot)$  is an element of  $SO(2)$ .

To apply the PDM to this problem, first note that the minimum only occurs when three of the constraints are satisfied (i.e. when for a choice of three values of  $i$  we have  $\Omega_i(q)\dot{q} = 0$ ), and that moreover, the constraints satisfied are precisely those which would otherwise dissipate the most energy if they were violated. The contact states that dissipate the most energy are those associated with the potential constraints having the largest three normal forces  $\alpha_i = N_i\mu_i$ . Based on these observations, if the center of mass determines the normal forces (based on assumptions about surface uniformity, etc.), and if  $\mu(x, y)$  is uniform, then the object's motion satisfies whichever constraints are closest to its center of mass. That is, the particular region in which the center of mass lies determines the first two actively satisfied constraints. The third actively satisfied constraint is determined by the friction model. The system equations are found by solving for the annihilator of the constraint  $\Omega(q)$ . If  $\mu_S < \mu_R$ , and if the two dominating actuators are indexed by  $i$  and  $j$ , then the kinematics are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{u_i[s_j((x_i-x_j)c_i+y_i s_i)+c_i c_j y_j]-u_j y_i}{(x_j-x_i)s_j+(y_i-y_j)c_j} \\ \frac{u_j x_i - u_i[c_i c_j x_i + s_i(x_j s_j + (y_i-y_j)c_j)]}{(x_j-x_i)s_j+(y_i-y_j)c_j} \\ \frac{u_j - u_i \cos(\theta_i - \theta_j)}{(x_i-x_j)s_j+(y_j-y_i)c_j} \end{bmatrix} \quad (6)$$

where  $c_i = \cos(\theta_i)$ ,  $s_i = \sin(\theta_i)$ , etc. It should be noted that here the index notation should be thought of as mapping  $(i, j)$  pairs to equations of motion in some neighborhood (not necessarily small) around the  $i^{\text{th}}$  and  $j^{\text{th}}$  actuator. The transition between the equations of motion determined by actuators  $i$  and  $j$  to equations of motion determined by actuators  $k$  and  $l$  will in general be determined by the location of center of mass. This in turn leads to the state space being divided up by transition boundaries between different sets of equations of motion. To write this as an MMDA system, we may rewrite the above system as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f_1 u_1 + f_2 u_2 \quad (7)$$

where

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

## E. Stability Theorems for Differential Inclusions

In [13] we showed that when one takes the contact mechanics and discreteness of the actuator array into account, the orientation  $\theta$  of the moving object is unstable

at the equilibrium configuration when one uses the programmable vector field approach. First we note that the system in Eq (7) is a differential inclusion of the type found in [5]. I.e. due to the switching of the control vector fields in Eq. (4), the governing equations take the form  $\dot{x} \in F(t, x)$ , where  $F(t, x)$  is a (set-valued) multi-function. Our analysis requires the following theorem. This theorem takes its first two parts from a theorem in [5] and is the generalization of time varying Lyapunov theory to differential inclusions. The third part is a natural extension of the Lyapunov theorem to exponential stability, and the proof is included here for completeness. We remind the reader that the upper and lower derivatives for a function  $V(t, x) \in C^1$  are defined by:

$$\dot{V}^* = \sup_{y \in F(t, x)} (V_t + \nabla V y) \quad \dot{V}_* = \inf_{y \in F(t, x)} (V_t + \nabla V y) \quad (8)$$

*Theorem 1:* Let, in a closed domain  $D(t_0 \leq t < \infty, |x| \leq \epsilon_0)$ , the differential inclusion  $\dot{x} \in F(t, x)$  satisfy the basic conditions of existence and  $0 \in F(t, 0)$ ; in this domain, let there exist functions  $V(t, x) \in C^1, V_0(x) \in C$  for which

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

Then:

1) If  $\dot{V}^* \leq 0$  in  $D$ , the solution  $x(t) = 0$  of the inclusion  $\dot{x} \in F(t, x)$  is stable.

2) If, moreover, there exist functions  $V_0(x), V_1(x) \in C, W(x) \in C$  (for  $|x| \leq \epsilon_0$ ) and

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

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

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

3) Moreover, if there exist  $k_1, k_2, k_3, c > 0$  such that

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

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

**Proof:** Here we only show part 3), since parts 1) and 2) are proven in [5]. Combining assumptions in 2) and 3) we know that

$$k_1 \|x\|^c \leq V(x) \leq k_2 \|x\|^c$$

and that

$$\dot{V}^* \leq -W(x) \leq -k_3 \|x\|^c \leq -\frac{k_3}{k_2} V(x)$$

now this implies by the comparison lemma (see [9, page 85]) that

$$V(x) \leq V(x_0) e^{-(k_3/k_2)(t-t_0)}$$

Therefore,

$$\begin{aligned} \|x\| &\leq \left[ \frac{V(x(t))}{k_1} \right]^{1/c} \leq \left[ \frac{V(x_0)e^{-(k_3/k_2)(t-t_0)}}{k_1} \right]^{1/c} \\ &\leq \left[ \frac{k_2 \|x_0\|^c e^{-(k_3/k_2)(t-t_0)}}{k_1} \right]^{1/c} \\ &= \left( \frac{k_2}{k_1} \right)^{1/c} \|x_0\| e^{-(k_3/k_2c)(t-t_0)} \end{aligned}$$

Therefore, the origin is exponentially stable.  $\square$

In our generalization we assume that all the actuators are a finite distance apart and make point contact with the object being manipulated. Moreover, we assume that at the  $i_{th}$  actuator we have an input denoted by  $u_i$  that controls the velocity of the actuator. Therefore, at each actuator we can define a vector quantity  $\Psi$  that takes a point in  $\mathbb{R}^2$  to  $\mathbb{R}^2$ , the tangent space. An *elliptic vector velocity field* is one one of the form  $\Psi(x, y) = (-ax, -by)$ . The following theorem indicates that the induced instability of the programmable vector field approach can arise in more general circumstances. The proof of the theorem relies essentially upon book-keeping the conditions found in parts 1) and 2) of Theorem 1, and can be found in [13].

*Theorem 2:* Given an elliptic vector velocity field  $\Psi(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , and a discrete planar array geometry in some neighborhood of the origin, the solution to the governing equations given by the PDM is stable in  $(x, y)$  and unstable in  $\theta$ . Moreover, if  $\mu_S < \mu_R$ , then such a system is stabilizable through a discontinuous feedback law.  $\diamond$

The case  $\mu_R < \mu_S$  may be stabilizable, but not in as straight forward a fashion.

### F. A LaSalle Theorem

The major goal of [12] was to “blend” the programmable vector field approach with our local feedback law of [13]. I.e., we used a programmable vector field to govern the object’s gross motions far away from the equilibrium point, and our local stabilizing feedback law in the vicinity of the equilibrium configuration. The intuition behind this approach is that if we can move a package from one point  $a$  in the plane to an equilibrium point  $b$ , and if we have feedback in a closed neighborhood  $\mathcal{M}$  of point  $b$ , we can allow the package to spin freely along its path to  $b$ , and wait to concern ourselves with the package’s orientation until it has entered  $\mathcal{M}$ .

To effect this blending, we developed a variation of the classical LaSalle Invariance Principle (see [10]). The basic difference between the classical version of the LaSalle theorem and ours is that we must consider systems governed by differential inclusions. In such systems, the idea of a “flow” does not necessarily include uniqueness. That is, our technical results must hold for *any* flow  $\phi(t)$  satisfying  $\dot{\phi} \in F(t, x)$ . The proof of the following theorem can be found in [12].

*Theorem 3:* Let  $\phi(t, x_0)$  be a flow that satisfies  $\dot{x} \in F(x, t)$ , starting from  $x_0$ . Let  $\mathcal{M}$  be a simply connected,

compact set which is positively invariant under all flows  $\phi(t, p)$  satisfying the differential inclusion  $\dot{x} \in F(x, t)$  ( $\mathcal{M}$  is positively invariant if  $\dot{V}^*(x) \leq 0$  for all  $x \in \mathcal{M}$ , where  $\dot{V}^*$  is defined in (8)). Now let there exist a Lyapunov function  $V$  and sets  $E$  and  $N$  such that

$$E = \{x \in \mathcal{M} \mid 0 \in \dot{V}(x)\}$$

$$N = \left\{ \bigcup \phi(t, x_0) \mid x_0 \in E \text{ and } \phi(t) \in E \forall t > 0 \right\}$$

(I.e., the Lyapunov function  $V$  is zero on the set  $E$ , and  $N$  is the union of all trajectories that start in  $E$  and remain in  $E$  for all  $t > 0$ ) Then, for all  $x \in \mathcal{M}$ ,  $\phi(t, x) \rightarrow N$  as  $t \rightarrow \infty$ .  $\diamond$

To apply this theorem to distributed manipulation, we must show that a distributed manipulator will satisfy the requirements and assumptions of Theorem 3. This will lead to the following Corollary of Theorem 3. Assume the distributed system can be represented by an array of actuators  $a_{ij}$  with the coordinate location of  $(x_i, y_j)$ , and assume that the PDM model solution depends only on the center of mass (equivalently, that the coefficient of friction is uniform). For us,  $\mathcal{M}$  is the distributed manipulators feedback region.

*Corollary 4:* Given a discrete planar array geometry, an elliptic vector velocity field  $\Psi(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  outside of  $\mathcal{M} = B_\epsilon \times \mathcal{S}_\infty$  for some  $\epsilon > 0$ , and a locally stabilizing feedback law (which we know exists by Theorem 2) the solution to the governing equations given by the PDM is globally stable.  $\diamond$

### III. The Main Result

In our generalization we assume that all the actuators are a finite distance apart and make point contact with the object being manipulated. The following theorem indicates that the induced instability of the programmable vector field approach can be fixed with a local feedback law, and that moreover the performance can be made exponential.

*Theorem 5:* Given a discrete planar array geometry, an elliptic vector velocity field  $\Psi(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  outside of  $\mathcal{M} = B_\epsilon \times S_1$  for some  $\epsilon > 0$ , and  $\mu_S < \mu_R$ , the solution to the governing equations given by the PDM is exponentially stabilizable.  $\diamond$

**Proof:** We explicitly show that the requirements of Theorem 1 are satisfied. First, we show that the system is locally exponentially stable, and then we use Corollary 4 to show that the exponential stability can be preserved using the programmable vector field outside a small compact set. First choose the Lyapunov function to be  $V(x) = \frac{1}{2}\|x\|^2$ . Moreover, choose  $V_0 = V_1 = W = V$  (this choice merely takes advantage of the fact that the governing equations are first order). Clearly  $V_0 \leq V \leq V_1$ . Now we must show that  $\dot{V}^* \leq -W$ , where  $\dot{V}^*$  was defined in Eq. (8). First we must compute  $\dot{V}^*$ .

Recall the governing equations from Eq. (6). Then, taking  $V(x, y, \theta) = \frac{1}{2}\|x\|^2 = \frac{1}{2}(x^2 + y^2 + \theta^2)$  we see that in

a region where only one set of governing equations apply (i.e.  $\dot{V}^* = \dot{V}$ )

$$\begin{aligned}\dot{V} &= x\dot{x} + y\dot{y} + \theta\dot{\theta} \\ &= \frac{u_j(\theta - yx_i + xy_i)}{\sin(\theta_j)(x_i - x_j) + \cos(\theta_j)(-y_i + y_j)} \\ &+ \frac{u_i(\cos(\theta_j)(y \sin(\theta_i)(y_i - y_j) - \cos(\theta_i)(\theta - yx_i + xy_j))}{\sin(\theta_j)(x_i - x_j) + \cos(\theta_j)(-y_i + y_j)} \\ &+ \frac{u_i(\sin(\theta_j)(x \cos(\theta_i)(-x_i + x_j) - \sin(\theta_i)(\theta - yx_j + xy_i))}{\sin(\theta_j)(x_i - x_j) + \cos(\theta_j)(-y_i + y_j)}\end{aligned}$$

or

$$\dot{V} = u_i f_i(x) + u_j f_j(x) \quad (9)$$

Note that this is split into two coefficients of  $u_i$  and  $u_j$ , which in turn implies that if we have full knowledge of the state, then we can always choose the inputs so as to make  $\dot{V}^* = \dot{V} \leq 0$ . Moreover,  $\dot{V}$  can be made to be always nonzero through a proper choice of inputs. Due to the form of Eq.(9), inputs  $u_i$  and  $u_j$  can always be chosen so that

$$\dot{V} \leq -V.$$

Now we have to check on the boundary between two different contact states. On the boundary

$$\dot{V}^* = \sup\left\{\frac{\partial V}{\partial q}\dot{q}_i, \frac{\partial V}{\partial q}\dot{q}_j\right\}$$

where

$$\frac{\partial V}{\partial q}\dot{q}_i = u_{i1}f_{i1} + u_{i2}f_{i2}$$

$$\frac{\partial V}{\partial q}\dot{q}_j = u_{j1}f_{j1} + u_{j2}f_{j2}$$

subject to the constraint that  $u_{j1} = u_{i2}$ . Substituting and comparing, we get

$$\dot{V}^* = \sup\{u_{i1}f_{i1} + u_{i2}f_{i2}, u_{i2}f_{j1} + u_{j2}f_{j2}\}$$

thus implying that despite the constraint both can be made arbitrarily negative, thus ensuring that

$$\dot{V}^* \leq -V.$$

Hence, the system is locally exponentially stabilizable.

Now, we will show that exponential stability can be maintained outside of  $\mathcal{M}$ . We already know from Corollary 4 that the origin is globally stable. All we need to show is that there is an exponential  $k_3\|x\|e^{-st}$  which provides an upper bound on  $V^*$ . From [13] we know that outside  $\mathcal{M}$  the  $x$  and  $y$  coordinates can be exponentially stabilized, but the  $\theta$  coordinate is only neutrally stable. Therefore, the maximum value of  $\|x, y, \theta\|$  is  $d(\partial\mathcal{M}, 0) + \pi^2$ , where  $d(\cdot, 0)$  is the maximum distance from the origin to the boundary of a set. Setting  $k_3 = d(\partial\mathcal{M}, 0) + \pi^2$  it is clear that outside of  $\mathcal{M}$  the solutions converge exponentially to  $\mathcal{M}$ , and inside  $\mathcal{M}$  we have already shown that the origin is exponentially stable. Therefore the origin is globally exponentially stable.  $\square$

## IV. Simulations

To illustrate these concepts, this section provides the results of simulations that model a planar distributed manipulation system with actuators located at  $(i, j)$  for  $i, j \in \mathbb{N}$ . Actuation is provided by unit radius rotating wheels (which rotate along axes orthogonal to the wheel rims depicted in Fig. 2) with constant friction coefficient  $\mu$  and point contact between the wheel rims and the manipulated object (a box in this case). The feedback region,  $\mathcal{M}$ , lies in the interior of a circle of radius 4 (in the length units used by the simulation, see Fig. 2), and the only four actuators inside  $\mathcal{M}$  are at  $(\pm 1, \pm 1)$ . Outside of  $\mathcal{M}$ , the actuators are located at  $(\pm i, \pm j)$  for  $i, j \geq 2$ . The four actuators inside the circle represent the four node system studied in [13]. The simulated task involves moving a unit mass box from  $\{3, 10, \frac{\pi}{4}\}$  to the origin in  $\mathbb{R}^2$ . The final configuration's orientation is stabilized to  $\theta = 0$ , where  $\theta$  is measured between the box's long axis and the  $x$  axis of  $\mathbb{R}^2$ .

The simulations were implemented in *Mathematica*, using its *NDSolve* integrator, modified to allow for differential inclusions. Some modification is necessary in order to avoid the numerical difficulties at switching boundaries ( $y = x, y = -x, x = 0, y = 0$  for these simulations). However, this is only a concern for switching boundaries which are stable or attracting, because if the trajectory intersects the boundary transversely, standard numerical schemes still work. For these simulations, we allowed the switching boundaries (which are stable) to have the averaged, projected dynamics. This, like the method of introducing hysteresis to simulate a discrete system as a hybrid automaton, produces numerically stable simulations. We should point out, however, that the choice of the averaged solution is only one possible choice satisfying the differential inclusion. That is, if we have a boundary  $N$  and vector fields  $g_{\sigma_1}$  on one side of the boundary and  $g_{\sigma_2}$  on the other, the choice of  $\dot{q} = \frac{g_{\sigma_1} + g_{\sigma_2}}{2}$  (i.e.  $\delta = \frac{1}{2}$ ) is just one choice satisfying  $\dot{q} \in F = \text{co}\{g_{\sigma_1}, g_{\sigma_2}\}$ . The main difficulty is that solutions of differential inclusions are necessarily nonunique, therefore implying that any simulation represents only one solution  $\phi$  to the differential inclusion  $\dot{\phi} \in F$ .

Outside  $\mathcal{M}$ , the programmable vector field is simply  $\{\dot{x}, \dot{y}\} = \{-x, -y\}$ . Inside  $\mathcal{M}$  we use the feedback law derived in [13] to stabilize the box to the origin. To show how to apply the theory, we will apply it near the boundary  $x = y$  for  $x, y > 0$  defined by the switching boundaries. Restricting our attention to this region provides all the salient features of the theory while keeping the problem tractable. Rescaling  $u_i$  by  $\frac{1}{\sqrt{2}}$ , the kinematics in this region are

$$\dot{q} = \begin{bmatrix} -u_1 + u_2 \\ -u_1 - u_2 \\ \text{co}\{-u_1, u_2\} \end{bmatrix}$$

Now we do a control lyapunov design using  $V =$

$\frac{1}{2}\|(x, y, \theta)\|^2$ . This means that  $\dot{V} = \frac{\partial V}{\partial q}\dot{q} = x(-u_1 + u_2) + y(-u_1 - u_2) + \theta(\text{co}\{-u_1, u_2\})$ . Rewriting  $\text{co}\{-u_1, u_2\}$  as  $\delta(-u_1) + (1 - \delta)u_2$  and thereby parametrizing all selections of  $\text{co}\{-u_1, u_2\}$  by  $\delta$ . Solving the equation  $\dot{V} = \frac{\partial V}{\partial q}\dot{q} = -k\|q\|^2$  (with  $k > 0$ ), we see that a choice of

$$\begin{aligned} u_1 &= \frac{-k\theta((1-\delta)\theta + x - y) + k\|(x, y, \theta)\|^2}{\delta\theta + x + y} \\ u_2 &= -k\theta \end{aligned}$$

makes  $\dot{V} = -k\|q\|^2$ . Therefore the system is exponentially stabilized. Now, this example should illustrate some of the weaknesses of the theory described in Section III, partially because estimation of  $\delta$  is necessary. In our numerical simulations we know that  $\delta = \frac{1}{2}$ , and the simulation works well. However, in a real system  $\delta$  would have to be estimated on-line. We are currently working on extending the work in [8] to our case so that the feedback law will not be sensitive to variations in  $\delta$ , and therefore not as sensitive to effects of chattering, etcetera. The simulation shown in Figure 2 shows snapshots of the box's position at integer time units  $t = 1, \dots, 12$ . The actual box's boundary is larger than the one shown in the figure. This was done so that the simulation could be visualized more easily.

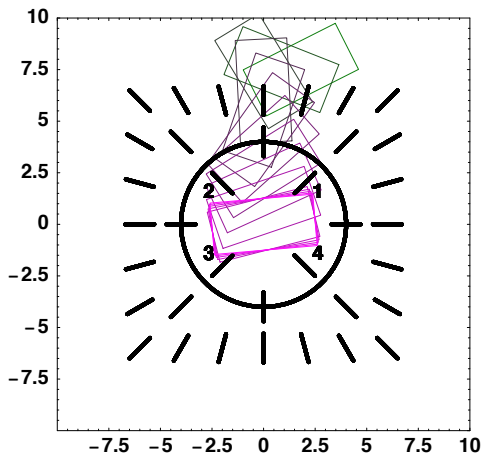


Fig. 2. A Box Transported to  $\{x, y, \theta\} = \{0, 0, 0\}$  from  $\{3, 10, \frac{\pi}{4}\}$

## V. Conclusions

In [12] we found that the locally stabilizing aspect of the controller obtained in [13] can be successfully used in conjunction with the global method of programmable vector fields to provide global stability without requiring global feedback. Here we provided a proof that exponential convergence can also be obtained in the absence of global feedback. However, it is clear that a more general and robust framework is needed to produce practical feedback stabilization. Our current work includes building an experimental setup using wheeled actuators and visual feedback to validate the theory provided both in this and future work.

**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.F. Böhringer, B. Donald, and N. MacDonald. Upper and lower bounds for programmable vector fields with applications to mems and vibratory plate parts feeders. In *Algorithms for Robotic Motion and Manipulation*, pages 255–276. A.K. Peters, Ltd., 1997.
- [3] K.F. Böhringer, B.R. Donald, L.E. Kavraki, and F. Lamiroux. A distributed, universal device for planar parts feeding: unique part orientation in programmable force fields. In *Distributed Manipulation*, pages 1–28. Kluwer Academic Pub., 2000.
- [4] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4), 1988.
- [5] A.F. Filippov. *Differential Equations with Discontinuous Right Hand Sides*. Kluwer Academic Publishers, 1988.
- [6] K.Y. Goldberg. Orienting polygonal parts without sensing. *Algorithmica*, 143(2/3/4):201–225, 1993.
- [7] S. Goyal, A. Ruina, and J. Papadopoulos. Planar sliding with dry friction. *WEAR*, 143:307–352, 1991.
- [8] João P. Hespanha and A. Stephen Morse. Stability of switched systems with average dwell-time. Technical report, EE-Systems, University of Southern California, 1999.
- [9] Hassan K. Khalil. *Nonlinear Systems (second edition)*. Prentice Hall, 1996.
- [10] J.P. LaSalle. Stability theory for ordinary differential equations. *J. Diff. Eq.*, 4:57–65, 1968.
- [11] T. D. Murphey and J. W. Burdick. Issues in controllability and motion planning for overconstrained wheeled vehicles. In *Proc Int Conf Math Theory Networks Systems (MTNS)*, Perpignan, France, 2000.
- [12] T. D. Murphey and J. W. Burdick. Global stability for distributed systems with changing contact states. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Hawaii, 2001.
- [13] T. D. Murphey and J. W. Burdick. On the stability and design of distributed systems. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.
- [14] A. Sudsang and L. Kavraki. A geometric approach to designing a programmable force field with a unique stable equilibrium for parts in the plane. In *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, 2001.