Stable Pushing: Mechanics, Controllability, and Planning

Kevin M. Lynch and Matthew T. Mason

The International Journal of Robotics Research 1996 15: 533
DOI: 10.1177/027836499601500602

Stable Pushing: Mechanics, Controllability, and Planning

Kevin M. Lynch
Matthew T. Mason
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

We would like to give robots the ability to position and orient parts in the plane by pushing, particularly when the parts are too large or heavy to be grasped and lifted. Unfortunately, the motion of a pushed object is generally unpredictable due to unknown support friction forces. With multiple pushing contact points, however, it is possible to find pushing directions that cause the object to remain fixed to the manipulator. These are called stable pushing directions. In this article we consider the problem of planning pushing paths using stable pushes. Pushing imposes a set of nonholonomic velocity constraints on the motion of the object, and we study the issues of local and global controllability during pushing with point contact or stable line contact. We describe a planner for finding stable pushing paths among obstacles, and the planner is demonstrated on several manipulation tasks.

1. Introduction

One of the most basic tasks for a robotic manipulator is to move an object from one place to another. A common solution is to equip the manipulator with a gripper and adopt the pick-and-place approach. By designing the grasp to resist all forces that could reasonably act on the object during the motion, grasp planning and path planning can be decoupled.

If the object is too large to be grasped or too heavy to be carried, however, this approach fails. It underutilizes the resources available to the robot, as it uses only the control forces that can be statically applied at the gripper. In general, the manipulator can apply forces through any of the frictional kinematic constraints that comprise it. Other useful sources of control forces include gravity, the frictional kinematic constraints (floor, walls, obstacles) making up the robot's environment, and dynamic forces. If the robot can reason about these forces, it can use a richer set of manipulation primitives, including pushing, throwing, and striking. One emphasis of our research is to study how the set of achievable tasks grows as we give the robot a better understanding of mechanics.

In this article we examine pushing, which provides a simple and practical solution to the problem of positioning and orienting objects in the plane, particularly when the manipulator lacks the size, strength, or dexterity to grasp and lift them. Because the object is not firmly grasped, however, the forces that can be applied are limited, and therefore the possible motions of the object are limited. Unlike pick and place, the "grasp" (pushing contact) and manipulator path cannot be decoupled. By modeling the support friction forces, however, we can simultaneously design the pushing contact and manipulator path such that the contact resists all expected forces during the motion.

This problem is made difficult by the indeterminacy of the distribution of support forces of the pushed object. The precise motion of a pushed object is usually unpredictable. If there are two or more pushing points, however, there may exist a space of pushing directions that admit only a single solution to the motion of the object: the motion that causes the object to maintain its configuration relative to the pusher. The object is effectively rigidly attached to the pusher, and the push is called a *stable push* (Lynch 1992). A pushing path is formed by stringing together stable pushes, as in Figure 1.

Our goal is to develop algorithms to automatically find pushing plans to position and orient parts in the plane. Toward this end, in this article we study the following three issues in pushing:

1. *Mechanics.* How does an object move when it is pushed? We describe a procedure that identifies a set

The International Journal of Robotics Research,
Vol. 15, No. 6, December 1996, pp. 533–556,
© 1996 Massachusetts Institute of Technology.

An earlier version of this article was presented at the First Workshop on the Algorithmic Foundations of Robotics (WAFR), 1994, and portions of this material appeared in the *Proceedings of the 1995 IEEE International Conference on Robotics and Automation.*

Kevin Lynch is currently located at: Biorobotics Division, Mechanical Engineering Laboratory, Namiki 1-2, Tsukuba, Ibaraki 305 Japan.

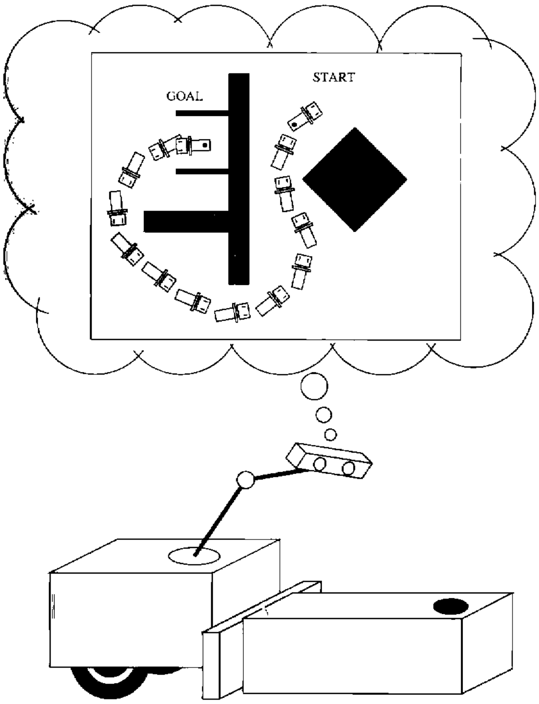

Fig. 1. A mobile robot pushing a box using stable pushes with line contact.

of stable pushing directions when the pusher makes line contact with the object.

2. *Controllability.* The directions an object can move during pushing are limited due to the limited set of forces that can be applied by the pusher. Given these limitations, our study of controllability is motivated by questions of whether or not it is possible to push the object to the goal configuration, with and without obstacles. We examine the local and global controllability of objects pushed with either point contact or stable line contact.
3. *Planning.* Pushing paths consist of sequences of stable pushes, and the space of stable pushing directions imposes nonholonomic constraints on the motion of the object. We draw on work on path planning for nonholonomic mobile robots by Barraquand and Latombe (1993) to construct a planner to find stable pushing paths among obstacles.

1.1. Related Work

1.1.1. Pushing

Mason (1986) identified pushing as an important manipulation process for manipulating several objects at once, for reducing uncertainty in part orientation, and as a precursor to grasping. Building on early work by Prescott (1923) and MacMillan (1936), Mason implemented a numerical routine to find the motion of an object with a known support distribution being pushed at a single point of contact. Recognizing that the support distribution is usually unknown, Mason derived a simple rule for determining the rotation sense of the pushed object that depends only on the center of mass of the object. Mason and Brost (1986) and Peshkin and Sanderson (1988b) followed this work by finding bounds on the rotation rate of the pushed object. Goyal, Ruina, and Papadopoulos (1991a,b) studied the relationship between the motion of the sliding object and the associated support friction when the support distribution is completely specified. Alexander and Maddocks (1993) considered the other extreme, when only the geometric extent of the support area is known, and described techniques to bound the possible motions of the pushed object.

These results have been used to plan manipulator pushing and grasping operations. Mason (1986) used pushing and grasping to reduce uncertainty. Mani and Wilson (1985) built a system for orienting a part in an initially unknown orientation by executing a series of linear pushes with a fence. Peshkin and Sanderson (1988a) and Brokowski et al. (1993) considered a similar problem where the part is carried on a conveyor belt and reoriented by interactions with fences suspended above the belt. Brost (1988) developed algorithms to identify stable parallel-jaw grasping motions for polygonal objects in the presence of uncertainty, and Goldberg (1993) developed a planner to find a sequence of parallel-jaw grasps to orient a polygonal part. Brost (1992) has also shown how to find the linear pushing motions resulting in a desired pusher/object equilibrium configuration. This is like "catching" the object by pushing it. Balorda (1990; 1993) has investigated catching by pushing with two points of contact. Mason (1989) has shown how to synthesize robot pushing motions to slide a block along a wall, a problem later studied by Mayeda and Wakatsuki (1991), who considered pushing forces out of the plane.

Feedback control of the motion of an object pushed with a single point of contact has been studied by many researchers, including Inaba and Inoue (1989), Gandolfo et al. (1991), Lynch et al. (1992), Okawa and Yokoyama (1992), and Salganicoff et al. (1993a). A control strategy for pushing by two cooperating mobile robots is described by Donald et al. (1993). Learning (Miura 1989; Zrimec 1990; Salganicoff et al. 1993b) and friction parameter estimation (Yoshikawa and Kurisu 1991; Lynch 1993) have also been proposed to improve control.

Particularly relevant to the topic of this article is work by Akella and Mason (1992), Narasimhan (1995), and Kurisu and Yoshikawa (1994). Akella and Mason have considered the problem of planning pushing sequences to reconfigure polygonal parts in the obstacle-free plane.

Each pusher motion is a linear motion, and the object can rotate without slipping on the straight edge of the pusher. The precise motion of the object during the push is unknown, but at the end of each push, a known edge of the object will be aligned with the pusher. Between pushes, the pusher breaks contact and recontacts the object. The approach is guaranteed to find an open-loop plan for any polygonal object and any initial and goal configuration.

Narasimhan (1995) and Kurisu and Yoshikawa (1994) have studied the problem of moving an object among obstacles by pushing with point contact. In Narasimhan's work, a holonomic path is found connecting the start and goal configurations, and at each step the robot chooses the push that will most likely keep the object close to the path. This choice is based on a model built from simulated or experimental data. Kurisu and Yoshikawa use optimal control techniques to solve for the entire manipulator trajectory in advance, using an estimate of the object's support distribution.

1.1.2. Nonprehensile Manipulation

Pushing is a type of graspless or *nonprehensile* manipulation. Nonprehensile manipulation exploits the mechanics of the task to achieve a goal state without grasping, allowing simple mechanisms to accomplish complex tasks. Other examples of nonprehensile manipulation include tumbling (Sawasaki et al. 1989) and pivoting (Aiyama et al. 1993) objects on a support surface; whole-arm manipulation (Salisbury et al. 1987; Trinkle et al. 1993); and positioning planar parts by tray-tilting (Erdmann and Mason 1988; Christiansen 1995), moving frictionless pins (Abell and Erdmann 1995), and striking (Higuchi 1985; Huang et al. 1995). Striking uses dynamic forces to achieve the goal; other examples of dynamic nonprehensile manipulation include juggling (Rizzi and Koditschek 1993; Schaal and Atkeson 1993; Zumel and Erdmann 1994), throwing (Mason and Lynch 1993; Lynch and Mason 1996), and rolling (Arai and Khatib 1994; Lynch and Mason 1996). See Mason and Lynch (1993) for other references.

1.1.3. Sufficiency Results in Manipulation

One of the primary contributions of this article is a characterization of the sufficiency of pushing for repositioning parts in the plane. Related results in manipulation are bounds on the number of point fingers necessary for grasping (Mishra et al. 1987; Markenscoff et al. 1990), the demonstration of the controllability of a ball rolling on a plane or another ball (Li and Canny 1990), and the classification of orientable parts by sensorless parallel-jaw grasping sequences (Goldberg 1993). Goldberg (1995) provides an interesting discussion on sufficiency and completeness results in robot motion planning. The results in the present article draw on ideas of controllability from nonlinear control theory; a good introduction is given by Nijmeijer and van der Schaft (1990).

1.2. Assumptions

1. Friction is assumed to conform to Coulomb's law. The frictional force at a sliding contact opposes the motion with magnitude $\mu_k f_n$, where f_n is the magnitude of the normal contact force and μ_k is the kinetic coefficient of friction. At a sticking contact, the frictional force can act in any tangential direction with any magnitude less than or equal to μf_n, where μ is the static coefficient of friction. For simplicity, we will assume that the static and kinetic coefficients of friction are equal.
2. All pushing forces lie in the horizontal support plane, and gravity acts along the vertical.
3. The pusher and slider move in the horizontal plane.
4. Friction properties are uniform over the support plane.
5. Pushing motions are slow enough that inertial forces are negligible. This is the quasistatic assumption. Pushing forces are always balanced by the support frictional forces acting on the object.

1.3. Definitions

The slider $\mathcal{S}$ is a rigid object in the plane $\mathcal{W} = \mathbf{R}^2$, and its configuration space $\mathcal{C}$ is $\mathbf{R}^2 \times S^1$. The slider is pushed by a rigid pusher $\mathcal{P}$ at a point or set of points on a closed, piecewise smooth curve Γ, which typically forms the perimeter of the slider $\mathcal{S}$. A world frame $\mathcal{F}_{\mathcal{W}}$ with origin $O_{\mathcal{W}}$ is fixed in the plane, and a slider frame $\mathcal{F}_{\mathcal{S}}$ with origin $O_{\mathcal{S}}$ is attached to the center of friction of the slider $\mathcal{S}$. (For a uniform coefficient of support friction, the center of friction of the slider is the point in the support plane beneath the center of mass [MacMillan 1936].) Configurations measured in the slider frame $\mathcal{F}_{\mathcal{S}}$ have coordinates $(x, y, \theta)^T$. The configuration $\mathbf{q} = (x_w, y_w, \theta_w)^T$ describes the position and orientation of the slider frame $\mathcal{F}_{\mathcal{S}}$ relative to the world frame $\mathcal{F}_{\mathcal{W}}$ (Fig. 2).

Generalized forces $\mathbf{f}$ (wrenches) and velocities $\mathbf{v}$ (twists) are always defined with respect to the slider frame $\mathcal{F}_{\mathcal{S}}$. A force $\mathbf{f} \in \mathbf{R}^3$ is given by its force and moment components $(f_x, f_y, m)^T$. A nonzero force $\mathbf{f}$ is the product of its magnitude f and its direction $\hat{\mathbf{f}}$. A force direction is a three-dimensional unit vector and may be represented as a point on the unit sphere ($\hat{\mathbf{f}} \in S^2$). The

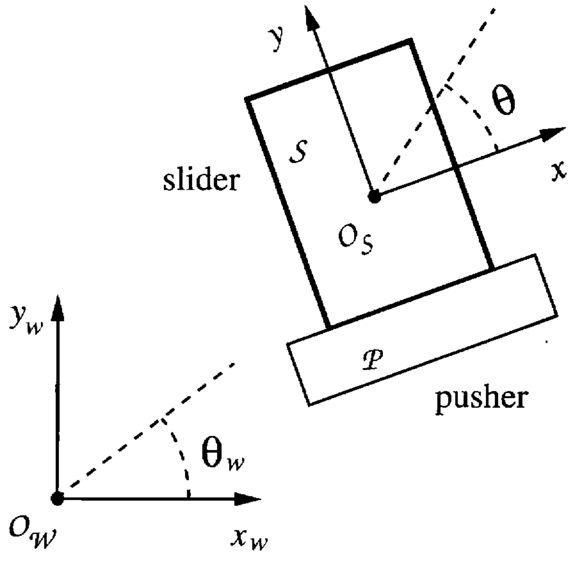

Fig. 2. The world frame $\mathcal{F}_{\mathcal{W}}$ and the slider frame $\mathcal{F}_{\mathcal{S}}$.

sphere of force directions is called the *force sphere*. Similarly, a nonzero velocity $\mathbf{v} = (v_x, v_y, \omega)^T$ is given by the product of its magnitude v and direction $\hat{\mathbf{v}}$, and the sphere of velocity directions is called the *velocity sphere*. We will sometimes represent a velocity direction by its center of rotation in $\mathcal{F}_{\mathcal{S}}$. The mapping COR($\cdot$) maps velocity directions to rotation centers in the slider frame $\mathcal{F}_{\mathcal{S}}$, such that COR($\hat{\mathbf{v}}$) returns the point about which the velocity direction $\hat{\mathbf{v}}$ is a pure rotation, along with the sense of rotation. The domain of the function COR($\cdot$) is the velocity sphere, and the range consists of two copies of the plane, one for each rotation sense, and a line at infinity for translations. Figure 3 illustrates the mapping from velocity directions to rotation centers.

For the quasistatic pushing problem, we are concerned only with force and velocity directions, not with their magnitudes. We assume only that the manipulator is strong enough to move the slider, and that it moves slowly enough to satisfy the quasistatic assumption. A pushing plan generated by the planner in Section 4 may be properly thought of as a pushing path, not a trajectory. To generate a manipulator trajectory from this path, times must be assigned to each point along the path such that the quasistatic assumption is satisfied.

1.4. Overview

In the next section we define the pushing control system, some basic definitions of controllability, and their application to the pushing control system. Armed with these tools, in Section 3 we study the mechanics of pushing and the controllability of objects pushed with either point contact or stable line contact. Finally, Section 4 demonstrates a planning algorithm for repositioning objects among obstacles using stable pushes.

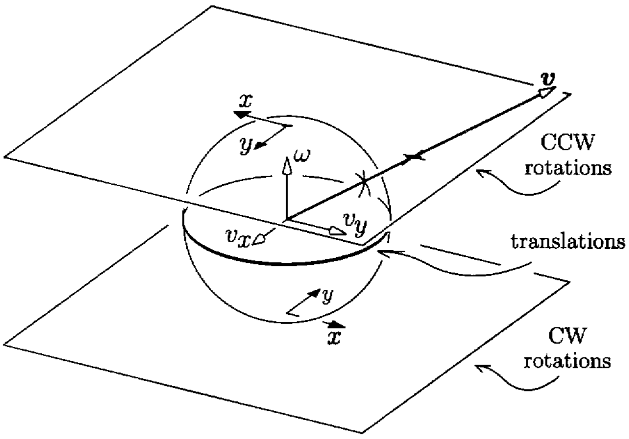

Fig. 3. The mapping COR($\cdot$) from velocity directions on the unit sphere to rotation centers in the slider frame $\mathcal{F}_{\mathcal{S}}$.

2. Controllability with Velocity Constraints

The set of velocity directions that the slider can follow during pushing is limited due to the limited set of force directions that can be applied by the pusher. These limitations constitute a set of nonholonomic constraints: constraints on the velocity of the slider that cannot be integrated to give configuration constraints. For example, a slider that can be pushed in one direction cannot be pulled in the opposite direction by simply reversing the motion of the pusher. Despite these constraints on the motion, we know by experience that it is often possible to move objects to desired configurations by pushing. In this section we formalize these ideas using tools from nonlinear control theory. We defer the problem of determining the motion of a pushed object to Section 3.

2.1. The Pushing Control System

The pushing control system can be described abstractly by the autonomous nonlinear control system $\dot{\mathbf{q}} = F(\mathbf{q}, \mathbf{c})$, where $\mathbf{c}$ is the control input describing the pushing contact configuration and the velocity of the pusher in the slider frame $\mathcal{F}_{\mathcal{S}}$. The motion of the slider in the world frame $\mathcal{F}_{\mathcal{W}}$ is a function F of the control input and the configuration of the slider. For the rest of this article, we will use the following more concrete description of the control system Σ:

$$\Sigma: \quad \dot{\mathbf{q}} = F(\mathbf{q}, \mathbf{c}_u) = X_u(\mathbf{q}),$$

$$\mathbf{q} \in \mathcal{C} = \mathbf{R}^2 \times S^1, \quad u \in \{0, \ldots, n\},$$

$$X_u(\mathbf{q}) = \begin{cases} (0,\ 0,\ 0)^T & \text{if } u = 0 \\ \begin{pmatrix} \cos\theta_w & -\sin\theta_w & 0 \\ \sin\theta_w & \cos\theta_w & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{v}_{ux} \\ \hat{v}_{uy} \\ \hat{\omega}_u \end{pmatrix} & \text{otherwise.} \end{cases}$$

A nonzero u chooses one of n distinct combinations of contact configurations and pushing velocities in the slider frame $\mathcal{F}_S$. Associated with each control $\mathbf{c}_u$ is a vector field X_u describing the motion of the slider in the world frame $\mathcal{F}_\mathcal{W}$. The tangent vector $X_u(\mathbf{q})$ is the (unit) velocity of the slider in the world frame $\mathcal{F}_\mathcal{W}$, and $\hat{\mathbf{v}}_u$ is the (unit) velocity of the slider in the slider frame $\mathcal{F}_S$. The set of nonzero vector fields X_u is denoted $\mathcal{X}$, and the set of nonzero velocity directions $\hat{\mathbf{v}}_u$ in the slider frame $\mathcal{F}_S$ is $\hat{\mathcal{V}}$. Each of the n nonzero controls results in a distinct velocity direction.

Three aspects of the control system Σ bear mentioning:

1. The absence of a drift vector field (a vector field that is only a function of the state of the slider) implies that the slider will not move when it is not pushed ($u = 0$). This is a consequence of the quasistatic assumption: the slider's kinetic energy is instantly dissipated by support friction, and the slider's state is merely its configuration.
2. For any constant control, the slider's velocity direction is constant in the slider frame $\mathcal{F}_S$.
3. The control system is not necessarily symmetric. In general, it is not possible to follow a vector field X_u backward. This is due to the unilateral nature of frictional contact: pushing forces must have a nonnegative component in the direction of the contact normal.

2.2. *Definitions of Controllability*

The pushing control system Σ, or equivalently the configuration of the slider $\mathcal{S}$, is *controllable from* $\mathbf{q}$ if, starting from $\mathbf{q}$, the slider can reach any point in the configuration space $\mathcal{C}$. The slider is *small-time locally controllable from* $\mathbf{q}$ if, for any neighborhood U of $\mathbf{q}$, the set of reachable configurations without leaving U contains a neighborhood of $\mathbf{q}$. The slider is *accessible from* $\mathbf{q}$ if the set of reachable configurations from $\mathbf{q}$ has nonempty interior in $\mathcal{C}$. The slider is *small-time accessible from* $\mathbf{q}$ if, for any neighborhood U of $\mathbf{q}$, the set of reachable configurations without leaving U has nonempty interior. For autonomous linear systems, all of these concepts are equivalent (Hermann and Krener 1977).

(Although the phrase "small-time" appears in these terms, time does not appear in their definitions as they are applied in this article. Other controllability definitions are given by Haynes and Hermes [1970], Sussmann and Jurdjevic [1972], Hermann and Krener [1977], Sussmann [1978; 1983], and Nijmeijer and van der Schaft [1990]. These definitions are not always consistent with each other. Sussmann [1983] provides the most complete set of definitions.)

If a controllability property holds for all $\mathbf{q} \in \mathcal{C}$, the phrase "from $\mathbf{q}$" can be omitted. For the control system Σ, any property that holds for any $\mathbf{q} \in \mathcal{C}$ holds for all $\mathbf{q} \in \mathcal{C}$. Similarly, any property that does not hold for some $\mathbf{q} \in \mathcal{C}$ does not hold for any $\mathbf{q} \in \mathcal{C}$.

By definition, controllability implies accessibility, and small-time local controllability implies small-time accessibility. By the connectivity of the configuration space $\mathcal{C}$, small-time local controllability implies controllability. Small-time local controllability ensures that the slider can follow any free path arbitrarily closely.

Of these properties, small-time accessibility can be established by an algebraic test on the set of vector fields $\mathcal{X}$. The Lie algebra $L(\mathcal{X})$ of the vector fields $\mathcal{X}$ is the space of linear combinations of these vector fields and the vector fields created by repeated Lie bracket operations. The Lie bracket of the vector fields X and Y is denoted $[X, Y]$. Defining $B_0(\mathcal{X}) = \mathcal{X}$ and $B_{k+1}(\mathcal{X}) = B_k(\mathcal{X}) \cup \{[X, Y] \text{ for all } X, Y \in B_k(\mathcal{X})\}$, the Lie algebra $L(\mathcal{X})$ is spanned by vector fields in $B_\infty(\mathcal{X})$. A control system is small-time accessible from $\mathbf{q}$ if it satisfies the *Lie Algebra Rank Condition*, which states that the tangent vectors at $\mathbf{q}$ of vector fields in $L(\mathcal{X})$ must span the tangent space at $\mathbf{q}$.

The Lie bracket $[X, Y]$ of the vector fields X and Y in local coordinates is

$$[X, Y](\mathbf{q}) = \frac{\partial Y(\mathbf{q})}{\partial \mathbf{q}} X(\mathbf{q}) - \frac{\partial X(\mathbf{q})}{\partial \mathbf{q}} Y(\mathbf{q}).$$

(Nijmeijer and van der Schaft [1990] provide a derivation.) For the pushing control system Σ, $\partial X(\mathbf{q})/\partial \mathbf{q}$ is given by

$$\begin{pmatrix} \partial X_{x_w}(\mathbf{q})/\partial x_w & \partial X_{x_w}(\mathbf{q})/\partial y_w & \partial X_{x_w}(\mathbf{q})/\partial \theta_w \\ \partial X_{y_w}(\mathbf{q})/\partial x_w & \partial X_{y_w}(\mathbf{q})/\partial y_w & \partial X_{y_w}(\mathbf{q})/\partial \theta_w \\ \partial X_{\theta_w}(\mathbf{q})/\partial x_w & \partial X_{\theta_w}(\mathbf{q})/\partial y_w & \partial X_{\theta_w}(\mathbf{q})/\partial \theta_w \end{pmatrix},$$

where

$$X = (X_{x_w}, X_{y_w}, X_{\theta_w})^T.$$

Using the definition of X_u from Section 2.1, $\partial X_u(\mathbf{q})/\partial \mathbf{q}$ evaluates simply to

$$\begin{pmatrix} 0 & 0 & -\hat{v}_{ux}\sin\theta_w - \hat{v}_{uy}\cos\theta_w \\ 0 & 0 & \hat{v}_{ux}\cos\theta_w - \hat{v}_{uy}\sin\theta_w \\ 0 & 0 & 0 \end{pmatrix}.$$

For the control system Σ, the Lie algebra $L(\mathcal{X})$ is spanned by vector fields in $B_1(\mathcal{X})$: the distribution defined by $B_1(\mathcal{X})$ is involutive. We need only look at the vector fields $\mathcal{X}$ and their Lie brackets to decide small-time accessibility.

If the control system is symmetric (all vector fields can be followed forward and backward), then small-time accessibility implies small-time local controllability. Sussmann (1987) proves this and other sufficient conditions for small-time local controllability.

2.3. Controllability of the Pushing Control System

If $n = 1$ for the control system Σ, then the slider is confined to a one-dimensional integral curve of X_1, and the control system Σ is not accessible. If $n = 2$, the Lie algebra $L(\mathcal{X})$ is spanned by X_1, X_2, and $X_3 = [X_1, X_2]$, where

$$\begin{aligned}
X_1 &= (\hat{v}_{1x}\cos\theta_w - \hat{v}_{1y}\sin\theta_w, \hat{v}_{1x}\sin\theta_w + \hat{v}_{1y}\cos\theta_w, \hat{\omega}_1)^T,\\
X_2 &= (\hat{v}_{2x}\cos\theta_w - \hat{v}_{2y}\sin\theta_w, \hat{v}_{2x}\sin\theta_w + \hat{v}_{2y}\cos\theta_w, \hat{\omega}_2)^T,\\
X_3 &= [X_1, X_2] = (\hat{\omega}_1(-\hat{v}_{2x}\sin\theta_w - \hat{v}_{2y}\cos\theta_w)\\
&\quad +\hat{\omega}_2(\hat{v}_{1x}\sin\theta_w + \hat{v}_{1y}\cos\theta_w),\\
&\quad \hat{\omega}_1(\hat{v}_{2x}\cos\theta_w - \hat{v}_{2y}\sin\theta_w)\\
&\quad +\hat{\omega}_2(-\hat{v}_{1x}\cos\theta_w + \hat{v}_{1y}\sin\theta_w), 0)^T.
\end{aligned}$$

The dimension of $L(\mathcal{X})$ is given by rank($X_1 \;\; X_2 \;\; X_3$). If the determinant of this matrix is nonzero, then its rank is 3. A simple calculation yields

$$\det(X_1 \;\; X_2 \;\; X_3) = (\hat{\omega}_2\hat{v}_{1x} - \hat{\omega}_1\hat{v}_{2x})^2 + (\hat{\omega}_2\hat{v}_{1y} - \hat{\omega}_1\hat{v}_{2y})^2.$$

The determinant is zero only if (1) $\hat{\omega}_1 = \hat{\omega}_2 = 0$ or (2) $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ are multiples of each other. (Since $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ are distinct unit vectors, this condition is equivalent to $\hat{\mathbf{v}}_1 = -\hat{\mathbf{v}}_2$.) If condition (1) holds, the slider cannot rotate. If condition (2) holds, the slider is confined to a one-dimensional curve of the configuration space $\mathcal{C}$. If neither of these conditions hold, the Lie bracket operation has essentially created a new linearly independent control vector field and the Lie Algebra Rank Condition is satisfied.

Proposition 1. The control system Σ is small-time accessible if and only if the set of velocity directions $\hat{\mathcal{V}}$ contains two velocity directions, $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$, such that they are not both translations ($\hat{\omega}_1 \neq 0$ or $\hat{\omega}_2 \neq 0$) and $\hat{\mathbf{v}}_1 \neq -\hat{\mathbf{v}}_2$.

Remark: As noted earlier, small-time accessibility implies accessibility on the configuration space $\mathcal{C}$. Here we note that any control system Σ that is accessible must also be small-time accessible. Thus, the conditions of Proposition 1 are also necessary and sufficient for accessibility.

Proposition 1 is a straightforward generalization of a result due to Barraquand and Latombe (1993) that states that the Lie Algebra Rank Condition is satisfied for any car-like mobile robot that can take at least two steering angles. A car-like mobile robot can drive both forward and backward, and this symmetry, coupled with small-time accessibility, implies small-time local controllability. As we have already noted, the pushing control system Σ may not be symmetric, so small-time local controllability does not follow from small-time accessibility.

For some systems, however, accessibility may imply controllability even when the system is not small-time locally controllable (Jurdjevic 1972; Jurdjevic and Sussmann 1972; Sussmann 1983). Consider the task of setting the minute hand of a watch if it can be rotated only in a clockwise direction. The configuration of the minute hand is not small-time locally controllable, but the topology of its configuration manifold S^1 renders the minute hand's configuration controllable.

For control systems Σ, it is easily shown that accessibility implies controllability on the configuration space $\mathcal{C}$, and the conditions of Proposition 1 are also necessary and sufficient for controllability.

Proposition 2. For the control system Σ, accessibility implies controllability. The slider $\mathcal{S}$ may be moved from any configuration to any other configuration in the obstacle-free plane if and only if the set of velocity directions $\hat{\mathcal{V}}$ contains two velocity directions, $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$, such that they are not both translations ($\hat{\omega}_1 \neq 0$ or $\hat{\omega}_2 \neq 0$) and $\hat{\mathbf{v}}_1 \neq -\hat{\mathbf{v}}_2$.

Proof: Accessibility is a necessary condition for controllability. The following argument shows that it is also sufficient. First consider the case $\hat{\omega}_1 \neq 0$, $\hat{\omega}_2 = 0$. The slider may reach any configuration $\mathbf{q}_{goal}$ from any other configuration $\mathbf{q}_{init}$ by following X_1, then X_2, then X_1. The rotation center COR($\hat{\mathbf{v}}_1$) is located at a point $R_1 = (-\hat{v}_{1y}/\hat{\omega}_1, \hat{v}_{1x}/\hat{\omega}_1)$ in the slider frame $\mathcal{F}_S$. We define a new frame $\mathcal{F}'_S$ with its origin at R_1. The frame $\mathcal{F}'_S$ is aligned with and fixed in the slider frame $\mathcal{F}_S$. In the frame $\mathcal{F}'_S$, the two velocity directions are a pure rotation $\hat{\mathbf{v}}'_1 = (0, 0, sgn(\hat{\omega}_1))^T$ and a pure translation $\hat{\mathbf{v}}'_2 = \hat{\mathbf{v}}_2$, and the problem is to transfer the frame $\mathcal{F}'_S$ from $\mathbf{q}'_{init}$ to $\mathbf{q}'_{goal}$. This is achieved by simply rotating the frame $\mathcal{F}'_S$ in place, translating it to the final position, and rotating it to the final orientation. Any of these steps could have zero length. An example is shown in Figure 4.

If both $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$ have nonzero angular components, with rotation centers at R_1 and R_2 in the slider frame $\mathcal{F}_S$, respectively, controllability can be demonstrated by showing that the slider can always translate to a configuration from which it can rotate to the goal. A translation is obtained by following X_1 and X_2 such that the total rotation is an integral multiple of 2π. The set of paths that follow X_1 and then X_2 and satisfy this condition defines a circle of final positions of the origin O_S of the

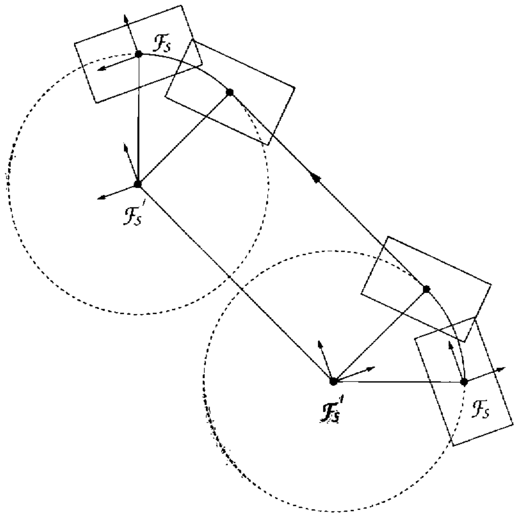

Fig. 4. An example path using one rotational and one translational velocity direction. The frames $\mathcal{F}_S$ and $\mathcal{F}_S'$ are drawn at the initial and final locations.

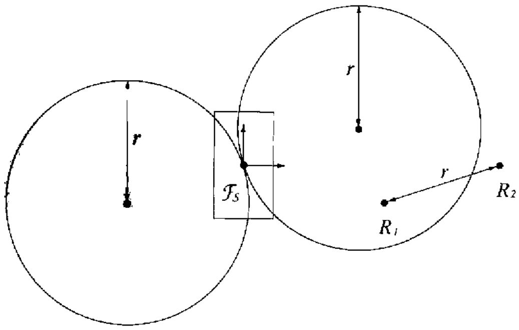

Fig. 5. The locus of points to which the slider frame $\mathcal{F}_S$ can translate, using one control change and rotation centers R_1 and R_2 in the slider frame $\mathcal{F}_S$. The left circle results from rotating about R_1 and then R_2 in the slider frame $\mathcal{F}_S$, and the right circle is obtained by reversing the order.

slider frame $\mathcal{F}_S$. The radius r of the circle is the distance between R_1 and R_2, and the center of the circle is a distance r from O_S on a ray from O_S parallel to the ray from R_2 through R_1. Translations consisting of paths following X_2 and then X_1 define a similar circle (Fig. 5). The locus of points to which the slider frame $\mathcal{F}_S$ can translate with only a single control change is given by these two circles. Each point on the locus is the origin of a similar, translated locus, and the slider frame $\mathcal{F}_S$ can translate to any point in the plane by concatenating translations with one control change. □

Corollary 1 follows immediately.

COROLLARY 1. The number of distinct combinations n of pushing contact configurations and pushing directions must be at least two for the configuration of the slider S to be controllable by pushing. From any set of controls yielding controllability, no more than two are needed.

For a controllable system Σ with two velocity directions ($n = 2$), the slider may have to travel a long distance to reach nearby configurations. Thus, the conditions of Proposition 2 are not sufficient for small-time local controllability. Before addressing the conditions for small-time local controllability, we establish the following fact.

PROPOSITION 3. Consider a set of velocity directions $\hat{\mathcal{V}}$ and its convex hull $CH_{S^2}(\hat{\mathcal{V}})$ on the velocity sphere. Any path from $\mathbf{q}_1$ to $\mathbf{q}_2$ using velocity directions in $CH_{S^2}(\hat{\mathcal{V}})$ can be followed arbitrarily closely by another path, also from $\mathbf{q}_1$ to $\mathbf{q}_2$, using only velocity directions in $\hat{\mathcal{V}}$.

Proof: Proposition 5 in Appendix B of Barraquand and Latombe (1993) proves the case when $\hat{\mathcal{V}}$ consists of two velocity directions. The result for any number of velocity directions follows by induction. □

On any open set of the configuration space $\mathcal{C}$, Proposition 3 says that we can consider the available velocity directions to be the convex hull of the velocity direction set $\hat{\mathcal{V}}$. Therefore, if the set of velocity directions $\hat{\mathcal{V}}$ contains four velocity directions that positively span the velocity sphere, the configuration of the slider S is small-time locally controllable. This also follows from the following theorem:

THEOREM 1. *(Sussmann 1978)* Let $\mathcal{X}$ be a finite set of vector fields on an open set of the state manifold containing $\mathbf{q}$. The set of nonzero tangent vectors at $\mathbf{q}$ is denoted $\mathcal{X}(\mathbf{q})$. Then:

1. If $\mathbf{0}$ is in the interior of the convex hull of $\mathcal{X}(\mathbf{q})$, the system is small-time locally controllable from $\mathbf{q}$.
2. If $\mathbf{0}$ does not belong to the convex hull of $\mathcal{X}(\mathbf{q})$, the system is not small-time locally controllable from $\mathbf{q}$.

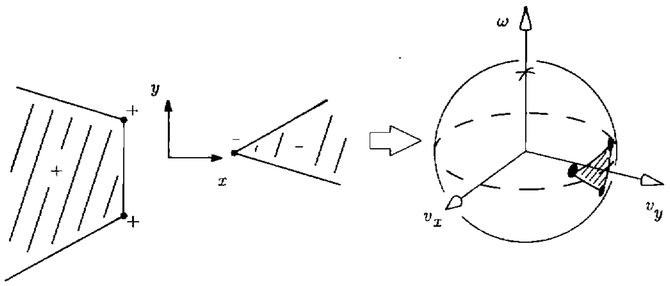

Fig. 6. The convex hull of three velocity directions, represented in the rotation center space and on the velocity sphere.

To apply this theorem, we define the convex hull in $\mathbf{R}^3$ of a set of unit velocities $\mathcal{X}(\mathbf{q})$ to be $CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$ with boundary $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$.

The first half of Theorem 1 indicates that if the velocity directions $\hat{\mathcal{V}}$ of the control system Σ positively span the velocity sphere, the control system Σ is small-time locally controllable. The second half of the theorem says that if the velocity directions $\hat{\mathcal{V}}$ are contained in any open hemisphere of the velocity sphere, then Σ is not small-time locally controllable. Theorem 1 does not completely answer the question of small-time local controllability at $\mathbf{q}$, however, as it does not address the case when $\mathbf{0}$ lies in $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$. To resolve this case, we must consider the derivatives of $\mathcal{X}(\mathbf{q})$ (Sussmann 1978).

If $\mathbf{0}$ lies in $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$, then there is a unique linear subspace Z of maximum dimension such that $\mathbf{0}$ lies in the interior of $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$ in this space Z. The set $\mathcal{X}_Z$ is the subset of vector fields $X \in \mathcal{X}$ such that the tangent vector $X(\mathbf{q})$ lies in the subspace Z. The dimension of the subspace Z is 1 if the portion of $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$ through $\mathbf{0}$ is a line segment, and $\mathcal{X}_Z$ consists of two opposite vector fields. The dimension of the subspace Z is 2 if the portion of $\partial CH_{\mathbf{R}^3}(\mathcal{X}(\mathbf{q}))$ through $\mathbf{0}$ is a planar region, and $\mathcal{X}_Z$ consists of at least three vector fields such that the associated velocity directions span a great circle of the velocity sphere.

The set of all vector fields $[X, Y]$ such that $X, Y \in \mathcal{X}_Z$ is denoted $\mathcal{X}_Z^1$. Applying Sussmann's (1978) sufficient condition for small-time local controllability, the control system Σ is small-time locally controllable if the convex hull of $\mathcal{X}(\mathbf{q})$ and $\mathcal{X}_Z^1(\mathbf{q})$ contains $\mathbf{0}$ in its interior. For the control system Σ, Sussmann's sufficient condition is also necessary.

When the dimension of Z is 1, the set of vector fields $\mathcal{X}_Z$ consists of two opposite vector fields. The Lie bracket of opposite vector fields is zero, so $\mathcal{X}_Z^1$ consists of the zero vector field. The convex hull of $\mathcal{X}(\mathbf{q})$ and $\mathcal{X}_Z^1(\mathbf{q})$ does not contain $\mathbf{0}$ in its interior, so the control system Σ is not small-time locally controllable.

Now consider the case where the dimension of the subspace Z is 2. In this case, the velocity directions corresponding to $\mathcal{X}_Z$ positively span a great circle of the velocity sphere. Provided that this great circle does not lie in the $\omega = 0$ plane, the control system Σ is small-time locally controllable. To see this, recall that two nonopposite velocity directions that are not both translations are sufficient for small-time accessibility. If both velocity directions can be reversed (a total of four velocity directions), then the system is small-time locally controllable. These velocity directions positively span a great circle of the velocity sphere such that $\hat{\omega}$ is not identically zero. By Proposition 3, on any open set of the configuration space $\mathcal{C}$, any set of velocity directions that positively span the same great circle is equivalent.

PROPOSITION 4. The control system Σ is small-time locally controllable if and only if the set of velocity directions $\hat{\mathcal{V}}$ positively spans a great circle of the velocity sphere that does not lie in the $\omega = 0$ plane.

This result is stated in terms of velocity directions on the velocity sphere, but it can just as easily be stated in terms of rotation centers. The positive span of two rotation centers of the same sense is given by the line segment of rotation centers of the same sense between the points. The positive span of rotation centers of opposite senses is given by all points on the line through the two rotation centers but not between them. The sense of the rotation centers changes at infinity, which corresponds to a translation (Fig. 6). A great circle on the velocity sphere is equivalent to a line of rotation centers with both rotation senses. Figure 7 gives examples of rotation center sets that yield small-time local controllability.

COROLLARY 2. The number of distinct combinations n of pushing contact configurations and pushing directions must be at least three for the configuration of the slider $\mathcal{S}$ to be small-time locally controllable by pushing. From any set of controls yielding small-time local controllability, no more than four are needed.

The second half of Corollary 2 follows from Proposition 4 by Carathéodory's Theorem (Grünbaum 1967).

3. Mechanics and Controllability of Pushed Objects

In this section we study the mechanics problem of determining the motion of a pushed object. Using these results and those of the previous section, we elucidate the controllability properties of objects pushed with either point contact or stable line contact.

3.1. Mechanics of Pushing

During quasistatic pushing, the force **f** applied by the pusher is equal to the frictional force that the slider applies to the support plane. The frictional force **f** that the slider applies to the support plane when moving with velocity **v** will be expressed in terms of the following:

x = point of contact $(x, y, 0)^T$ between the slider $\mathcal{S}$ and the support surface in the slider reference frame $\mathcal{F}_S$

d**x** = differential element of support area

$p(\mathbf{x})$ = support pressure at **x**

$\mu_s(\mathbf{x})$ = support friction coefficient at **x**

$s(\mathbf{x})$ = the product of $p(\mathbf{x})$ and $\mu_s(\mathbf{x})$, referred to as the support friction distribution

$\mathbf{v}(\mathbf{x})$ = the linear velocity of **x**, given by $(v_x - \omega y, v_y - \omega x, 0)^T$, where the slider velocity **v** is $(v_x, v_y, \omega)^T$

$\mathbf{f}_{xy}$ = the linear components $(f_x, f_y, 0)^T$ of **f**

The origin O_S of the slider frame $\mathcal{F}_S$ is located at the center of friction of the slider, which is the unique point of the slider such that $\int_S \mathbf{x}s(\mathbf{x})d\mathbf{x} = \mathbf{0}$. When $\mu_s(\mathbf{x})$ is constant, O_S is located directly beneath the center of mass (MacMillan 1936). All forces and velocities are expressed with respect to the slider frame $\mathcal{F}_S$.

The force $\mathbf{f}_{xy}$ and moment m components of the force **f** applied to the support plane by a slider $\mathcal{S}$ are given by Mason (1986):

$$\mathbf{f}_{xy} = \int_S \frac{\mathbf{v}(\mathbf{x})}{|\mathbf{v}(\mathbf{x})|} s(\mathbf{x})d\mathbf{x}$$

$$m\hat{\mathbf{k}} = \int_S \mathbf{x} \times \frac{\mathbf{v}(\mathbf{x})}{|\mathbf{v}(\mathbf{x})|} s(\mathbf{x})d\mathbf{x},$$

where $\hat{\mathbf{k}}$ is the unit vector $(0, 0, 1)^T$. These expressions simply state that the differential frictional force applied by the slider at each support point **x** acts in the direction of the velocity of **x** with magnitude $s(\mathbf{x})$.

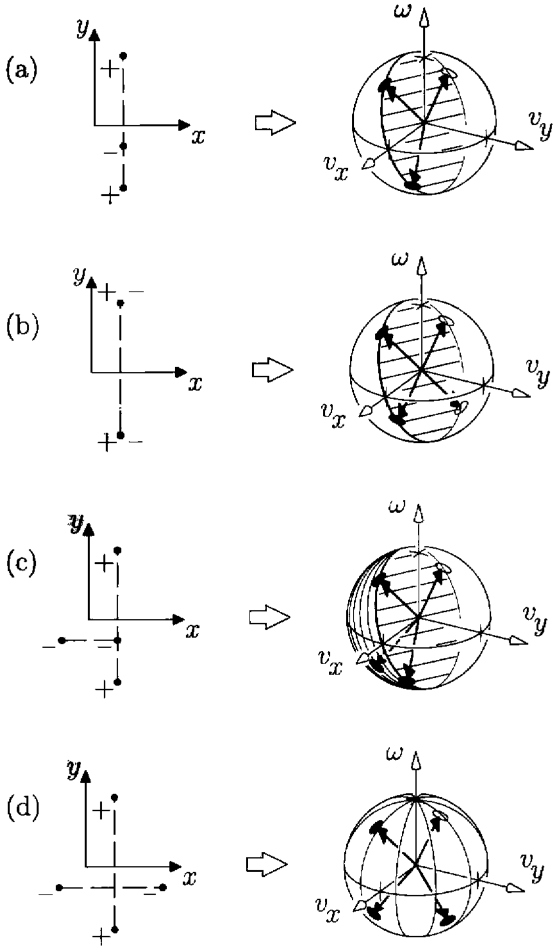

Fig. 7. Examples of rotation center sets that yield small-time local controllability. A, These three rotation centers positively span a great circle of the velocity sphere. B, These four rotation centers positively span a great circle of the velocity sphere. C, These four rotation centers positively span a closed hemisphere of the velocity sphere. D, These four rotation centers positively span the velocity sphere.

3.1.1. The Limit Surface

As the slider's velocity direction $\hat{\mathbf{v}}$ moves over the velocity sphere, the force **f** moves on a two-dimensional surface in the three-dimensional force space. This closed convex surface is called the *limit surface* by Goyal et al. (1991a). The limit surface encloses the set of all forces that can be statically applied to the slider, and during quasistatic motion the applied force lies on the limit surface. The slider's velocity direction vector $\hat{\mathbf{v}}$ is normal to the limit surface at the force **f**. If the force **f** lies on the limit surface with an associated velocity direction $\hat{\mathbf{v}}$, then the force −**f** also lies on the limit surface with an associated velocity direction $-\hat{\mathbf{v}}$ (Fig. 8).

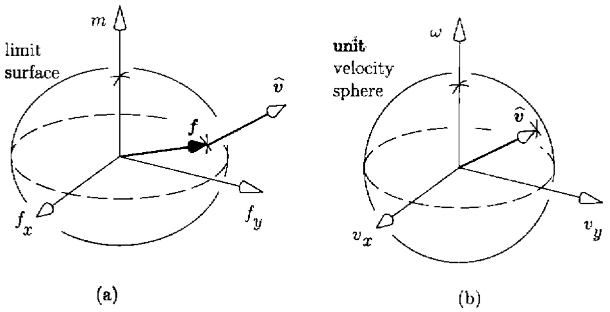

Fig. 8. Mapping a force **f** *through a limit surface to a velocity direction* $\hat{\mathbf{v}}$.

If the support friction distribution $s(\mathbf{x})$ is finite everywhere, the limit surface is smooth and strictly convex, defining a continuous one-to-one mapping from the set of force directions $\hat{\mathbf{f}} \in S^2$ to the set of velocity directions $\hat{\mathbf{v}} \in S^2$. If the applied force has zero moment about the center of friction, the resulting slider velocity is translational and parallel to the applied force.

If any single point $\mathbf{x}_0$ supports a finite force with a nonzero coefficient of friction $\mu_s(\mathbf{x}_0)$, however, the mapping is no longer one-to-one. The support pressure $p(\mathbf{x}_0)$ and support friction $s(\mathbf{x}_0)$ is infinite, and the integral of $s(\mathbf{x}_0)$ over the point $\mathbf{x}_0$ is nonzero. The limit surface therefore contains flat facets mapping sets of force directions to the same velocity direction: rotation about $\mathbf{x}_0$.

If the support friction distribution $s(\mathbf{x})$ is infinite only at points on a line, and $s(\mathbf{x})$ integrates to zero over the rest of the support surface, then the limit surface contains vertices. The normals to the limit surface at these vertices are not uniquely defined: the same force maps to a set of possible velocity directions.

Goyal et al. (1991a,b) provide more details on the properties of the limit surface.

3.1.2. Solving for the Motion of a Pushed Object

Each contact point between the pusher and the slider may be sticking, breaking free, or sliding to the left or right. The *contact mode* describes the qualitative behavior of each contact point between the pusher and the slider. For each possible contact mode i, there is a space of possible slider velocities $V_{k,i}$ that are kinematically consistent with that contact mode and the known pusher velocity (Lynch 1992). By Coulomb's law, each contact mode also specifies a polyhedral cone of possible pushing forces in the three-dimensional force space. This cone is the convex hull of the individual friction cones at the sticking contacts and the friction cone edges at the sliding contacts (Erdmann 1984, 1994). This composite friction cone is intersected with the limit surface to find a cone of possible velocities $V_{f,i}$ (Fig. 9). If $V_{k,i} \cap V_{f,i} = \emptyset$, contact mode i cannot occur; otherwise, contact mode i is feasible and any of the velocities in the intersection set is a possible solution to the motion of the slider. There may be more than one solution under Coulomb's law of friction.

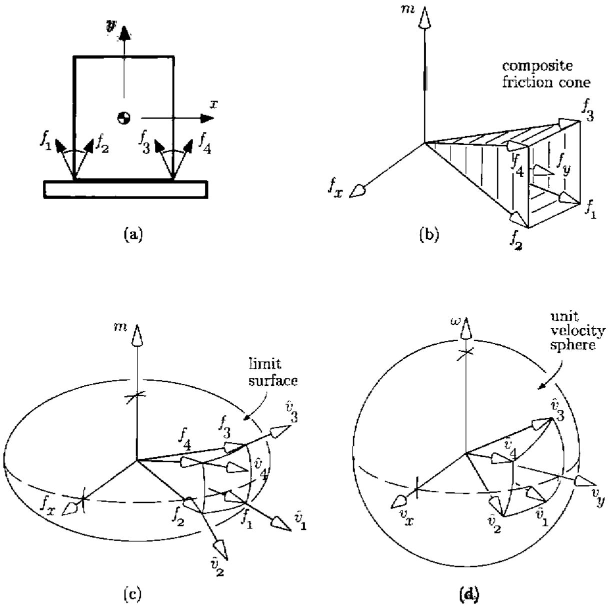

Fig. 9. A, The forces that the pusher can apply to the slider during sticking contact are represented by the two friction cones. B, The convex hull of these friction cones in the three-dimensional force space. The result is a composite friction cone of possible pushing forces. C, Mapping these forces through the limit surface for the slider. D, The slider velocity directions corresponding to forces inside the composite friction cone.

3.2. Pushing with Point Contact

3.2.1. Mechanics of Pushing with Point Contact

When the slider is pushed at a single point of contact, there are only three possible contact modes: sticking contact, left sliding, and right sliding. Mason (1986) developed search procedures to find the slider motion resulting in force/moment balance (for a known support friction distribution $s(\mathbf{x})$) for sticking and sliding contact. The actual motion is given by the contact mode that is consistent with all kinematic and force constraints. Peshkin and Sanderson (1988b) and Alexander and Maddocks (1993) derived similar search procedures that find the slider motion by minimizing the power loss due to frictional sliding. Peshkin and Sanderson (1989) proved that the power minimization approach is equivalent to the force/moment balance approach.

In general, the support friction distribution $s(\mathbf{x})$ is indeterminate. For this reason, several authors have investigated the use of weaker models of the support friction distribution (Mason 1986; Mason and Brost 1986; Peshkin and Sanderson 1988b; Alexander and Maddocks 1993). With these weaker models, it is usually impossible to find the exact motion of the slider. We do not need to be able to determine the motion of a pushed object to prove controllability results, however; we only need general properties of the limit surface and the available pushing contacts.

3.2.2. *Controllability by Pushing with Point Contact*

Open-loop pushing with point contact is inherently unstable, and many researchers have constructed point contact pushing control systems using visual (Inaba and Inoue 1989; Gandolfo et al. 1991; Salganicoff et al. 1993a; Narasimhan 1995), and tactile feedback (Okawa and Yokoyama 1992; Lynch et al. 1992). Here we examine the controllability of such a system.

PROPOSITION 5. The configuration of a slider $\mathcal{S}$ with a bounded support friction distribution $s(\mathbf{x})$ is controllable by pushing if and only if the pusher can apply two pushing force directions, $\hat{\mathbf{f}}_1$ and $\hat{\mathbf{f}}_2$, such that they do not both pass through the center of friction ($\hat{m}_1 \neq 0$ or $\hat{m}_2 \neq 0$) and $\hat{\mathbf{f}}_1 \neq -\hat{\mathbf{f}}_2$.

Proof: Because the support friction distribution $s(\mathbf{x})$ is bounded, the limit surface has no facets, and therefore the two force directions map through the limit surface to two distinct velocity directions. At least one of the force directions has nonzero moment, so at least one of the velocity directions has a nonzero angular component. Because the two force directions are not opposite, the two velocity directions are also not opposite. Therefore, by Proposition 2, the slider is controllable. □

A slider that is uncontrollable by pushing is a disk centered at its center of friction with a pushing friction coefficient of zero. All pushing forces pass through the center of friction, creating zero moment about the center of friction. The slider cannot be rotated (except nondeterministically if its limit surface contains vertices).

Theorem 2 is a direct application of Proposition 5 to polygonal sliders.

THEOREM 2. The configuration of a polygonal slider $\mathcal{S}$ with a bounded support friction distribution $s(\mathbf{x})$ is controllable by pushing with point contact on any edge. It is also controllable from any vertex that has nonzero friction and is not at the center of friction.

A slider that is controllable by pushing may have to be pushed a long distance to reach nearby configurations. If the object is small-time locally controllable, however, it can follow any path arbitrarily closely. To find conditions for small-time local controllability of a slider, first recall that the set of available pushing contacts is given by Γ, a closed, piecewise smooth curve. At each point of Γ that is not a vertex, the curve Γ has a unique inwardly pointing contact normal. At a vertex, we assume that the contact normal can take any direction in the range specified by the contact normals adjacent to the vertex. Each contact point and contact normal specifies a pushing force direction that can be applied to the slider $\mathcal{S}$, and the curve of all such force directions is denoted $\hat{\mathbf{f}}(\Gamma)$. Because Γ is a closed curve, $\hat{\mathbf{f}}(\Gamma)$ is a (possibly self-intersecting) closed curve of force directions on the force sphere.

THEOREM 3. The configuration of any slider $\mathcal{S}$ with a closed, piecewise smooth curve Γ of available pushing contact points is small-time locally controllable by pushing with point contact, unless the pushing contact is frictionless and Γ is a circle centered at the slider's center of friction (a frictionless disk).

Proof: *Case 1: Γ not a circle.* Following the argument of Hong et al. (1990), $\hat{\mathbf{f}}(\Gamma)$ must contain at least two pairs of opposite force directions. By the limit surface mapping, these forces yield two pairs of opposite velocity directions that span a great circle of the velocity sphere. By Proposition 4, the slider is small-time locally controllable, unless this great circle lies in the $\omega = 0$ plane. In this case we use the result of Mishra et al. (1987), which states that $\hat{\mathbf{f}}(\Gamma)$ positively spans the force sphere. Therefore, $\hat{\mathbf{f}}(\Gamma)$ contains forces with positive and negative moment, and the slider can be rotated clockwise or counterclockwise. The $\omega = 0$ great circle and any clockwise and counterclockwise directions positively span the velocity sphere, and the slider is small-time locally controllable.

Case 2: Γ a circle. Every pair of diametrically opposed points on Γ gives rise to a pair of opposite velocity directions. If the center of friction is offset from the center of the circle, then only one pair lies in the $\omega = 0$ plane, and therefore any two pairs of opposite velocity directions yield small-time local controllability. If the center of friction is at the center of the circle and there is nonzero friction at the pushing contact, the slider can be translated in any direction and rotated using frictional forces to create moment about the center of friction. The slider is small-time locally controllable. If the contact is frictionless, however, the object cannot be rotated. A frictionless disk centered at its center of friction is the only type of slider that is not small-time locally controllable by point contact pushing. (If the slider's limit surface contains vertices, it may rotate nondeterministically.) □

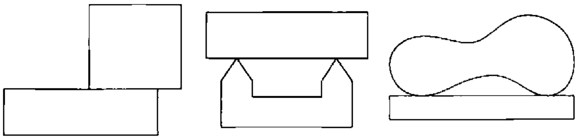

Fig. 10. Examples of line contact.

Theorem 3 implies that a two-degree-of-freedom robot (a point translating in the plane) can push any object (other than a frictionless disk) to follow any planar path arbitrarily closely.

3.3. Stable Pushing with Line Contact

3.3.1. Mechanics of Stable Pushing with Line Contact

In the previous section we examined the controllability of sliders pushed with point contact, but we would also like to synthesize pushing controllers. Unfortunately, the motion of a slider pushed with a single point of contact is usually unpredictable, because it depends on the unknown and generally indeterminate support friction distribution $s(\mathbf{x})$. If there are two or more simultaneous pushing contacts, however, there may exist a space of pushing directions that, despite some uncertainty in the support friction distribution $s(\mathbf{x})$, result in a predictable motion of the slider: sticking at all contact points. We call such a push a *stable push*, and we will use these stable pushes to execute open-loop pushing plans.

In this section we focus on stable pushing with line contact: all contacts between the pusher and the slider are collinear with contact normals perpendicular to the line (Fig. 10). A line contact is equivalent to two contact points at the ends of the line segment. We also assume that the friction coefficient at all pushing contacts is the same.

For a given line contact, we use the following definitions:

- $\hat{\mathcal{V}}_{stable}$: The set of pushing directions such that the slider remains fixed to the pusher during the motion.
- $\hat{\mathcal{V}}_{\mathcal{F}}$: The set of pushing directions such that one solution to the motion of the slider is to remain fixed to the pusher. This set of velocity directions is found by intersecting the composite friction cone $\mathcal{F}$ from the line contact with the limit surface (see Fig. 9).

If a velocity direction $\hat{\mathbf{v}}$ is in $\hat{\mathcal{V}}_{stable}$, then it is also in $\hat{\mathcal{V}}_{\mathcal{F}}$, but the converse is not necessarily true. Although stable contact is always a possible solution if the pushing direction $\hat{\mathbf{v}}$ is in $\hat{\mathcal{V}}_{\mathcal{F}}$, there may be other solutions. It is necessary to prove all other contact modes inconsistent. In this section we describe the procedure STABLE for finding a subset of $\hat{\mathcal{V}}_{\mathcal{F}}$ and provide a theorem stating that this subset also belongs to $\hat{\mathcal{V}}_{stable}$.

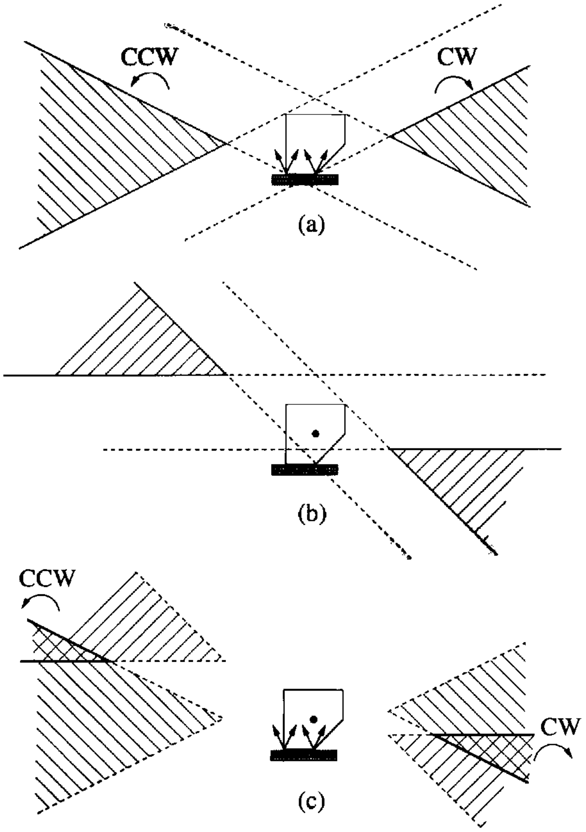

Fig. 11. Illustration of the procedure STABLE. *A, Rotation centers that can be achieved by forces inside the friction cone angular limits. The friction coefficient is 0.5. B, Rotation centers that can be achieved by forces passing between the line contact end points. C, The intersection of the closed regions found in A and B correspond to rotation centers that can be achieved by forces inside the composite friction cone $\mathcal{F}$ defined by the line contact.*

Usually the support friction distribution $s(\mathbf{x})$ is unknown, and therefore we cannot determine $\hat{\mathcal{V}}_{\mathcal{F}}$ exactly. Instead, we assume that the center of friction and the shape of the slider are known. Lynch (1992) presented an algorithm for finding an approximation to $\hat{\mathcal{V}}_{\mathcal{F}}$ for a slider with a known center of friction. This algorithm utilizes results due to Peshkin and Sanderson (1988b) on the possible support friction distributions of a disk slider.

Here we describe the simpler procedure STABLE for finding a conservative approximation to $\hat{\mathcal{V}}_{\mathcal{F}}$. We will illustrate the procedure in the rotation center space. Without loss of generality, assume that the line contact is horizontal on the page with an upward pointing contact normal, as in Figure 11.

Procedure Stable

1. The coefficient of friction defines two friction cone edges at angles $\tan^{-1}\mu$ to the contact normal, where μ is the coefficient of friction. For each edge of the friction cone, draw two lines perpendicular to the friction cone edge such that the entire slider is contained between the two lines. For an applied force at an edge of the friction cone, the resulting rotation center must lie in its respective band (Mason and Brost 1986; Alexander and Maddocks 1993). Counterclockwise (respectively clockwise) rotation centers between the two bands and to the left (resp. right) of the slider correspond to force angles inside the angular limits of the friction cone (see Fig. 11*A*).
2. For each end point of the line contact, draw two lines perpendicular to the line through the center of friction and the end point. One of these lines is the perpendicular bisector between the contact point and the center of friction (Mason and Brost 1986). The other is a distance r^2/p from the center of friction and on the opposite side from the end point, where p is the distance from the end point to the center of friction and r is the distance from the center of friction to the most distant support point of the slider. (This "tip line" should actually be slightly more distant from the center of friction [Peshkin and Sanderson 1988b].) The rotation center from a force through this end point must lie in the band between these two lines (Mason and Brost 1986). All rotation centers between the two bands correspond to forces that pass between the two end points (see Fig. 11*B*).
3. The intersection of the closed regions found in (1) and (2) yield a set of rotation centers corresponding to forces that are guaranteed to lie on or inside the composite friction cone $\mathcal{F}$ from the line pushing contact. This is a conservative approximation to $\hat{\mathcal{V}}_{\mathcal{F}}$ (see Fig. 11*C*).

Stable misses some tight turning rotation centers but finds all translations belonging to $\hat{\mathcal{V}}_{\mathcal{F}}$. If the composite friction cone $\mathcal{F}$ contains any pure force (zero moment about the center of friction) in its interior, Stable will find a closed, convex polygon of velocity directions with nonempty interior and a range of translation directions. If the composite friction cone $\mathcal{F}$ contains only a single pure force direction, necessarily on the boundary of $\mathcal{F}$, Stable finds only a single translation in the direction of this pure force. If the composite friction cone $\mathcal{F}$ does not contain a pure force, Stable finds no velocity directions belonging to $\hat{\mathcal{V}}_{\mathcal{F}}$. In fact, with no information about the support friction distribution $s(\mathbf{x})$ other than the center of friction, no velocity direction is guaranteed to be in $\hat{\mathcal{V}}_{\mathcal{F}}$ if the line contact cannot apply a force through the center of friction.

Proposition 6. If the pusher $\mathcal{P}$ makes line contact with the slider $\mathcal{S}$ with a composite friction cone $\mathcal{F}$ that does not include a force through the center of friction, then, in the absence of any other information about the support friction distribution $s(\mathbf{x})$, no single velocity direction is guaranteed to be achievable by a force in $\mathcal{F}$.

Proof: With no information about the support friction distribution $s(\mathbf{x})$, we can always choose the support friction distribution to be concentrated arbitrarily close to the center of friction. In this case, all rotation centers not located at the center of friction correspond to pushing forces passing through or arbitrarily close to the center of friction, which are not included in the composite friction cone $\mathcal{F}$. If the rotation center is located at the center of friction, any support friction distribution $s(\mathbf{x})$ that is symmetric about the center of friction corresponds to a pushing force that is a pure moment. A pure moment cannot be applied by line contact with a finite object. □

Proving that a pushing direction in the set found by Stable belongs to $\mathcal{V}_{stable}$ requires proving all other contact modes inconsistent. Here we state the relevant result, omitting the proof by case analysis.

Theorem 4. Given a pusher in line contact with a slider $\mathcal{S}$ with a bounded support friction distribution $s(\mathbf{x})$, let $\hat{\mathcal{V}}_{\mathcal{F}}$ be the set of rotation centers resulting from forces in the composite friction cone $\mathcal{F}$. Draw two lines perpendicular to the line contact such that the entire slider $\mathcal{S}$ is contained between the two lines. All rotation centers in $\hat{\mathcal{V}}_{\mathcal{F}}$ and outside the two lines are guaranteed to belong to the set of stable pushing directions $\hat{\mathcal{V}}_{stable}$. Therefore, all velocity directions found by Stable belong to $\hat{\mathcal{V}}_{stable}$.

3.3.2. *Controllability by Stable Pushing with Line Contact*

If the composite friction cone $\mathcal{F}$ from the line pushing contact contains a pure force in its interior, then Stable finds a convex set of velocity directions $\hat{\mathcal{V}}_{\mathcal{F}}$ with nonempty interior (including a range of translations). By Proposition 2, we get the following.

Theorem 5. If a slider $\mathcal{S}$ is pushed with line contact with a composite friction cone $\mathcal{F}$ such that $\mathcal{F}$ contains a pure force (zero moment about the center of friction of the slider) in its interior, then the configuration of the slider is controllable by the stable pushes found by Stable.

If Theorem 5 is satisfied, the slider can be pushed to any configuration in the obstacle-free plane using the stable pushes found by STABLE. We can apply Theorem 5 to prove the following result regarding polygonal sliders.

THEOREM 6. For a polygonal slider $\mathcal{S}$ and a sufficiently long straight-edge pusher $\mathcal{P}$, any edge of the convex hull of $\mathcal{S}$ can be used as the line pushing contact. If the center of friction of $\mathcal{S}$ does not lie on a vertex of the convex hull, and there is nonzero friction at the line contacts, then there is at least one line contact from which the slider is controllable by the stable pushes found by STABLE.

Proof: The center of friction only lies on a vertex of the convex hull if the support friction distribution $s(\mathbf{x})$ integrates to zero everywhere else. Otherwise, there is at least one normal to the interior of an edge of the convex hull that passes through the center of friction. A force $\mathbf{f}$ along that normal is in the interior of the composite friction cone for that edge. By Theorem 5, the slider is controllable from that edge by the stable pushes of STABLE. □

It is intuitively clear that a slider is not small-time locally controllable by pushing a single edge. It is impossible to "pull" the slider backward, and therefore it cannot follow all paths arbitrarily closely. If we allow the pusher to change contact configurations, however, in some cases the configuration of the slider may be made small-time locally controllable by stable pushes with line contact. The following theorem gives a sufficient condition for a slider to be small-time locally controllable by line contact pushing.

THEOREM 7. Given a set of line pushing contacts with composite friction cones $\mathcal{F}_i$, find the set of all pure forces (zero moment about the center of friction of the slider) interior to at least one of the composite friction cones $\mathcal{F}_i$. If these pure forces positively span the plane of pure forces, then the configuration of the slider $\mathcal{S}$ is small-time locally controllable by the stable pushes found by STABLE.

Proof: If the conditions of Theorem 7 are satisfied, then STABLE will find a set of translation directions that positively spans the $\omega = 0$ great circle. Because each of these translation directions has a neighborhood of velocity directions also in the set found by STABLE, the velocity directions found by STABLE positively span the velocity sphere. □

The slider of Figure 11 is small-time locally controllable by stable pushing with the two line contacts shown in Figure 12.

Applying Theorem 7, it is easy to show that any rectangular or regular $2k$-gon slider is small-time locally controllable by the stable pushes found by STABLE, provided the center of friction is in the interior of the slider and there is nonzero friction at the edge contacts. It suffices to show there are two opposing edges with opposing pure forces in the interior of their respective friction cones.

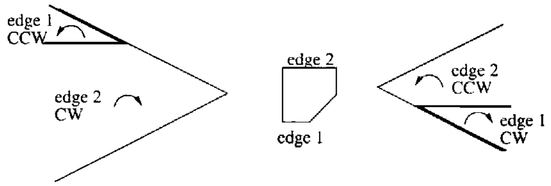

Fig. 12. The slider of Figure 11 is small-time locally controllable by stable pushing at two edges using pushes found by STABLE. *The friction coefficient is 0.5.*

4. Planning Stable Pushing Paths among Obstacles

The stable pushing directions impose nonholonomic constraints on the motion of the pusher, and the problem is to plan free pushing paths among obstacles subject to these nonholonomic constraints. This problem is similar to the problem of planning paths for car-like mobile robots, the subject of much recent robotics research. The configuration space of a car-like mobile robot is also $\mathbf{R}^2 \times S^1$, but its feasible velocity direction set is only one-dimensional, corresponding to the angle of the steering wheel. Despite this, Laumond (1986) showed that a car-like robot that can reverse can reach any configuration in any open connected subset of its free configuration space. Barraquand and Latombe (1993) showed that only two steering directions are required.

Many path planners for car-like mobile robots have been proposed; see Chapter 9 of Latombe (1991) for a survey. In our work on planning pushing paths, we chose to adapt an algorithm by Barraquand and Latombe (1993) due to its simplicity and its adaptability to two-dimensional velocity direction sets. The resulting algorithm closely parallels that described by Barraquand and Latombe. We provide only a brief description below. We refer the interested reader to Barraquand and Latombe (1993) for details.

4.1. Algorithm

To plan stable pushing paths, we first choose a discrete set of m line pushing contacts. For each pushing contact $i = 1, \ldots, m$, we calculate the set of stable pushing directions using the procedure STABLE. In light of Proposition 3, the planner uses discrete sets of velocity directions $\hat{\mathcal{V}}^i_{extremal}$, where $\hat{\mathcal{V}}^i_{extremal}$ is the set of vertices of the convex polygon of velocity directions found by STABLE for pushing contact i. We denote the union of $\hat{\mathcal{V}}^i_{extremal}$ for all i to be $\hat{\mathcal{V}}_{extremal}$.

The planner, shown below in pseudo-code, is a simple best-first search using a variation of Dijkstra's algorithm (Aho et al. 1974). The planner constructs a tree T of configurations reached in the search and a list $OPEN$ of configurations in T whose successors have not yet been generated. Configurations in $OPEN$ are sorted by the costs of their paths. The planner either returns failure or a path to a user-specified goal neighborhood. Note that the planner is not exact, as it only finds a path to a goal neighborhood.

```
program push_planner
  initialize $T$, $OPEN$ with start config $q_{init}$
  while $OPEN$ not empty
    $q \leftarrow$ first in $OPEN$, remove from $OPEN$
    if $q$ is in the goal region
      report success
    if $q$ is not near previously occupied config
      mark $q$ occupied
      for each pushing contact $i = 1, \ldots, m$
        for each pushing direction in $\hat{\mathcal{V}}^i_{extremal}$
          integrate forward from $q$ a small distance,
          compute cost of path to the new config $q_{new}$
          if (cost <= MAXCOST and
            path is collision-free)
            make $q_{new}$ a successor to $q$ in $T$
            and place $q_{new}$ in $OPEN$, sorted by cost
  report failure
end
```

4.2. Details of the Implementation

4.2.1. Collision Detection

The workspace is an open rectangular subset of $\mathbf{R}^2$, and obstacles are represented as closed polygons. For a particular pushing contact configuration, $\mathcal{PS}$ is the closed region of the workspace occupied by the pusher $\mathcal{P}$ and the slider $\mathcal{S}$. $\mathcal{PS}(\mathbf{q})$ is the closed region of the workspace occupied by the pusher and the slider at the configuration $\mathbf{q}$. The obstacle space $\mathcal{C}_{obs}$ is the closed set of configurations $\mathbf{q}$ such that $\mathcal{PS}(\mathbf{q})$ intersects an obstacle or the walls bounding the workspace. The free space $\mathcal{C}_{free}$ is the open set of the configuration space $\mathcal{C}$ complementary to $\mathcal{C}_{obs}$.

The free space $\mathcal{C}_{free}$ changes with the pushing contact, but for simplicity the planner uses a single representation of free space for all pushing contacts. This simplification is appropriate when the pusher $\mathcal{P}$ is much smaller or much larger than the slider $\mathcal{S}$. The region occupied by the pusher and the slider is treated as the smallest disk that encloses them for all pushing contact configurations. A collision occurs when this disk intersects the polygonal obstacles. This representation of the free space $\mathcal{C}_{free}$ is not exact, but the resulting code is simple and the collision detection routine is fast.

The planner checks for collisions at each new configuration in the search, not along the path. For this reason, the disk approximation to $\mathcal{PS}$ is grown by the maximum distance any point in $\mathcal{PS}$ can move in a single step. This ensures that if a new configuration is free, then the path that transferred it there is also free. This approach is simple and fast but somewhat conservative: the planner may not find paths through tight spots where paths exist.

4.2.2. Cost Function

In the current implementation, the cost of a path is the sum of an integer a times the number of pushing steps, an integer b times the number of control changes, and an integer c times the number of changes of the pushing contact. These values must all be nonnegative. The user can control the maximum permissible cost $MAXCOST$ for a path.

Because path costs are always integral, the sorted list $OPEN$ is represented by a 1-D array of linked lists, where each array index represents the cost of the paths to the configurations in its linked list. Inserting a new configuration into $OPEN$ consists of simply appending it to the end of the appropriate linked list and therefore takes constant time.

4.2.3. Pruning

The planner prunes configurations that are sufficiently close to configurations that have been reached with the same or lower cost and the same pushing contact. Two configurations are considered sufficiently close if they occupy the same cell of a predefined grid on the configuration space.

4.2.4. Parameters

The user must specify the parameters defining the size of the goal neighborhood $\mathcal{G}(\mathbf{q}_{goal})$, the length of the integration step, and the resolution of the configuration space grid used to check for prior occupancy. These parameters are interdependent. The resolution of the grid should be sufficiently fine that the application of any control moves the configuration to a new grid cell, and the goal neighborhood should be large enough that $\mathcal{PS}$ does not jump over it. The user must also specify the maximum number of configurations that the planner will explore before returning failure.

4.2.5. Changing the Pushing Contact

We assume that the pusher $\mathcal{P}$ can change pushing contacts at any time. If the slider is pushed by a manipulator capable of moving above the obstacles, such as a robot arm pushing objects on a table, this is a reasonable assumption. If the pusher is constrained to move in the plane, such as a mobile robot, however, this planner does not address the problem of finding paths between pushing contacts.

4.3. Properties of the Planner

This planner inherits properties of the planner described by Barraquand and Latombe (1993). Changing contact configurations in the pushing case is essentially equivalent to shifting between forward and reverse in the car-like robot case. (Note that we assume there is always a path for the pusher from one feasible pushing contact to another.) With an exact collision detection routine, it is possible to choose search parameters such that the following properties hold:

- Completeness. *If there is a feasible pushing path from the initial configuration* $\mathbf{q}_{init}$ *to the goal neighborhood* $\mathcal{G}(\mathbf{q}_{goal})$ *using the stable pushing directions found by* STABLE, *then the planner will find a pushing path.*
- Optimality. *If the cost of the pushing path is given by the number of changes of the pushing contact, then the planner will find a pushing path with* λ *changes of the pushing contact, where* λ *is the minimum number of contact changes for any feasible pushing path connecting the initial configuration* $\mathbf{q}_{init}$ *to the goal neighborhood* $\mathcal{G}(\mathbf{q}_{goal})$ *using velocity directions found by* STABLE.

Now assume that the extent of the pusher $\mathcal{P}$ is negligible so that $\mathcal{PS}$ is equivalent to the slider $\mathcal{S}$. If $\hat{\mathcal{V}}_{extremal}$ satisfies the conditions for small-time local controllability, then, with an exact collision detection routine, it is possible to choose search parameters such that the following property holds:

- Existence of a solution. *If the initial configuration* $\mathbf{q}_{init}$ *and the goal configuration* $\mathbf{q}_{goal}$ *lie in the same connected component of the slider's free configuration space* $\mathcal{C}_{free}$, *then the planner will find a pushing path connecting* $\mathbf{q}_{init}$ *to the goal neighborhood* $\mathcal{G}(\mathbf{q}_{goal})$.

The proofs of these properties do not suggest how small to choose the integration step or how to choose the other search parameters. See Barraquand and Latombe (1993) for a more detailed discussion of the properties of the planner.

4.4. Experimental Results

The planner is implemented in C on a Sparc 20. This section presents some pushing paths generated for the slider of Figure 12.

In the first two examples, a mobile robot pushes the slider from edge 1 in Figure 12. Because the mobile robot pushes from only a single edge, the slider is not small-time locally controllable. The extremal velocity directions $\hat{\mathcal{V}}_{extremal}$ found by STABLE consist of two rotations and two translations. The goal region is about ±25% the length of the slider and ±2 degrees.

In Figure 13, the slider is pushed to the goal by an omnidirectional mobile robot. This path took about 7 seconds to generate. The same problem is presented to a car-like mobile robot, which imposes additional constraints on the possible pushing directions. Figure 14 shows the intersection of the stable pushing directions with the set of possible robot velocity directions, limited by the rolling constraint of parallel wheels and a minimum turning radius. Combining the constraints, we see that the robot cannot execute a left-turn stable push: the extremal velocity directions consist of a straight-ahead motion and a right turn. (Even if the straight-ahead motion is not extremal, it should be included in the planner's control set to smooth paths.) Figure 15 shows the path generated for the car-like mobile robot. The robot is forced to take a longer path due to its kinematic constraints. Planning time was 1 second. In both examples, the cost function is $a = 1$, $b = 5$, and $c = \infty$ (edge changes are disallowed). A nonzero value of b tends to smooth the paths by eliminating excessive control changes.

In the next two examples, a robot arm pushes the slider on a table, and the arm is capable of lifting up and changing the pushing contact. The pusher can contact the slider at the two opposite edges shown in Figure 12. The resulting sets of stable pushing directions yield small-time local controllability. For each edge, the planner uses the four extremal velocity directions found by STABLE. The area of the pusher $\mathcal{P}$ in the plane is assumed to be negligible.

Figures 16 and 17 show pushing paths that solve the same problem using different cost functions. In these examples the goal region is about ±10% the length of the slider and ±2 degrees. In Figure 16, the cost function is $a = 1$, $b = 5$, and $c = 0$: short paths with few control changes are preferred. The resulting path took 35 seconds to find. In Figure 17, the cost function is $a = 0$, $b = 1$, and $c = 10$: paths that minimize contact changes are preferred. The planner found this path in 2 seconds.

Figure 18 presents another example of maneuvering the slider in a cluttered workspace using two pushing edges. The size of the goal region is the same as the previous

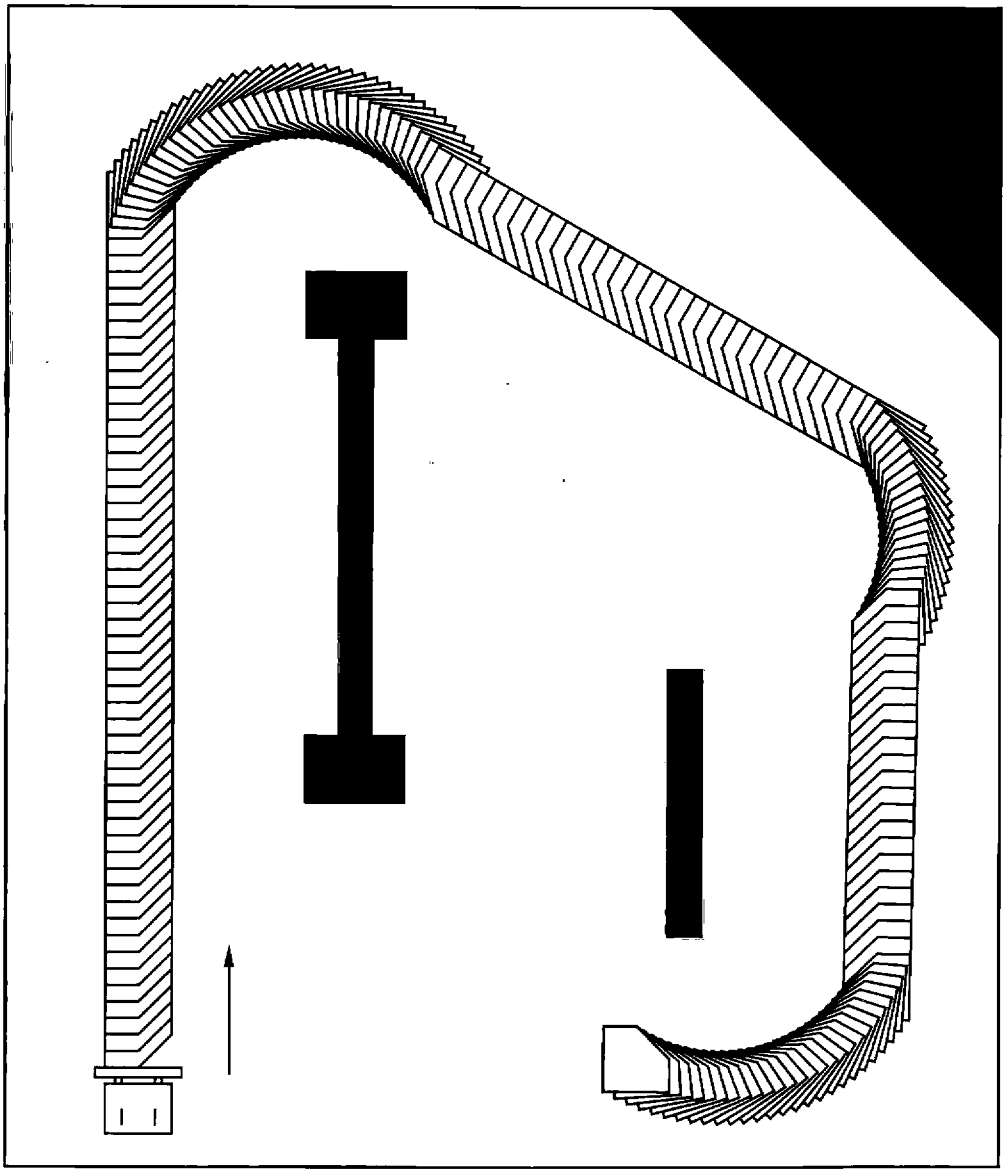

Fig. 13. The slider being pushed to the goal by an omnidirectional mobile robot. For clarity, the mobile robot is drawn only at the beginning of the path.

two examples, and the cost function is $a = 0$, $b = 1$, and $c = 10$. Planning time was 43 seconds.

Our experience shows that this simple push planner can quickly and reliably solve complex manipulation problems. The user must choose the integration step size with care, however. If the step is too small, the search will reach its memory capacity before it reaches the goal region; if the step is too large, tight paths will be missed, and the goal size may have to be increased.

We are using an Adept 550 tabletop robot to execute pushing plans similar to those above, with object dimensions of about 30 mm. The user enters the pushing problem via a graphical user interface, and the solution is automatically downloaded and executed by the robot. The plans are robust. More information about the implementation, the planner code, and the graphical user interface can be found on the World Wide Web at `http://www.cs.cmu.edu/~mlab`.

4.5. Variations on the Planner

To shrink the size of the goal neighborhood without sacrificing much speed in planning, the integration step and grid cell sizes could be decreased in the neighborhood of the goal. This would allow fine positioning near the goal.

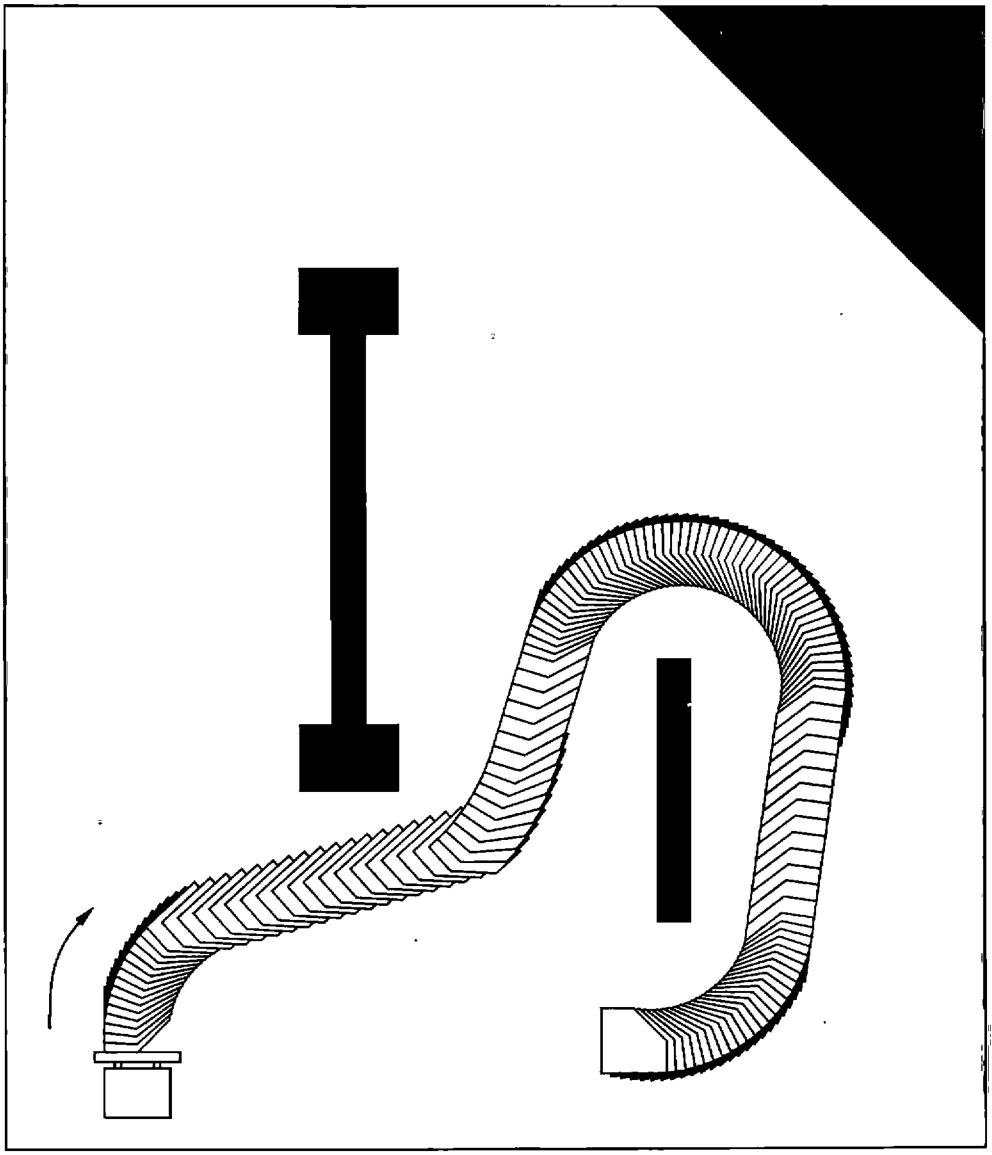

Fig. 15. The slider being pushed to the goal by a car-like mobile robot.

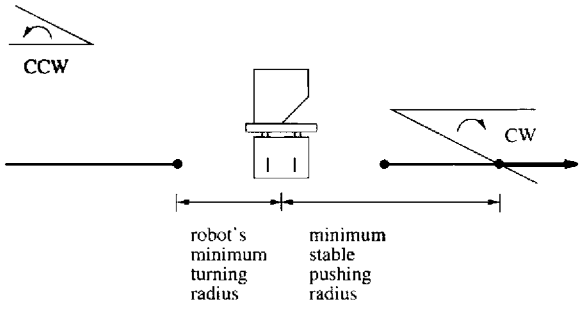

Fig. 14. Intersecting the feasible velocity directions of the car-like mobile robot pusher (represented as rotation centers) with the stable pushing directions for the slider. The intersection prohibits left turns.

The planner could be modified to find a path to an exact goal configuration instead of a goal neighborhood. A very simple way to implement this is to plan a path to a goal neighborhood as before, then back up along the path until a configuration is found from which $\mathcal{PS}$ can move exactly to the goal configuration using a single pushing direction in $\hat{\mathcal{V}}_{stable}$. This is possible due to the fact that the space of stable pushing directions has nonempty interior on the velocity sphere. (This is not true for the case of a car-like mobile robot pusher.) The preimage of the goal configuration under all constant stable pushing directions therefore encloses a volume of the configuration space $\mathcal{C}$ with nonempty interior. An even better solution is to define

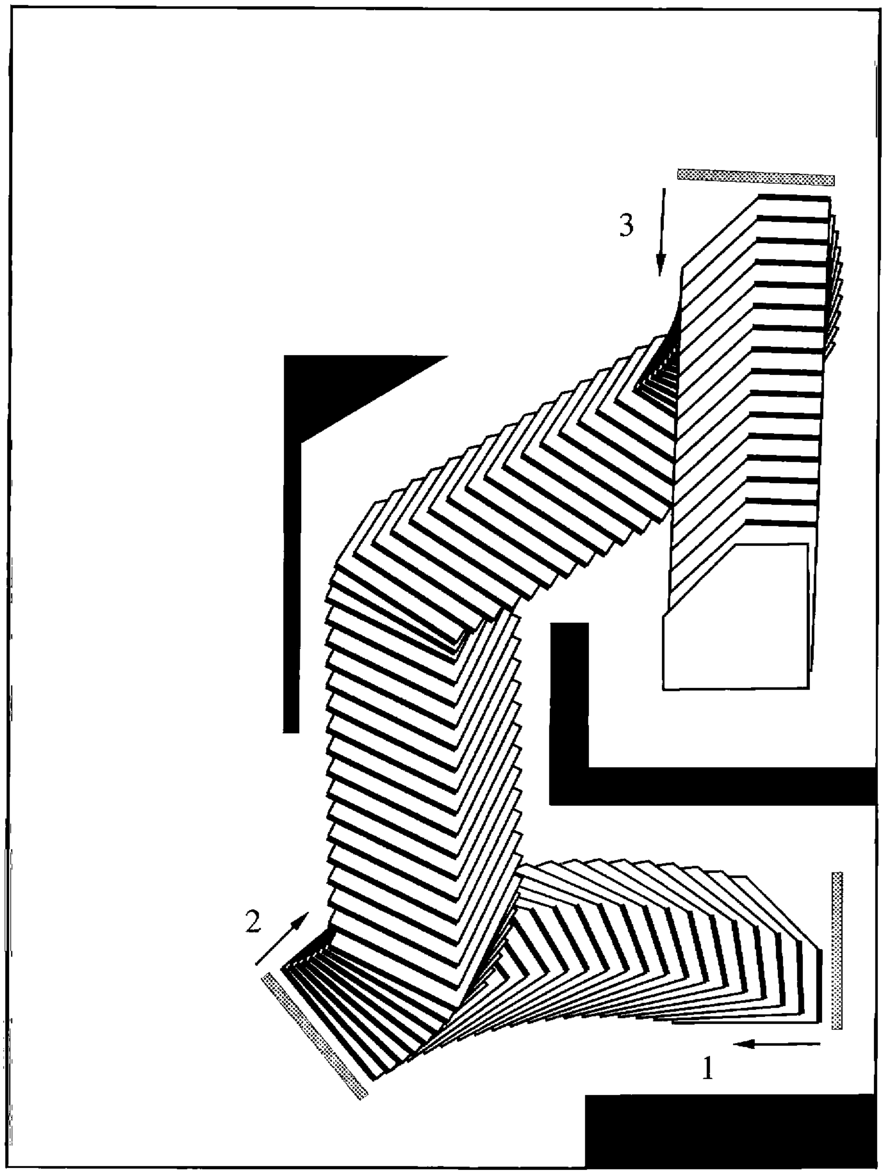

Fig. 16. A pushing plan found with cost function $a = 1$, $b = 5$, and $c = 0$.

this preimage as the goal region. Another approach is to find an exact holonomic path and transform it into one that obeys the nonholonomic constraints (Laumond et al. 1994).

The examples presented in Section 4.4 use stable translational velocity directions that border on being unstable. These pushing paths may be made robust to bounded uncertainty in the pushing friction coefficient and the location of the center of friction by propagating this uncertainty through the procedure STABLE. The result is to shrink the set returned by STABLE. Brost (1988; 1992) has applied this idea to several manipulation tasks.

4.6. Open Problems

This planner ignores the issue of finding the path for the pusher from one feasible pushing contact to another but assumes that a free path always exists. If the pusher is constrained to move in the plane of the obstacles, this problem must be addressed. Wilfong (1988) describes an algorithm for a convex polygon translating among obstacles and pushing another convex polygon at edge contacts, without considering pushing direction constraints. In the terminology of Alami et al. (1989), a path between pushing contacts is a *transit path*, and a pushing path is a *transfer path*. They describe an approach to planning both the transit and transfer paths using

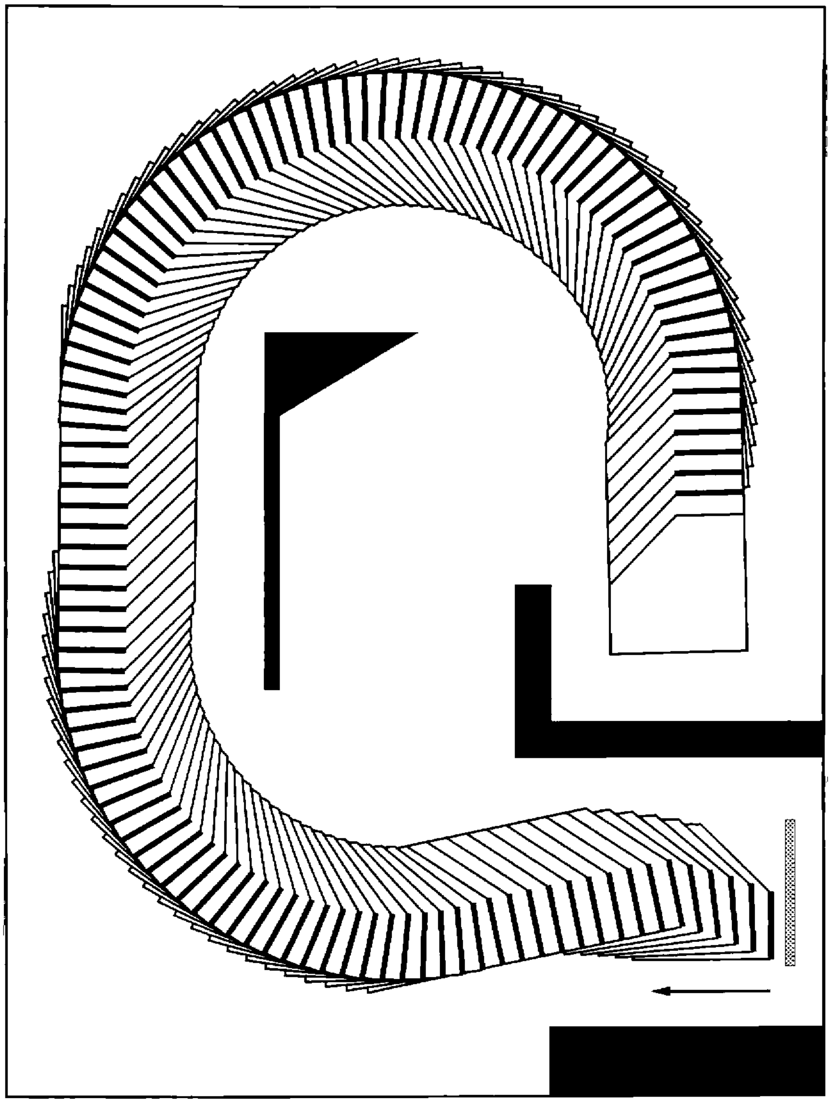

Fig. 17. A pushing plan found with cost function $a = 0$, $b = 1$, and $c = 10$.

manipulation *graphs*. In their problem, the robot and the manipulated object translate in a planar workspace with obstacles, and the robot can grasp the object at a discrete set of locations and place the object at a discrete set of configurations. The planner of Dacre-Wright et al. (1992) addresses the case of a convex planar robot and object where the robot can grasp the object at any contact. Koga and Latombe (1994) use manipulation graphs to find plans for a multiple arm robot manipulating a single object among obstacles.

The push planner is asymptotically optimal in the sense that if a feasible pushing path exists, then with an appropriate choice of search parameters, it will find a path that minimizes the number of changes in the pushing contact. We can make no claim, however, that the resulting path will be "short." An open problem is how to find shortest paths for a polygonal set of feasible velocity directions. Dubins (1957) found a set of canonical paths in the obstacle-free plane that is guaranteed to include the minimum arclength path for a car-like mobile robot that can only go forward. This result has been utilized by Jacobs and Canny (1989) in planning paths among obstacles. Reeds and Shepp (1990) enumerated a set of canonical paths, guaranteed to include the shortest path in the obstacle-free plane, for a car-like mobile robot that can reverse. This result has also been used in planning paths among obstacles (Laumond et al. 1994). As far as we are aware, there have been no similar results for more general sets of velocity directions.

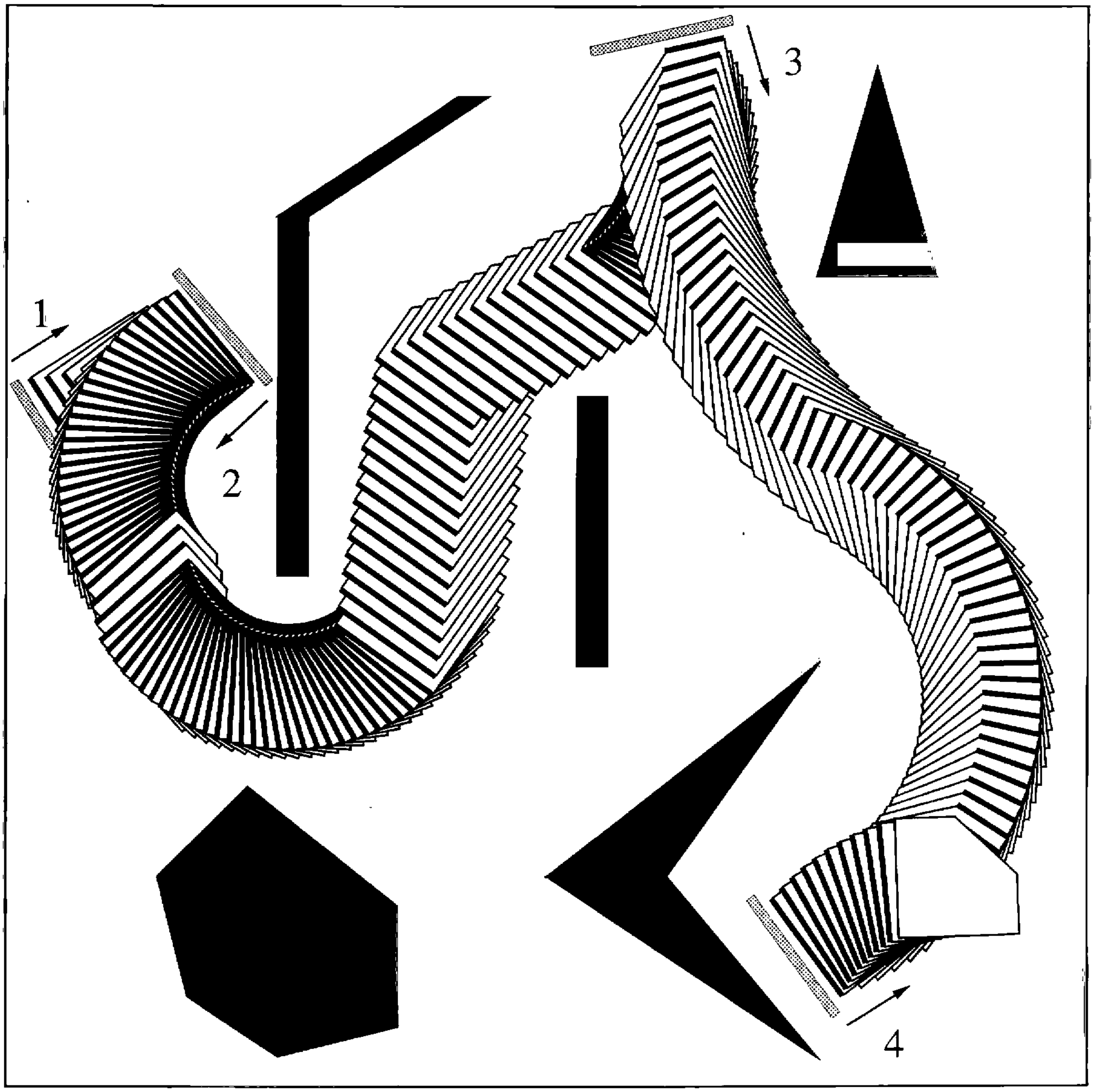

Fig. 18. A pushing plan found with cost function $a = 0$, $b = 1$, *and* $c = 10$.

We have assumed that the set of possible pushing contacts is specified by the user. An interesting problem is how to choose the set of pushing contacts based on knowledge of the pusher, the slider, the environment, and the initial and goal configurations. We have focused on line contact pushing in this article, but a similar approach can be used with other types of pushing contacts.

To manipulate an unknown object, a robot could estimate the friction parameters of the object (Yoshikawa and Kurisu 1991; Lynch 1993), perform the mechanics analysis, and use the results in the planner described here. Alternatively, the robot could empirically determine a set of stable pushing directions and plug these directly into the planner.

The pushing paths described in this article are stabilized by using more than one contact point between the pusher and the slider. No sensing is required: the inherent mechanics of the task essentially close a tight feedback loop. Nevertheless, unmodeled effects could cause the pusher to lose control of the motion of the object. The planner described here could be considered the feedforward component of a pushing control system.

5. Conclusion

A model of the mechanics of a task is a resource for the robot, just as actuators and sensors are resources. The effective use of frictional, gravitational, and dynamic forces can substitute for extra actuators; the expectation derived from a good model can minimize sensing requirements. We have used a quasistatic model to derive an algorithm to automatically find sensorless pushing plans.

As the mechanics model becomes more detailed, however, it becomes more challenging to assess the capabilities of a robot. The set of accessible configurations of the manipulated object is no longer just the set of configurations the end effector can reach. We must answer the fundamental question, "Can the object be pushed from here to there?" This article begins to answer that question by elucidating some of the controllability properties of objects pushed with point or line contact.

Acknowledgments

We thank Mike Erdmann, Mark Wheeler, and the anonymous reviewers for their helpful comments, and Costa Nikou for his work on the graphical user interface for the push planner. This work was supported by NSF grants IRI-9114208 and IRI-9318496.

References

Abell, T., and Erdmann, M. 1995. Stably supported rotations of a planar polygon with two frictionless contacts. *Proc. 1995 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 411–418.

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. 1974. *The Design and Analysis of Computer Algorithms*. Reading, MA: Addison-Wesley.

Aiyama, Y., Inaba, M., and Inoue, H. 1993. Pivoting: A new method of graspless manipulation of object by robot fingers. *Proc. 1993 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 136–143.

Akella, S., and Mason, M. T. 1992. Posing polygonal objects in the plane by pushing. In *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 2255–2262.

Alami, R., Simeon, T., and Laumond, J.-P. 1989. A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps. *Fifth Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 453–459.

Alexander, J. C., and Maddocks, J. H. 1993. Bounds on the friction-dominated motion of a pushed object. *Int. J. Robot. Res.* 12(3):231–248.

Arai, H., and Khatib, O. 1994. Experiments with dynamic skills. *Proc. 1994 Japan–USA Symp. on Flexible Automation*, pp. 81–84.

Balorda, Z. 1990. Reducing uncertainty of objects by robot pushing. *Proc. 1990 IEEE Int. Conf. Robotics and Automation*, pp. 1051–1056.

Balorda, Z. 1993. Automatic planning of robot pushing operations. *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Vol. 1, pp. 732–737.

Barraquand, J., and Latombe, J.-C. 1993. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica* 10:121–155.

Brokowski, M., Peshkin, M., and Goldberg, K. 1993. Curved fences for part alignment. *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Vol. 3, pp. 467–473.

Brost, R. C. 1988. Automatic grasp planning in the presence of uncertainty. *Int. J. Robot. Res.* 7(1):3–17.

Brost, R. C. 1992. Dynamic analysis of planar manipulation tasks. *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 2247–2254.

Christiansen, A. D. 1995. A machine learning perspective on modeling robot actions. In Goldberg, K., Halperin, D., Latombe, J-C., and Wilson, R. (eds.): *Algorithmic Foundations of Robotics*. Boston, MA: A. K. Peters, pp. 319–330.

Dacre-Wright, B., Laumond, J.-P., and Alami, R. 1992. Motion planning for a robot and a movable object amidst polygonal obstacles. *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 2474–2480.

Donald, B. R., Jennings, J., and Rus, D. 1993. Information invariants for cooperating autonomous mobile robots. *Sixth Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 29–48.

Dubins, L. E. 1957. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Am. J. Math.* 79:497–516.

Erdmann, M. A. 1984. On motion planning with uncertainty. Master's thesis, Department of Computer Science and Electrical Engineering, Massachusetts Institute of Technology.

Erdmann, M. A. 1994. On a representation of friction in configuration space. *Int. J. Robot. Res.* 13(3):240–271.

Erdmann, M. A., and Mason, M. T. 1988. An exploration of sensorless manipulation. *IEEE Trans. Robot. Automation* 4(4):369–379.

Gandolfo, F., Tistarelli, M., and Sandini, G. 1991. Visual monitoring of robot actions. *Proc. 1991 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 269–275.

Goldberg, K. Y. 1993. Orienting polygonal parts without sensors. *Algorithmica* 10:201–225.

Goldberg, K. Y. 1995. Completeness in robot motion planning. In Goldberg, K., Halperin, D., Latombe, J-C., and Wilson, R. (eds.): *Algorithmic Foundations of Robotics*. Boston, MA: A. K. Peters, pp. 419–429.

Goyal, S., Ruina, A., and Papadopoulos, J. 1991a. Planar sliding with dry friction. Part 1. Limit surface and moment function. *Wear* 143:307–330.

Goyal, S., Ruina, A., and Papadopoulos, J. 1991b. Planar sliding with dry friction. Part 2. Dynamics of motion. *Wear* 143:331–352.

Grünbaum, B. 1967. *Convex Polytopes*. London: Interscience.

Haynes, G. W., and Hermes, H. 1970. Nonlinear controllability via Lie theory. *SIAM J. Control* 8(4):450–460.

Hermann, R., and Krener, A. J. 1977. Nonlinear controllability and observability. *IEEE Trans. Auto. Control* AC-22(5):728–740.

Higuchi, T. 1985. Application of electromagnetic impulsive force to precise positioning tools in robot systems. *Second Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 281–285.

Hong, J., Lafferiere, G., Mishra, B., and Tan, X. 1990. Fine manipulation with multifinger hands. *Proc. 1990 IEEE Int. Conf. Robotics and Automation*, pp. 1568–1573.

Huang, W., Krotkov, E. P., and Mason, M. T. 1995. Impulsive manipulation. *Proc. 1995 IEEE Int. Conf. Robotics and Automation*, pp. 120–125.

Inaba, M., and Inoue, H. 1989. Vision-based robot programming. *Fifth Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 129–136.

Jacobs, P., and Canny, J. 1989. Planning smooth paths for mobile robots. *Proc. 1989 IEEE Int. Conf. Robotics and Automation*, pp. 2–7.

Jurdjevic, V. 1972. Certain controllability properties of analytic control systems. *SIAM J. Control* 10(2):354–360.

Jurdjevic, V., and Sussmann, H. J. 1972. Control systems on Lie groups. *Journal Differential Equations*, 12:313–329.

Koga, Y., and Latombe, J.-C. 1994. On multi-arm manipulation planning. *Proc. 1994 IEEE Int. Conf. Robotics and Automation*, Vol. 2, pp. 945–952.

Kurisu, M., and Yoshikawa, T. 1994. Trajectory planning for an object in pushing operation. *Proc. 1994 Japan–USA Symp. on Flexible Automation*, pp. 1009–1016.

Latombe, J.-C. 1991. *Robot Motion Planning*. Boston: Kluwer Academic Publishers.

Laumond, J.-P. 1986. Feasible trajectories for mobile robots with kinematic and environment constraints. *Proc. 1986 Int. Conf. Intelligent Autonomous Systems*, pp. 346–354.

Laumond, J.-P., Jacobs, P. E., Taix, M., and Murray, R. M. 1994. A motion planner for nonholonomic mobile robots. *IEEE Trans. Robot. Automation*, 10(5):577–593.

Li, Z., and Canny, J. 1990. Motion of two rigid bodies with rolling constraint. *IEEE Trans. Robot. Automation*, 6(1):62–72.

Lynch, K. M. 1992. The mechanics of fine manipulation by pushing. *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 2269–2276.

Lynch, K. M. 1993. Estimating the friction parameters of pushed objects. *Proc. 1993 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 186–193.

Lynch, K. M., Maekawa, H., and Tanie, K. 1992. Manipulation and active sensing by pushing using tactile feedback. *Proc. 1992 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 416–421.

Lynch, K. M. and Mason, M. T. 1996. Dynamic underactuated nonprehensile manipulation. *Proc. 1996 IEEE/RSJ Int. Conf. Intelligent Robots and Systems* (Forthcoming).

MacMillan, W. D. 1936. *Dynamics of Rigid Bodies*. New York: Dover.

Mani, M., and Wilson, W. 1985. A programmable orienting system for flat parts. *North American Manufacturing Research Institute Conference XIII.*

Markenscoff, X., Ni, L., and Papadimitriou, C. H. 1990. The geometry of grasping. *Int. J. Robot. Res.* 9(1):61–74.

Mason, M. T. 1986. Mechanics and planning of manipulator pushing operations. *Int. J. Robot. Res.* 5(3):53–71.

Mason, M. T. 1989. Compliant sliding of a block along a wall. *Proc. 1989 Int. Symp. Experimental Robotics*, pp. 568–578.

Mason, M. T., and Brost, R. C. 1986. Automatic grasp planning: An operation space approach. *Sixth Symp. on Theory and Practice of Robots and Manipulators.* Alma Press, pp. 321–328.

Mason, M. T., and Lynch, K. M. 1993. Dynamic manipulation. *Proc. 1993 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 152–159.

Mayeda, H., and Wakatsuki, Y. 1991. Strategies for pushing a 3D block along a wall. *Proc. 1991 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 461–466.

Mishra, B., Schwartz, J. T., and Sharir, M. 1987. On the existence and synthesis of multifinger positive grips. *Algorithmica* 2(4):541–558.

Miura, J. 1989. Integration of problem solving and learning in intelligent robots. *Fifth Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 13–20.

Narasimhan, S. 1995. Task-level strategies for robot tasks. Ph.D. thesis, Department of Computer Science and Electrical Engineering, Massachusetts Institute of Technology.

Nijmeijer, H., and van der Schaft, A. J. 1990. *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag.

Okawa, Y., and Yokoyama, K. 1992. Control of a mobile robot for the push-a-box operation. *Proc. 1992 IEEE Int. Conf. Robotics and Automation*, pp. 761–766.

Peshkin, M. A., and Sanderson, A. C. 1988a. Planning robotic manipulation strategies for workpieces that slide. *IEEE J. Robot. Automation* 4(5):524–531.

Peshkin, M. A., and Sanderson, A. C. 1988b. The motion of a pushed, sliding workpiece. *IEEE J. Robot. Automation* 4(6):569–598.

Peshkin, M. A., and Sanderson, A. C. 1989. Minimization of energy in quasi-static manipulation. *IEEE Trans. Robot. Automation* 5(1):53–60.

Prescott, J. 1923. *Mechanics of Particles and Rigid Bodies.* London: Longmans, Green, and Co.

Reeds, J. A., and Shepp, L. A. 1990. Optimal paths for a car that goes both forwards and backwards. *Pacific J. Math.* 145(2):367–393.

Rizzi, A. A., and Koditschek, D. E. 1993. Further progress in robot juggling: The spatial two-juggle. *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Vol. 3, pp. 919–924.

Salganicoff, M., Metta, G., Oddera, A., and Sandini, G. 1993a. A direct approach to vision guided manipulation. *Proc. 1993 Int. Conf. Advanced Robotics.*

Salganicoff, M., Metta, G., Oddera, A., and Sandini, G. 1993b. A vision-based learning method for pushing manipulation. *AAAI Fall Symp. on Machine Learning in Computer Vision.*

Salisbury, K., Eberman, B., Levin, M., and Townsend, W. 1987. The design and control of an experimental whole-arm manipulator. *Fourth Int. Symp. Robotics Res.* Cambridge, MA: MIT Press, pp. 233–241.

Sawasaki, N., Inaba, M., and Inoue, H. 1989. Tumbling objects using a multi-fingered robot. *20th Int. Symp. Industrial Robots and Robot Exhibition*, pp. 609–616.

Schaal, S., and Atkeson, C. G. 1993. Open loop stable control strategies for robot juggling. *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Vol. 3, pp. 913–918.

Sussmann, H. J. 1978. A sufficient condition for local controllability. *SIAM J. Control Optimization* 16(5):790–802.

Sussmann, H. J. 1983. Lie brackets, real analyticity and geometric control. In Brockett, R. W., Millman, R. S., and Sussmann, H. J. (eds.): *Differential Geometric Control Theory.* Boston, MA: Birkhäuser, pp. 1–116.

Sussmann, H. J. 1987. A general theorem on local controllability. *SIAM J. Control Optimization* 25(1):158–194.

Sussmann, H. J., and Jurdjevic, V. 1972. Controllability of nonlinear systems. *J. Differential Equations* 12:95–116.

Trinkle, J. C., Ram, R. C., Farahat, A. O., and Stiller, P. F. 1993. Dexterous manipulation planning and execution of an enveloped slippery workpiece. *Proc. 1993 IEEE Int. Conf. Robotics and Automation*, Vol. 2, pp. 442–448.

Wilfong, G. 1988. Motion planning in the presence of movable obstacles. *Proc. 4th ACM Symp. Computational Geometry*, pp. 279–288.

Yoshikawa, T., and Kurisu, M. 1991. Identification of the center of friction from pushing an object by a mobile robot. *Proc. 1991 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 449–454.

Zrimec, T. 1990. Towards autonomous learning of behavior by a robot. Ph.D. thesis, Department of Computer and Information Science, University of Ljubljana.

Zumel, N. B., and Erdmann, M. A. 1994. Balancing of a planar bouncing object. *Proc. 1994 IEEE Int. Conf. Robotics and Automation*, pp. 2949–2954.