Karl-Friedrich Böhringer

University of Washington, Seattle
Department of Electrical Engineering
234 EE/CSE Building
Box 352500
Seattle, WA 98195-2500
karl@ee.washington.edu
www.ee.washington.edu/faculty/karl

Bruce Randall Donald

Dartmouth College
Department of Computer Science
6211 Sudikoff Laboratory
Hanover, New Hampshire 03755-3510, USA
brd@cs.darmouth.edu www.cs.dartmouth.edu/~brd

Noel C. MacDonald

Cornell University
Department of Electrical Engineering and
Cornell Nanofabrication Facility
408 Phillips Hall
Ithaca, New York 14853, USA
mmacd@ee.cornell.edu
www.engr.cornell.edu/ee/MacDonald.html

Programmable Force Fields for Distributed Manipulation, with Applications to MEMS Actuator Arrays and Vibratory Parts Feeders*

Abstract

Programmable force vector fields can be used to control a variety of flexible planar parts feeders such as massively parallel microactuator arrays or transversely vibrating (macroscopic) plates. These new automation designs promise great flexibility, speed, and dexterity—we believe they may be employed to position, orient, singulate, sort, feed, and assemble parts. However, since they have only recently been invented, programming and controlling them for manipulation tasks is challenging. When a part is placed on our devices, the programmed vector field induces a force and moment upon it. Over time, the part may come to rest in a dynamic equilibrium state. By chaining sequences of force fields, the equilibrium states of a part in the field may be cascaded to obtain a desired final state. The resulting strategies require no sensing, and enjoy efficient planning algorithms.

This paper begins by describing new experimental devices that can implement programmable force fields. In particular, we describe our progress in building the M-CHIP (Manipulation CHIP), a massively parallel array of programmable micromotion pixels. Both the M-CHIP and other microarray devices, as well as macroscopic devices such as transversely vibrating plates, may be programmed with vector fields, and their behavior predicted and controlled using our equilibrium analysis. *We demonstrate lower bounds (i.e., impossibility results) on what the devices* cannot *do, and results on a* classification of control strategies *yielding design criteria by which well-behaved manipulation strategies may be developed. We provide sufficient conditions for programmable fields to induce well-behaved equilibria on every part placed on our devices. We define* composition operators *to build complex strategies from simple ones, and show the resulting fields are also well behaved. We discuss whether fields outside this class can be useful and free of pathology.*

Using these tools, we describe new manipulation algorithms. In particular, we improve existing planning algorithms by a quadratic factor, and the plan length by a linear factor. Using our new and improved strategies, we show how to simultaneously orient and pose any part, without sensing, from an arbitrary initial configuration. We relax earlier dynamic and mechanical assumptions to obtain more robust and flexible strategies.

Finally, we consider parts feeders that can only implement a very limited "vocabulary" of vector fields (as opposed to the pixel-wise programmability assumed above). We show how to plan and execute parts posing and orienting strategies for these devices, but with a significant increase in planning complexity and some sacrifice in completeness guarantees. We discuss the trade-off between mechanical complexity and planning complexity.

*An earlier, much shorter version of this paper was presented at the Workshop on Algorithmic Foundations of Robotics (Toulouse, France, July 1996).

The International Journal of Robotics Research
Vol. 18, No. 2, February 1999, pp. xxx-xxx,

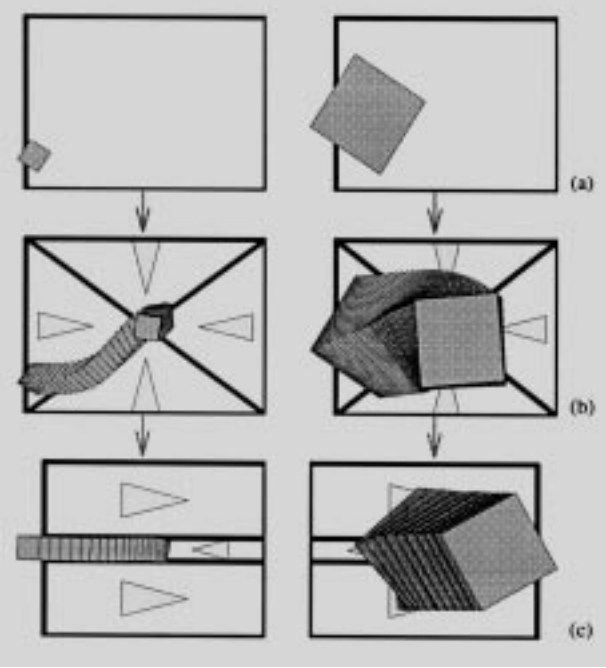

Fig. 1. Sensorless sorting using force vector fields: parts of different sizes are first centered, then subsequently separated, depending on their size.

1. Introduction

Programmable force fields offer a fundamentally new approach to automated parts manipulation. Instead of handling a part directly (e.g., with a robot gripper), a force field surrounding the part causes it to move. Programmable force fields promise great flexibility, speed, and dexterity for a wide variety of tasks such as parts orienting, positioning, singulating, sorting, feeding, and assembly. Recently, several devices have been invented that can implement programmable force fields: in particular, actuator arrays fabricated with Micro Electro Mechanical System (MEMS) technology, as well as macroscopic vibrating plates. These new automation designs permit distributed, parallel, nonprehensile, sensorless manipulation tasks that make them particularly attractive for handling batch microfabricated parts, whose small dimensions and large numbers would prohibit conventional pick-and-place operations.

A wealth of geometric and algorithmic problems arise in the control and programming of manipulation systems with many independent actuators. The theory of programmable force fields represents the first systematic, computational attack on massively parallel distributed manipulation based on geometric and physical reasoning. The goal of this paper is to develop a science base for manipulation using programmable force fields, and to demonstrate experiments with prototype devices that support this theory. We present combinatorially precise planning algorithms that synthesize strategies for controlling and coordinating a very large number of distributed actuators in a principled, task-level fashion.

When a part is placed on such a device, the programmed vector field induces a force and moment upon it. Over time, the part may come to rest in a dynamic equilibrium state. In principle, we have tremendous flexibility in choosing the vector field, since using, e.g., MEMS array technologies, the force field may be programmed pixel-wise. Hence, we have a lot of control over the resulting equilibrium states. By chaining sequences of vector fields, the equilibria may be cascaded to obtain a desired final state—for example, this state may represent a unique orientation or pose of the part. A system with such a behavior exhibits the *feeding property* (Akella et al. 1995):

A system has the feeding property over a set of parts $\mathcal{P}$ and a set of initial configurations $\mathcal{I}$ if, given any part $P \in \mathcal{P}$, there is some output configuration $\mathbf{q}$ such that the system can move P to $\mathbf{q}$ from any location in $\mathcal{I}$.

Our work on programmable vector fields is related to nonprehensile manipulation [Donald, Jennings, and Rus 1995; Zumel and Erdmann 1996; Erdmann and Mason 1996; Erdmann l996]: in both cases, parts are manipulated without form or force closure.

This paper describes our experimental devices, a technique for analyzing them called *equilibrium analysis*, lower bounds (i.e., impossibility results) on what the devices *cannot* do, and results of a classification of control strategies yielding design criteria for useful manipulation strategies. Then we describe new manipulation algorithms using these tools. In particular, we improve earlier planning algorithms by a quadratic factor, show how to simultaneously orient and pose a part, and relax dynamic and mechanical assumptions to obtain more robust and flexible strategies.

One corollary of our results is a method for coordinating the actions of a large distributed actuation system. Such systems comprise arrays with up to tens of thousands of independently servoable actuator cells, which we call *motion pixels*. We show how these systems can be programmed in a fine-grained, SIMD (single instruction multiple data) fashion to exert force fields on the manipulated object, thereby accomplishing massively parallel distributed manipulation. Moreover, the theory of programmable force fields gives a method for controlling a large number of distributed actuators in a principled, geometric, task-level fashion. Whereas many control theories for multiple independent actuators break down as the number of actuators becomes large, our systems should only become more robust as the actuators become denser and more numerous.

The theory developed in this paper is applicable to any controllable array capable of generating force vector fields, and it is independent of the specific device hardware. We have tested it thoroughly in collaboration with J. Suh and G. Kovacs on a MEMS actuator array developed at Stanford (Böhringer et al. 1997c). This microcilia device consists of a 16×16 array of

motion pixels, which covers an area of about 2 cm $\times$ 2 cm. Each pixel consists of four thermobimorph actuators. Actuators in each direction can be controlled independently by a graphical user interface on a personal computer. Böhringer and coworkers (1997b) reported on experiments in sensorless manipulation with the microcilia device. Small chips were placed at arbitrary initial positions on the array and were translated, rotated, centered, and aligned by the array without sensor feedback. These experiments constitute strong evidence in support of our theory of sensorless manipulation.

In this paper, we focus on the theoretical foundations of manipulation with programmable force fields. We pose the question, Which force fields are suitable for manipulation strategies? In particular, we ask whether the fields may be *classified*. That is, can we characterize all those force fields in which every part has stable equilibria? While this question has been well studied for a point mass in a field, the issue is more subtle when lifted to a body with finite area, due to the moment covector. To answer, we first demonstrate impossibility results, in the form of "lower bounds": there exist perfectly plausible fields that induce *no* stable equilibrium in simple parts.

Fortunately, there is also good news. We present conditions for fields to induce well-behaved equilibria,
by exploiting the theory of potential fields. While potential fields have been widely used in robot control (Khatib 1986; Koditschek and Rimon 1988; Rimon and Koditschek 1992; Reif and Wang 1995], microactuator arrays present us with the ability to explicitly program the applied force at every point in a vector field. Whereas previous work has developed control strategies with artificial potential fields, our fields are nonartificial (i.e., physical). Artificial potential fields require a tight feedback loop, in which at each clock tick, the robot senses its state and looks up a control (i.e., a vector) using a state-indexed navigation function (i.e., a vector field). In contrast, physical potential fields employ no sensing, and the motion of the manipulated object evolves in an open-loop manner (for example, like a particle in a gravity field). This alone makes our application of potential-field theory to microdevices unique and novel. Moreover, such fields can be composed using addition, sequential composition, "parallel" composition by superposition of controls, or by a new kind of "morphing" of control signals, which we will define.

Previous results on array manipulation strategies may be formalized using equilibrium analysis. Böhringer and colleagues proposed a family of control strategies called *squeeze patterns*, and a planning algorithm for parts orientation. This first result proved an $O(n^2)$ upper bound on the number E of orientation equilibria of a nonpathological (see Section 3.2) planar part with n vertices. This yields an $O(E^2) = O(n^4)$ planning algorithm to uniquely orient a part, under certain geometric, dynamic, and mechanical assumptions. In this paper, we argue that this bound on equilibria appears tight. This results in a high planning and execution complexity.

Using our equilibrium analysis, we introduce *radial* fields, which satisfy our stability property. Radial fields can then be combined with squeeze fields. We show this has several benefits:

1. the number of equilibria drops to $E = O(n)$;
2. the planning complexity drops to $O(E^2) = O(n^2)$;
3. throughout the strategy execution, every part rotates about one fixed, unique point (after the first step); and
4. this means that we can dispense with one critical assumption (called 2PHASE by Böhringer and coworkers (1994a)): we no longer need to assume that the translational and rotational motions induced by the array interact in a "quasi-static" and "sequential" manner.

We motivate our results by beginning with a description of the experimental devices we are interested in programming. In particular, we describe our progress in building the M-CHIP (Manipulation CHIP), a massively parallel array of programmable micromotion pixels. As proof of the concept, we demonstrate a prototype M-CHIP containing up to 15,000 silicon actuators in 1 in^2. Our strategies are also applicable to macroscopic partsfeeders. We describe a planar, vibratory orienting and manipulation device that also uses our novel strategies.

Both of these devices portend several key practical issues. First, the strategies employed by our improved algorithms and analysis require significant mechanical and control complexity—even though they require no sensing. While we believe such mechanisms are feasible to build using the silicon MEMS technologies we advocate, it is undeniable that no such device exists yet (the M-CHIP has pixel-wise programmability, but the first generation does not have sufficient directional resolution to implement highly accurate radial strategies). For this reason, we introduce and analyze strategies composed of field sequences that we know are implementable using current (microscopic or macroscopic) technology. Each strategy is a sequence of pairs of squeezes satisfying certain "orthogonality" properties. Under these assumptions, we can ensure:

1. equilibrium stability,
2. relaxed mechanical and dynamical assumptions (the same as point 4, above), and
3. complexity and completeness guarantees.

The framework is quite general, and applies to any set of primitive operations satisfying certain "finite equilibrium" properties (which we define)—hence it has broad applicability to a wide range of devices. In particular, we view the restricted class of fields as a *vocabulary* and its rules of composition as a *grammar*, resulting in a language of manipulation strategies. Using our grammar, the resulting strategies are guaranteed to be well-behaved.

Finally, both our radial strategies and our finite manipulation grammar have the following advantage over previous manipulation algorithms for programmable vector fields:

previous algorithms such as those described by Böhringer and colleagues (1994a, 1996a) guarantee to uniquely orient a part, but the translational position of the part is unknown at the strategy's termination. Both of our new algorithms guarantee to position the part uniquely (up to part symmetry) in translation as well as orientation space. Like the algorithms in Böhringer's work (1994a, 1996a), the new algorithms require no sensing, and work from any initial configuration to uniquely pose the part. In particular, the initial configuration is never known to the (sensorless) execution system, which functions in an open-loop manner.

The complexity and completeness guarantees we obtain for manipulation grammars are considerably weaker than for the ideal radial strategies. For radial strategies, we show that any nonpathological planar part with finite area contact can be placed in a unique pose in $O(E) = O(n)$ steps. Under the simplified manipulation grammar, our planner is guaranteed to find a strategy if one exists (if one does not exist, the planner will signal this). However, it is not known whether there exists a strategy for every part. This lack of completeness of manipulation grammar strategies stands in contrast to the complete general squeeze and radial algorithms for which a guaranteed strategy exists for all parts. Moreover, the planning algorithm is worst-case exponential instead of merely quadratic.

Finally, the desire to implement complicated fields raises the question of control uncertainty. We close by describing how families of potential functions can be used to represent control uncertainty and analyzed for their impact on equilibria, and we give an outlook on still-open problems and future work.

2. Experimental Apparatus: Parts Feeders

It is often extremely costly to maintain part order throughout the manufacture cycle. For example, instead of keeping parts in pallets, they are often delivered in bags or boxes, whence they must be picked out and sorted. A parts feeder is a machine that orients such parts before they are fed to an assembly station. Currently, the design of parts feeders is a black art that is responsible for up to 30% of the cost and 50% of work-cell failures (Nevins and Whitney 1978; Boothroyd, Poli, and Murch 1982; Farnum and Davis 1986; Schroer 1987; Singer and Seering 1987). "*The real problem is not part transfer but part orientation,*" according to Frank Riley of the Bodine Corporation (Riley 1983, p. 316, his italics). Thus, although part feeding accounts for a large portion of assembly cost, there is not much scientific basis for automating the process.

The most common type of parts feeder is the *vibratory bowl feeder*, where parts in a bowl are vibrated using a rotary motion, so that they climb a helical track. As they climb, a sequence of baffles and cutouts in the track create a mechanical "filter" that causes parts in all but one orientation to fall back into the bowl for another attempt at running the gauntlet (Boothroyd, Poli, and Murch 1982; Riley 1983; Sandler 1991).

Sony's APOS parts feeder (Hitakawa 1988) uses an array of nests (silhouette traps) cut into a vibrating plate. The nests and the vibratory motion are designed so that the part will remain in the nest only in one particular orientation. By tilting the plate and letting parts flow across it, the nests eventually fill up with parts in the desired orientation. Although the vibratory motion is under software control, specialized mechanical nests must be designed for each part (Moncevicz, Jakiela, and Ulrich 1991).

The reason for the success of vibratory bowl feeders and the Sony APOS system is the underlying principle of *sensorless manipulation* (Erdmann and Mason 1988) that allows parts positioning and orienting without sensor feedback. This principle is even more important at small scales, because sensor data will be less accurate and more difficult to obtain. The APOS system or bowl feeders are unlikely to work in the micro domain: instead, novel device designs for micromanipulation tasks are required. The theory of sensorless manipulation is the science base for developing and controlling such devices.

Reducing the amount of required sensing is an example of *minimalism* (Canny and Goldberg 1994; Böhringer et al. 1995b), which pursues the following agenda: for a given robot task, find the minimal configuration of resources required to solve the task. Minimalism is interesting, because doing task A without resource B proves that B is somehow inessential to the information structure of the task. In robotics, minimalism has become increasingly influential. Raibert and colleagues (1993) showed that walking and running machines could be built without static stability. Erdmann and Mason (1988) showed how to do dexterous manipulation without sensing. McGeer (1990) built a biped, kneed walker without sensors, computers, or actuators. Canny and Goldberg (1994) argued that minimalism has a long tradition in industrial manufacturing, and developed geometric algorithms for orienting parts using simple grippers and accurate, low-cost light beams. Brooks (1986) developed online algorithms that rely less extensively on planning and world models. Donald, Jennings, and Rus (1995) and Böhringer et al. (1995b) have built distributed teams of mobile robots that cooperate in manipulation without explicit communication. We intend to use these results for our experiments in micromanipulation, and to examine how they relate to our theoretical proofs of minimalist systems.

2.1. Microfabricated Actuator Arrays

A wide variety of micromechanical structures (devices with features in the μm range) has been built recently by using processing techniques known from the VLSI industry (see, for example, the work of Gabriel (1995), MacDonald and colleagues (1997) and MacDonald (forthcoming). Various

microsensors and microactuators have been shown to perform successfully; e.g., a single-chip air-bag sensor is commercially available (Analog Devices 1991), and video projections using an integrated, monolithic mirror array have been demonstrated recently (Sampsell 1993) and are now starting to replace conventional projection systems. A fully integrated scanning tunneling microscope (STM) has been developed in our group (Xu, Miller, and MacDonald 1995; MacDonald et al. 1997). However, the fabrication, control, and programming of microdevices that can interact and actively change their environment remains challenging.

Problems arise from:

1. unknown material properties and the lack of adequate models for mechanisms at very small scales,
2. the limited range of motion and force that can be generated with microactuators,
3. the lack of sufficient sensor information with regard to manipulation tasks, and
4. design limitations and geometric tolerances due to the fabrication process.

Several MEMS researchers, among others (Fujita 1993; Storment et al. 1994; Liu and Will 1995; Jacobson et al. 1995; Suh et al. 1996) have proposed MEMS manipulator arrays. For an overview, see the work of Liu and Will (1995) or Böhringer and colleagues (1994a, 1994b).

Our arrays (Fig. 2) are fabricated using a SCREAM (Single-Crystal Silicon Reactive Etching and Metallization) process developed in the Cornell Nanofabrication Facility (Zhang and MacDonald 1992, Shaw, Zhang, and MacDonald 1993). The SCREAM process is low temperature, and does not interfere with traditional VLSI (Shaw and MacDonald 1996). Hence it opens the door to building monolithic microelectromechanical systems with integrated microactuators and control circuitry on the same wafer.

One of the goals of research in microactuators is to develop devices for manipulating other small components; for example, to accurately position micromachined components for inspection or assembly purposes. Fabrication constraints limit the design of most of these components (usually small chiplets made from silicon wafers) to extruded planar shapes, so manipulation in the plane is sufficient for many applications. For example, a microactuator array has been successfully employed to replace a 3-DOF stage in a scanning electron microscope (SEM) (Darling et al. 1997).

Our design is based on microfabricated torsional resonators (Mihailovich et al. 1993; Mihailovich and MacDonald 1996). Each unit device consists of a rectangular grid etched out of single-crystal silicon suspended by two rods that act as torsional springs (Fig. 3). The grid is about 200 μm long, and extends 120 μm on each side of the rod. The rods are 150 μm long. The current asymmetrical design has 5-μm high protruding tips on one side of the grid that make contact with an object lying on top of the actuator (Fig. 4). The other side

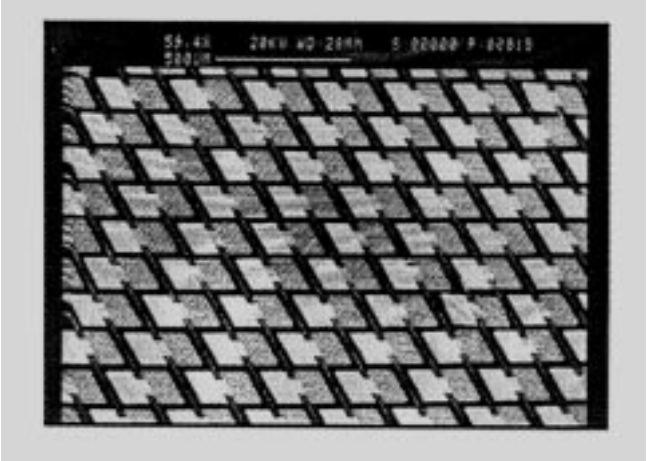

Fig. 2. A prototype M-CHIP fabricated in 1995: a large unidirectional actuator array (viewed via scanning electron microscopy). Each actuator is 180 $\times$ 240 μm in size. Detail from a 1 in^2 array with more than 15,000 actuators (For more pictures on device design and fabrication, see the World Wide Web at http://www.cs.cornell.edu/home/karl/MicroActuators.)

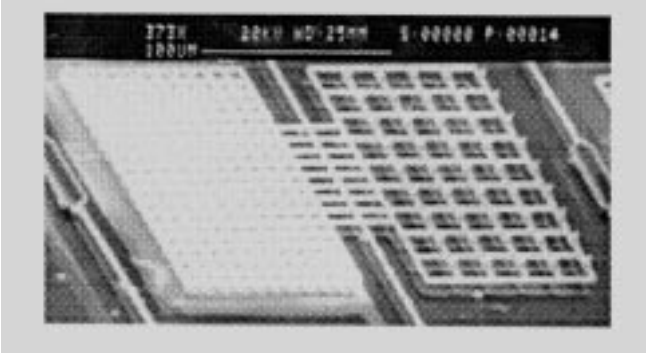

Fig. 3. Released asymmetric actuator for the M-CHIP (viewed via scanning electron microscopy): a dense grid (10 μm spacing) with an aluminum electrode underneath (left); a grid with 5-μm high poles (right).

of the actuator consists of a denser grid above an aluminum electrode. If a voltage is applied between the silicon substrate and the electrode, the dense grid above the electrode is pulled downward by the resulting electrostatic force. Simultaneously, the other side of the device (with the tips) is deflected several μm out of the plane. Hence, an object can be lifted and pushed sideways by the actuator.

Because of its low inertia (resonance in the high-kHz range), the device can be driven in a wide frequency range from DC to several 100 kHz AC. Our actuators need not be operated at resonance: they can also be servoed to periodically "hit" an object on top, thereby applying both lateral and vertical forces. Our calculations, simulations, and experiments have shown that the force generated with a torsional actuator

is approximately 10 μN, which corresponds to a force-per-area ratio of 100 μN/mm^2, which is large enough to levitate a piece of paper (1 μN/mm^2) or a silicon wafer (10 μN/mm^2).

Each actuator can generate motion in one specific direction if it is activated; otherwise, it acts as a passive frictional contact. Figure 2 shows a small section of such a unidirectional actuator array, which consists of more than 15,000 individual actuators. The combination and selective activation of several actuators with different motion bias allows us to generate various motions in discrete directions, spanning the plane (Fig. 5).

The microscopic features of these actuators pose a possible disadvantage, which may make them less useful in harsh or dirty environments. Macroscopic objects and forces can easily damage microactuators. For example, careful handling is required when placing objects on the array. However, silicon is a surprisingly flexible material at microscopic scales (Peterson 1982), and extremely large elastic deformations are possible without structural damage (Taher, Saif, and MacDonald 1995). Another concern are dust particles that could jam the microactuators. As a remedy, tiny venting holes can be etched from the backside of the substrate, such that dust particles are removed by a constant flow of air. Such air jets are also useful for levitating or manipulating objects (Pister Fearing, and Howe 1990; Konishi and Fujita 1993).

The fabrication process and mechanism analysis have been described in more detail in other works (Böhringer et al. 1994a, 1994b; Böhringer, Donald, and MacDonald 1996b).

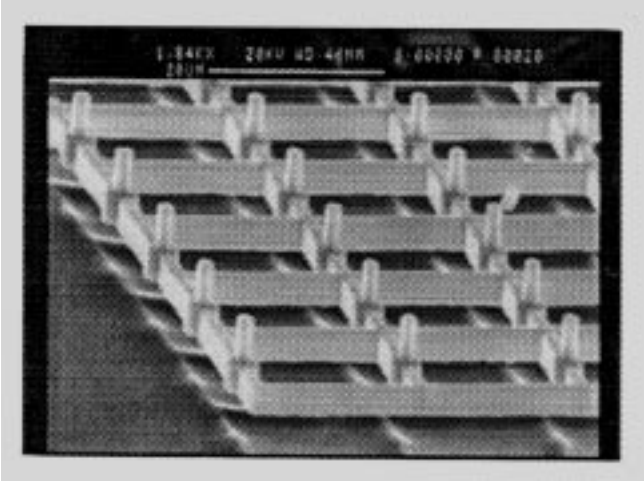

Fig. 4. Released M-CHIP actuators consisting of single-crystal silicon with 5-μm high tips.

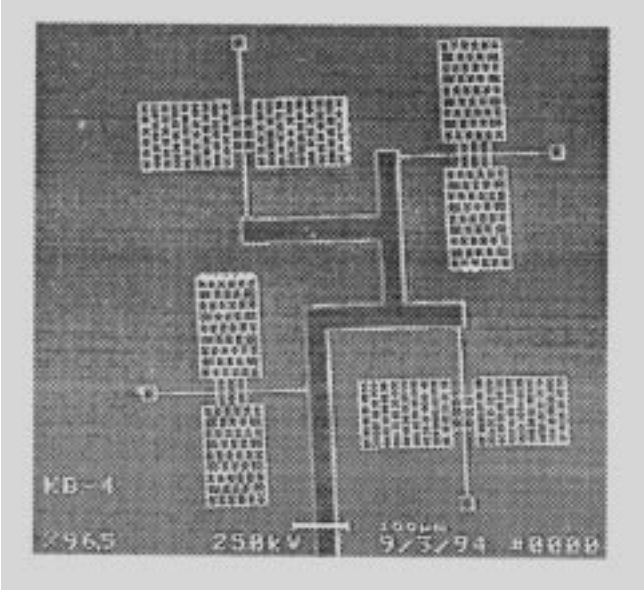

Fig. 5. Released M-CHIP prototype motion pixel, consisting of actuators oriented in four different directions.

2.2. Macroscopic Vibratory Parts Feeder

Böhringer and colleagues (1995a) have presented a device that uses the force field created by transverse vibrations of a plate to position and align parts. The device consists of an aluminum plate that is attached to a commercially available electrodynamic vibration generator,[1] with a linear travel of 0.02m, and the capability to produce a force of up to 500 N (Fig. 6). The input signal, specifying the waveform corresponding to the desired oscillations, is fed to a single-coil armature, which moves in a constant field produced by a ceramic permanent magnet in a center-gap configuration.

For low amplitudes and frequencies, the plate moves longitudinally with no perceptible transverse vibrations. However,

1. Model VT-100G, Vibration Test Systems, Akron, Ohio, USA.

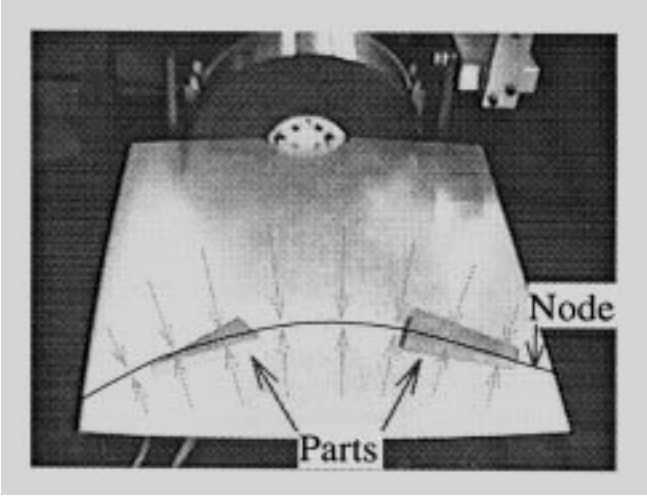

Fig. 6. Vibratory plate parts feeder: an aluminum plate (size 50 cm $\times$ 40 cm) exhibits a vibratory minimum. Parts are attracted to this nodal line, and reach equilibrium there. (See also the World Wide Web at www.ee.washington.edu/faculty/karl/Research/VibratoryPlate.)

as the frequency of oscillations is increased, transverse vibrations of the plate become more pronounced. The resulting motion is similar to the forced transverse vibration of a rectangular plate, clamped on one edge and free along the other three sides. This vibratory motion creates a force field in which particles are attracted to locations with minimal vibration, called the *nodal lines*. This field can be programmed by changing the frequency, or by employing clamps as programmable fixtures that create various vibratory nodes.

Figure 6 shows two parts, shaped like a triangle and a trapezoid, after they have reached their stable poses. To better illustrate the orienting effect, the curve showing the nodal line has been drawn by hand. Note that this device can only use the finite manipulation grammar described in Section 6.2, since it can only generate a constrained set of vibratory patterns, and cannot implement radial strategies.

3. Equilibrium Analysis for Programmable Vector Fields

For the generation of manipulation strategies with programmable vector fields, it is essential to be able to predict the motion of a part in the field. Particularly important is determining the stable equilibrium poses that a part can reach in which all forces and moments are balanced. This *equilibrium analysis* was introduced in our short conference paper (Böhringer et al. 1994a), where we presented a theory of manipulation for programmable vector fields, and an algorithm that generates manipulation strategies to orient polygonal parts without sensor feedback using a sequence of *squeeze fields*. We now review the algorithm from that work and give a detailed proof of its complexity bounds. The tools developed here are essential to understanding our new and improved results, and will be used throughout this paper to develop complexity bounds for our distributed manipulation algorithms.

In general, we assume that the dynamics of a part moving in the force field is governed by first-order dynamics. This assumption is based on extensive experimentation with the devices presented in Section 2. In a first-order system, the velocity of a part is directly proportional to the force acting on it. Basically, it is a rigid-body dynamical system that is heavily damped.

3.1. Squeeze Fields and Equilibria

In the work of Böhringer and colleagues (1994a), we proposed a family of control strategies called squeeze fields and a planning algorithm for parts orientation.

DEFINITION 1. Assume l is a straight line through the origin. A *squeeze field* f is a two-dimensional force-vector field defined as follows:

1. if $z \in \mathbb{R}^2$ lies on l, then $f(z) = 0$; and
2. if z does not lie on l,then $f(z)$ is the unit vector normal to l and pointing toward l.

We refer to the line l as the *squeeze line*, because l lies in the center of the squeeze field. See Figure 7 for examples of squeeze fields.

Assuming quasi-static motion, an object will move perpendicularly toward the line l and come to rest there. We are interested in the motion of an arbitrarily shaped (not necessarily small) part P. Let us call P_1, P_2 the regions of P that lie to the left and to the right of l, respectively, and c_1, c_2 their centers of area. In a rest position, both translational and rotational forces must be in equilibrium. We obtain the following two conditions:

1. The areas P_1 and P_2 must be equal, and
2. The vector $c_2 - c_1$ must be normal to l.

Part P has a translational motion component that is normal to l if condition 1 does not hold, and P has a rotational motion component if condition 2 does not hold (see Fig. 8). This assumes a uniform force distribution over the surface of P, which is a reasonable assumption for a flat part that is in contact with a large number of elastic actuators.

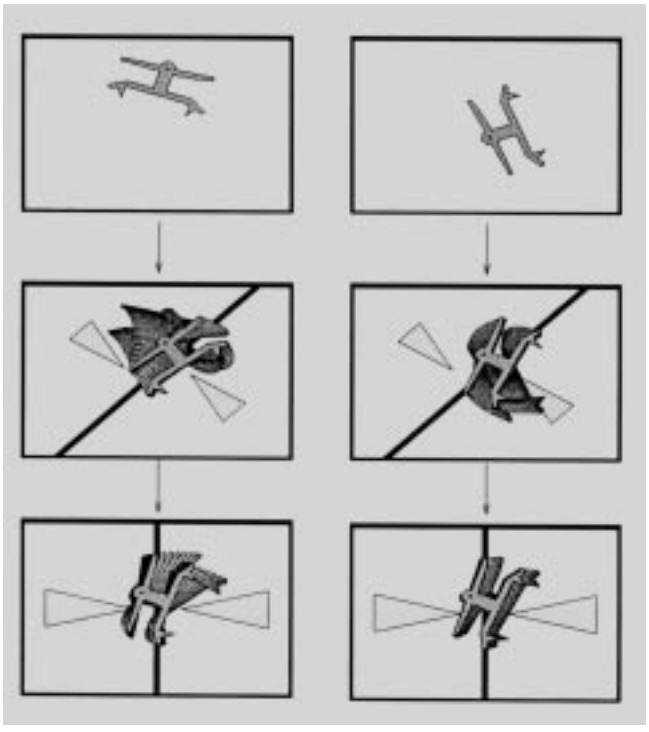

Fig. 7. Sensorless parts orienting using force-vector fields: the part reaches unique orientation after two subsequent squeezes. There exist such orientating strategies for all polygonal parts. (See the World Wide Web at www.ee.washington.edu/faculty/karl/PFF for an animated simulation.)

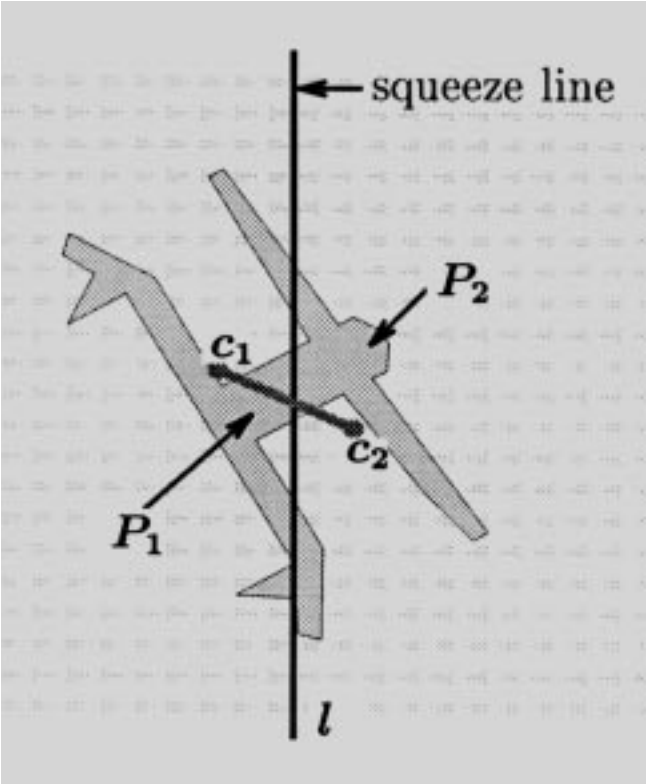

Fig. 8. Equilibrium condition: to balance the force and moment acting on P in a unit squeeze field, the two areas P_1 and P_2 must be equal (i.e., l must be a bisector), and the line connecting the centers of area c_1 and c_2 must be perpendicular to the node line.

DEFINITION 2. A part P is in *translation equilibrium* if the forces acting on P are balanced; P is in *orientation equilibrium* if the moments acting on P are balanced. *Total equilibrium* is simultaneous translation and orientation equilibrium.

Let (x_0, y_0, θ_0) be an equilibrium pose of P. (x_0, y_0) is the corresponding translation equilibrium, and θ_0 is the corresponding orientation equilibrium.

Note that conditions 1 and 2 do *not* imply that in equilibrium, the center of area of P has to coincide with the squeeze line l. For example, consider a large and a small square connected by a long rod of negligible width (Fig. 9). If the rod is long enough, the center of area will lie outside of the large square. However, in equilibrium, the squeeze line l will always intersect the large square.

3.2. Polygon Bisectors and Complexity

Consider a polygonal part P in a unit squeeze field, as described in Section 3.1. In this section, we describe how to determine the orientations θ_i in which P achieves equilibrium. This construction will show that equilibria *always exist*, as long as the contact areas have finite size, and that for connected parts, the orientation equilibria are discrete. More precisely, if a connected part is in equilibrium in a squeeze field, there are discrete values for its orientation and its offset from the center of the squeeze line. The equilibrium is of course independent of its position along the squeeze line. Hence, in the remainder of Section 3, when using the term "discrete equilibria," we mean that the orientation and offset of the part is discrete. We will derive upper bounds on the number of these discrete equilibria.

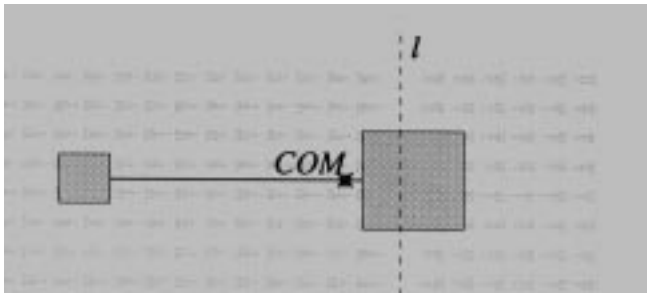

Fig. 9. A part consisting of two squares connected by a long, thin rod. The part is in total equilibrium, but its COM does not coincide with the squeeze line l.

DEFINITION 3. A bisector of a polygon P is a line that cuts P into two regions of equal area.

PROPOSITION 1. Let P be a polygon whose interior is connected. There exist $O(kn^2)$ bisectors such that P is in equilibrium when placed in a squeeze field where the bisector coincides with the squeeze line. n is the part complexity measured as the number of polygon vertices, and k denotes the maximum number of polygon edges that a bisector can cross.

If P is convex, then the number of bisectors is bounded by $O(n)$.

For most part geometries, k is a small constant.[2] However, in the worst case, pathological parts can reach $k = O(n)$. A spiral-shaped part (e.g., a rectilinear part) would be an example for such a pathological case, because every bisector intersects $O(n)$ polygon edges.

LEMMA 1. Given a polygon P and a line $l : y = mx + c$, let n be the number of vertices of P:

1. there exist $O(n^2)$ combinatorially different ways how a line l can intersect P;
2. let a and b be the intersections of bisector l with the convex hull of P. As m varies from $-\infty$ to $+\infty$, a and b progress monotonically counterclockwise about the convex hull of P; and
3. if the interior of P is connected, then there exists a unique bisector of P for every $m \in \mathbb{R}$.

2. In particular, in an earlier work (Böhringer et al. 1994a), we assumed that $k = O(1)$.

Combinatorially equivalent intersections of polygon P are all those placements of the intersecting line l such that the sets of left and right polygon vertices are fixed. A necessary condition for combinatorial equivalence is that l intersects the same ordered set of polygon edges.

Proof. There are $O(n^2)$ different placements for l such that it coincides with more than one vertex of P. Hence, all placements of l fall into one of $O(n^2)$ combinatorially equivalent classes. This was proven by Díaz and O'Rourke (1990, Lemma 3.1).

Assume l is a bisector of P with a fixed slope m. Since the interior of P is connected, the intersection between l and P must be a line segment of nonzero length. Hence a translation of l (e.g., toward the left) will cause a strictly monotonous decrease in the left-area segment of P, and vice versa. Therefore, the bisector placement of l for a given slope m is unique. □

Consider the bisector l of polygon P for changing m values, as described in Lemma 5. The intersections of l with the convex hull of P, a and b, progress monotonically about the convex hull. In general, this progression corresponds to a rotation and a translation of l.

In the following proof for Proposition 1, we investigate the relationship between the location of the bisector and the corresponding left and right areas of P and its respective centers of area.

This will allow us to show that for combinatorially equivalent bisector placements, there are only a finite number of possible equilibria, and this number is bounded by $O(k)$, where $k \leq n$ is the number of polygon edges that the bisector intersects.

Proof (Proposition 1). Consider two combinatorially equivalent placements of bisector l on polygon P. We will show that the number of equilibria for this bisector placement is bounded by $O(k)$. Since there are $O(n^2)$ such placements for P (see Lemma 1), the total number of equilibria will be $O(kn^2)$.

Rotating the Bisector. Consider the line l and a point s that lies on l (Fig. 10). The direction of l is given by a vector r. Assume for now that the line l intersects two edges of the polygon P in the points r_1 and r_2. Also assume that these edges have directions a_1 and a_2. Now consider another line l' with direction r' that intersects l in s. Assume that l and l' have combinatorially equivalent intersections with polygon P, and that l' intersects the polygon edges in r'_1 and r'_2. Let us write $r_i = s + \rho_i r$ and $r'_i = s + \rho'_i r'$ for $i = 1, 2$. Then the polygon area between l and l' is

$$A = \frac{1}{2}\left(\rho'_2\rho_2 - \rho'_1\rho_1\right)\left(r' \times r\right).$$

In the general case where l and l' intersect multiple edges of some arbitrary polygon P at points $r_1, r_2, \ldots, r_k$ and $r'_1, r'_2, \ldots, r'_k$ (k even), the polygon area between l and l' is

$$A = \frac{1}{2}\left(r' \times r\right)\sum_{i=1}^{k}(-1)^i \rho'_i\rho_i.$$

Without loss of generality, let $\rho_k \neq 0$. Then r' can be written as $r' = r + \alpha a_k$ for some $\alpha \in \mathbb{R}$, and the above equation becomes

$$\begin{aligned} A &= \frac{1}{2}\left((r + \alpha a_k) \times r\right)\sum_{i=1}^{k}(-1)^i \rho'_i\rho_i, \\ &= \frac{\alpha}{2}\left(a_k \times r\right)\sum_{i=1}^{k}(-1)^i \rho'_i\rho_i. \end{aligned} \tag{1}$$

From the two vector equations $r'_i = s + p'_i r'$ and $r'_i = s + \rho_i r + \lambda a_i, \lambda \in \mathbb{R}$, we can determine ρ'_i as

$$\rho'_i = \frac{\rho_i(a_i \times r)}{(a_i \times r) + \alpha(a_i \times a_k)}. \tag{2}$$

If we also choose the edge-direction vectors a_i such that $(a_i \times r) = 1$, then eqs. (1) and (2) simplify to the following rational functions in α:

$$\rho'_i = \frac{\rho_i}{1 + \alpha(a_i \times a_k)}, \tag{3}$$

$$A = \frac{\alpha}{2}\sum_{i=1}^{k}(-1)^i \frac{\rho_i^2}{1 + \alpha(a_i \times a_k)}. \tag{4}$$

Let us look at the denominator $d_i(\alpha) = 1 + \alpha(a_i \times a_k)$ in more detail. This is important because we shall see that in

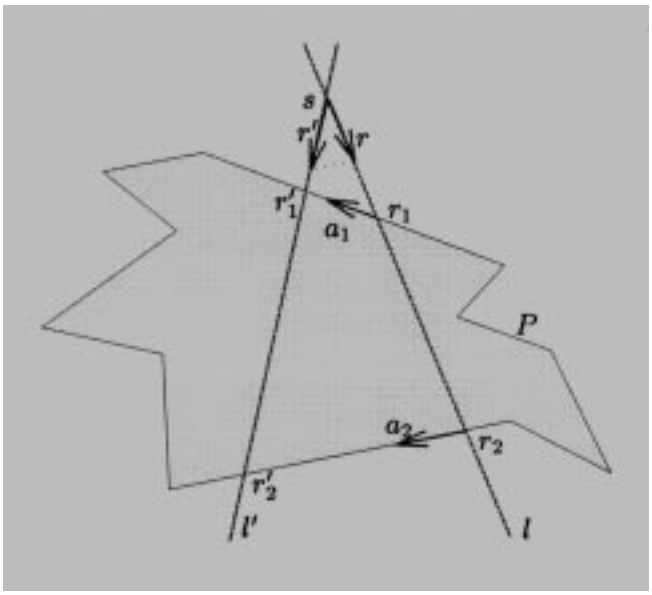

Fig. 10. Two nonparallel lines l and l' in a combinatorially equivalent intersection with polygon P.

all formulas we will obtain, the denominators consist only of $d_i(\alpha)$. For an arbitrary polygon, d_i is a linear function of α. If all a_i are parallel, then $d_i = 1$. If the polygon is rectilinear, i.e., all a_i are either parallel or perpendicular, then $d_i(\alpha) = 1$ if $a_i || a_k$, and $d_i(\alpha) = 1 + \alpha a_\perp$ if $a_i \perp a_k$, where $a_\perp$ is constant. So in this case, there are only two different constant denominators, one of which is 1.

Translating the Bisector. We now consider the case where l' shifts parallel (Fig. 11). Analogously to the previous paragraph, let $r_i' = s' + \rho_i' r'$, and $r_i'' = s'' + \rho_i'' r'$. Also, let the vector between s' and s'' be $s'' - s' = \beta a_2$. Then the polygon area between l' and l'' is

$$\begin{aligned} B &= \beta a_2 \times \frac{1}{2}\left(\left(r_2' + r_2''\right) - \left(r_1' + r_1''\right)\right), \\ &= \frac{\beta}{2}\left(\rho_2' + \rho_2'' - \rho_1' - \rho_1''\right)\left(a_2 \times (r + \alpha a_2)\right), \\ &= \frac{\beta}{2}\left(\rho_2' + \rho_2'' - \rho_1' - \rho_1''\right). \end{aligned} \tag{5}$$

In the general case, l' and l'' intersect multiple edges of some arbitrary polygon P at points $r_1', r_2', \ldots, r_k'$ and $r_1'', r_2'', \ldots, r_k''$. Now the ρ_i'' can be determined from the two vector equations $r_i'' = r_i' + \lambda a_i$, $\lambda \in \mathbb{R}$, and $r_i'' = s'' + \rho_i'' r'$:

$$\begin{aligned} \rho_i'' &= \rho_i' - \beta \frac{a_i \times a_k}{a_i \times r'}, \\ &= \rho_i' - \beta \frac{a_i \times a_k}{1 + \alpha(a_i \times a_k)}, \\ &= \frac{\rho_i - \beta(a_i \times a_k)}{1 + \alpha(a_i \times a_k}. \end{aligned} \tag{6}$$

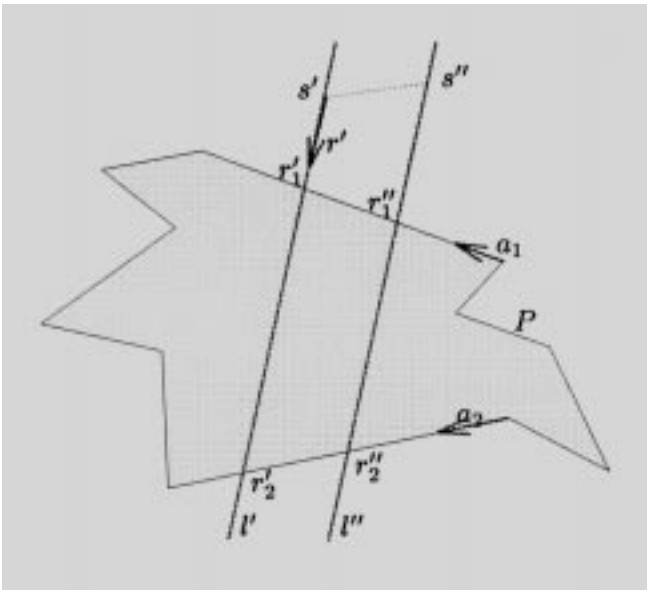

Fig. 11. Two parallel lines l' and l'' in combinatorially equivalent intersection with polygon P.

Then the polygon area between l' and l'' is

$$\begin{aligned} B &= \frac{\beta}{2}\sum_{i=1}^{k}(-1)^i(\rho_i' + \rho_i''), \\ &= \frac{\beta}{2}\sum_{i=1}^{k}(-1)^i \frac{(\rho_i - \beta(a_i \times a_k)}{1 + \alpha(a_i \times a_k)}. \end{aligned} \tag{7}$$

This is a quadratic polynomial in β (unless all a_i are parallel, in which case it simplifies to the linear equation $B = \beta \sum_{i=1}^{k}(-1)^i \rho_i$).

Maintaining the Bisector Property. From the above two paragraphs, we see that if the bisector l is rotated to l', then the left and right areas are changed by a value A ($\neq 0$ in general) as described in eq. (4). Hence, a subsequent shift of l' is necessary to restore the bisector property, by changing the areas by a value B, as described in eq. (7).

This implies the condition $A + B = 0$, with A and B given by eqs. (4) and (7):

$$\begin{aligned} A + B &= \frac{1}{2}\sum_{i=1}^{k}(-1)^i \frac{\alpha\rho_i^2 + 2\beta\rho_i - \beta^2(a_i \times a_k)}{1 + \alpha(a_i \times a_k)}. \\ &= 0. \end{aligned} \tag{8}$$

This equation ensures that l is a bisector of P. It is a necessary and sufficient condition for translation equilibrium in a unit-squeeze field. Equation (8) is a rational equation in α, and a quadratic polynomial equation in β. Hence for all combinatorially equivalent bisectors, we can obtain an explicit formula to describe β as a function of α.

In general, eq. (8) is equivalent to a polynomial in α and β whose degree depends on the number k of polygon edges intersected by the bisectors l, l', or l''. The degree of this polynomial is limited by k for α, and by 2 for β. In the rectilinear case, the degrees for α and β are limited by 2. In the case where all a_i are parallel, eq. (8) simplifies to a linear equation: $\sum_{i=1}^{k}(-1)^i(\alpha\frac{\rho_i}{2} + \beta)\rho_i = 0$.

Moment Equilibrium. After rotating (parameter α, obtain l') and translating (parameter β, obtain l'') the bisector l, its intersections with the polygon edges move from r_i to

$$\begin{aligned} r_i'' &= s + \rho_i'' r' + \beta a_k, \\ &= s + \frac{\rho_i - \beta(a_i \times a_k)}{1 + \alpha(a_i \times a_k)}(r + \alpha a_k) + \beta a_k. \end{aligned} \tag{9}$$

If all a_i are parallel, this simplifies to $r_i'' = s + \rho_i r + (\alpha\rho_i + \beta)a_k$.

Suppose that c_l and c_r, are the left and the right centers of area of P, and A_l and A_r are the respective area sections, so $A_l + A_r = A$. We are interested in how these points change when the bisector changes. Note that always $c = \frac{1}{A}(A_l c_l + A_r c_r)$, and if P is bisected (i.e., $A_l = A_r = \frac{1}{2}A$) then $c = \frac{1}{2}(c_l + c_r)$.

We consider the area between l and l'', which can be written as a sum of quadrangles (r_i, r_k, r_k'', r_i''). The weighted center area of this area can be determined as

$$C = \sum_{i=1}^{k} (-1)^i \frac{1}{6}((r_i + r_k)(r_i \times r_k) + (r_k + r_k'')(r_k \times r_k'') + (r_k'' + r_i'')(r_k'' \times r_i'') + (r_i'' + r_i)(r_i'' \times r_i)). \tag{10}$$

For the left areas, the following relationship holds (assuming $A_l'' \neq 0$):

$$A_l'' c_l'' = A_l c_l + C$$
$$\Rightarrow \quad c_l'' = \frac{A_l}{A_l''} c_l + \frac{1}{A_l''} C,$$

and similarly, for the right areas (assuming $A_r'' \neq 0$):

$$c_r'' = \frac{A_r}{A_r''} c_r - \frac{1}{A_l''} C.$$

Hence,

$$c_r'' - c_r'' = \frac{A_l}{A_l''} c_t - \frac{A_r}{A_r''} c_r + \left(\frac{1}{A_l''} + \frac{1}{A_r''} \right) C.$$

Both l and l'' are bisectors, so $A_l = A_r = A_l'' = A_r'' = \frac{A}{2}$, and

$$c_l'' - c_r'' = c_l - c_r + \frac{4}{A} C.$$

For orientation equilibrium, we require that the line connecting the centers of area, $c_r'' - c_l''$, and the direction of the bisector r', are perpendicular:

$$(c_l'' - c_r'') \cdot r' = (c_l - c_r + \frac{4}{A} C) \cdot r', \quad = 0. \tag{11}$$

The value of $C = C(\alpha, \beta)$ can be determined by using eqs. (9) and (10), and the equation $r' = r + \alpha a_k$. Equation (11) is a necessary and sufficient condition for orientation equilibrium.

By using the expressions derived in eqs. (1)–(10), both eqs. (8) (for translation equilibrium) and (11) (for orientation equilibrium) can be expressed with rational functions in α and β whose numerator (respectively, denominator) degrees are $O(k)$ (respectively, $O(1)$) for α and 2 for β. Hence, we can obtain a system of two polynomial equations of degree $O(k)$ for α and 2 for β. This system has at most $O(k)$ solutions, resulting in $O(k)$ total equilibria for bisector placements that are combinatorially equivalent. Since there are (n^2) combinatorially different bisector placements, there are at most $O(kn^2)$ total equilibria. []

3.3. Planning of Manipulation Strategies

In this section, we present an algorithm for sensorless parts alignment with squeeze fields (Böhringer et al., 1994a; Böhringer, Donald, and MacDonald 1996a). Recall from Section 3.2 that in squeeze fields, the equilibria for connected polygons are discrete (modulo a neutrally stable translation parallel to the squeeze line, which we will disregard for the remainder of Section 3).

To model our actuator arrays and vibratory devices, we made the following assumptions:

Density: the generated forces can be described by a vector field, i.e., the individual microactuators are dense compared to the size of the moving part; and

2Phase: the motion of a part has two phases: (1) pure translation towards l until the part is in translation equilibrium, and (2) motion in translation equilibrium until orientation equilibrium is reached.

Note that due to the elasticity and oscillation of the actuator surfaces, we can assume continuous area contact, and not just contact in three or a few points. If a part moves while in translation equilibrium, in general the motion is not a pure rotation, but also has a translational component. Therefore, relaxing the 2Phase assumption is one of the key results of this paper.

DEFINITION 4. Let θ be the orientation of a connected polygon P in a squeeze field, and let us assume that condition 1 holds. The turn function $t : \theta \to \{-1, 0, 1\}$ describes the instantaneous rotational motion of P:

$$t(\theta) = \begin{cases} 1 & \text{if } P \text{ will turn counterclockwise,} \\ -1 & \text{i f } P \text{ will turn clockwise,} \\ 0 & \text{if } P \text{ is in total equilibrium.} \end{cases}$$

See Figure 12 for an illustration. The turn function $t(\theta)$ can be obtained, for example, by taking the sign of the lifted moment $M_P(\mathbf{z})$ for poses $\mathbf{z} = (x, y, \theta)$, in which the lifted force $f_P(\mathbf{z})$ is zero.

Definition 6 immediately implies the following lemma.

LEMMA 2 (Böhringer, MacDonald, and Donald 1996a). Let P be a polygon with orientation θ in a squeeze field such that condition 1 holds. P is stable if $t(\theta) = 0, t(\theta+) \leq 0$, and $t(\theta-) \geq 0$; otherwise, P is unstable.

Proof. Assume the part P is in a pose (x, y, θ) such that condition 1 is satisfied. This implies that the translational forces acting on P balance out. If in addition $t(\theta) = 0$, then the effective moment is zero, and P is in total equilibrium. Now consider a small perturbation $\delta_\theta > 0$ of the orientation θ of P while condition 1 is still satisfied. For a stable equilibrium, the moment resulting from the perturbation δ_θ must not aggravate, but rather counteract, the perturbation. This is true if and only if $t(\theta + \delta_\theta) \leq 0$ and $t(\theta - \delta_\theta) > 0$.

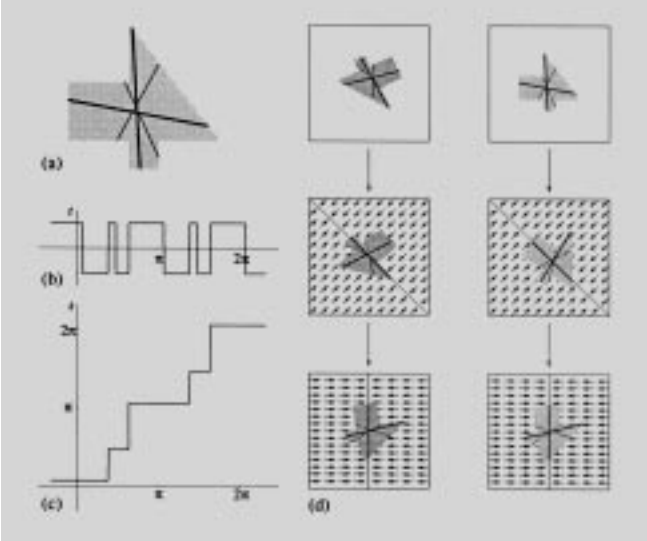

Fig. 12. (a) Polygonal part: stable (thick line) and unstable (thin line) bisectors are also shown. (b) Turn function, which predicts the orientations of the bisectors. Stable (respectively, unstable) bisectors correspond to angles at which the turn function changes from +1 to −1 (respectively, from −1 to +1). (c) Squeeze function, constructed from the turn function. (d) Alignment strategy for two arbitrary initial configurations. (See the World Wide Web at www.ee.washington.edu/faculty/karl/Research/ for an animated simulation.)

Using this lemma, we can identify all stable orientations, which allows us to construct the squeeze function (Goldberg 1993) of P (see Fig. 12c); i.e., the mapping from an initial orientation of P to the stable equilibrium that it will reach in the squeeze field:

LEMMA 3. Let P be a polygonal part on an actuator array $\mathcal{A}$ such that Density and 2Phase hold. Given the turn function t of P, its corresponding squeeze function $s : \mathbb{S}^1 \to \mathbb{S}^1$ is constructed as follows:

1. all stable equilibrium orientations θ map identically to θ;
2. all unstable equilibrium orientations map (by convention[3]) to the nearest counterclockwise stable orientation; and
3. all orientations θ with $t(\theta) = 1(-1)$ map to the nearest counterclockwise (clockwise) stable orientation.

Then, s describes the orientation transition of P induced by $\mathcal{A}$.

Proof. Assume that part P initially is in pose (x, y, θ) in array $\mathcal{A}$. Because of the 2Phase assumption, we can assume that P translates toward the center line l until condition 1 is satisfied without changing its orientation θ. P will change its orientation until the moment is zero, i.e., $t = 0$: a positive moment, $(t > 0)$ causes counterclockwise motion, and a negative moment $(t < 0)$ causes clockwise motion until the next root of t is reached. □

We conclude that any connected polygonal part, when put in a squeeze field, reaches one of a *finite* number of possible orientation equilibria (Böhringer et al. 1994a; Böhringer, Donald, and MacDonald 1996a). The motion of the part and, in particular, the mapping between initial orientation and equilibrium orientation is described by the squeeze function, which is derived from the turn function (as described in Lemma 3). Note that all squeeze functions derived from turn functions are monotone step-shaped functions.

Goldberg (1993) has given an algorithm that automatically synthesizes a manipulation strategy to uniquely orient a part, given its squeeze function. While Goldberg's algorithm was designed for squeezes with a robotic parallel-jaw gripper, in fact, it is more general, and can be used for arbitrary monotone step-shaped squeeze functions. The output of Goldberg's algorithm is a sequence of angles that specify the required directions of the squeezes; therefore, these angles specify the direction of the squeeze line in our force-vector fields (for example, the two-step strategies in Figures 7 and 12d). It is important to note that the equilibria obtained by a MEMS squeeze field and by a parallel-jaw gripper will typically be different, even when the squeeze directions are identical. For example, to see this, consider squeezing a square-shaped part (Fig. 13). Stable and unstable equilibria are reversed. This shows that our mechanical analysis of equilibrium is different from that of the parallel-jaw gripper. Let us summarize these results in the following statements.

PROPOSITION 2. Let P be a polygon whose interior is connected. There exists an alignment strategy consisting of a

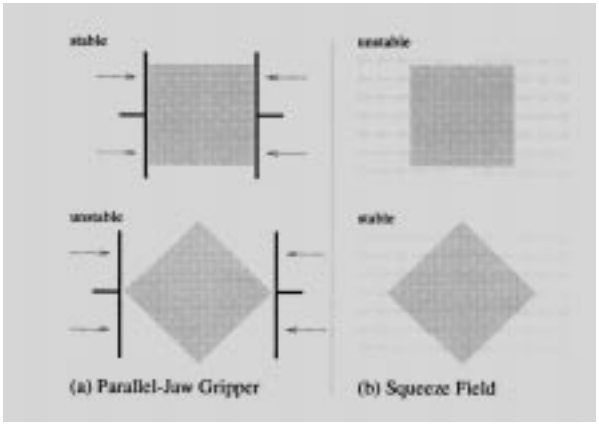

Fig. 13. Equilibrium configurations for a square-shaped part using (a) a frictionless parallel-jaw gripper, and (b) a MEMS squeeze field. In this example, stable and unstable equilibria are reversed.

3. Equally, one could define t to map unstable equilibrium orientations to the nearest clockwise stable equilibrium. This choice for a set of measure zero does not affect our subsequent analysis and algorithms.

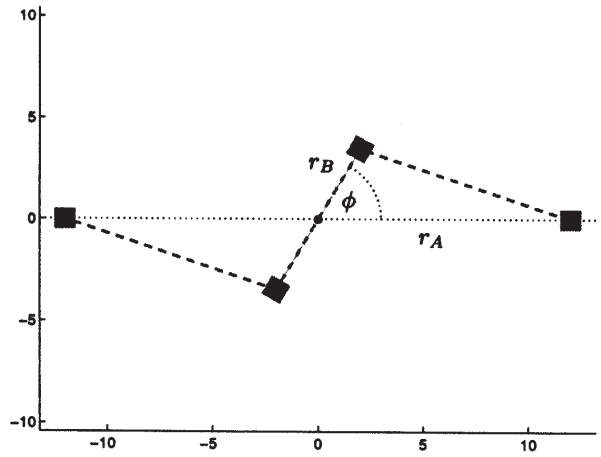

Fig. 18. An S-shaped part with four rigidly connected square "feet" in configuration $(x, y, \theta) = (0, 0, 0)$, $r_A = 12$, $r_B = 4$, and $\phi = 60°$.

60°, and the feet have area size 10. Notice that in poses with θ-angles corresponding to minima in the potential, the moment has a root with negative slope, which indicates a stable (orientation) equilibrium. Figures 20b and 20c show the (normalized) moments and potentials for parts with feet sizes 5 and 1, respectively. We observe that with decreasing contact areas, these functions become "less smooth," and the slope at the moment root increases. Figure 20d depicts moment and potential for a part with infinitesimally small feet. In this case, the moment function does not have a root at the minimum of the potential; rather, it exhibits a discontinuity at this orientation. This has the consequence that the part is not stable in this pose. In fact, for the moment function in Figure 20d, there exist no roots with negative slope, and hence there exists no stable equilibrium.

This observation can be made mathematically precise. The exact equations for the lifted potential and the moment of P_S are

$$U_{P_S} = 2r_A|\cos\theta| + 2r_B|\cos(\theta+\phi)|, \tag{17}$$
$$M_{P_S} = 2r_A S(\theta) + 2r_B S(\theta+\phi),$$

$$\text{with } S(\theta) = \begin{cases} \sin\theta & \text{if } 0 \le \theta < \pi/2 \text{ or } 3/2\pi < \theta < 2\pi, \\ -\sin\theta & \text{if } \pi/2 < \theta < 3/2\pi, \\ 0 & \text{if } \theta = \pi/2 \text{ or } \theta = 3/2\pi. \end{cases} \tag{18}$$

The potential minimum is reached at $\theta = \pi/2$ or $\theta = 3/2\pi$. However, we see that, for example, $M_{P_S}(\pi/2) = -2r_B S(\pi/2 + \phi) = -2r_B \cos\phi \neq 0$. Furthermore, $M_{P_S}(\pi/2-) > 0$, and $M(\pi/2+) < 0$. This implies that the part P_S will oscillate about $\theta = \pi/2$. Under first-order dynamics, this oscillation will be infinitesimally small, because any infinitesimal angular deflection of P_S results in a restoring moment with opposed orientation. Under second-order dynamics, the part may have a finite oscillation amplitude because of the inertia of the part. However, damping will reduce this amplitude over time.

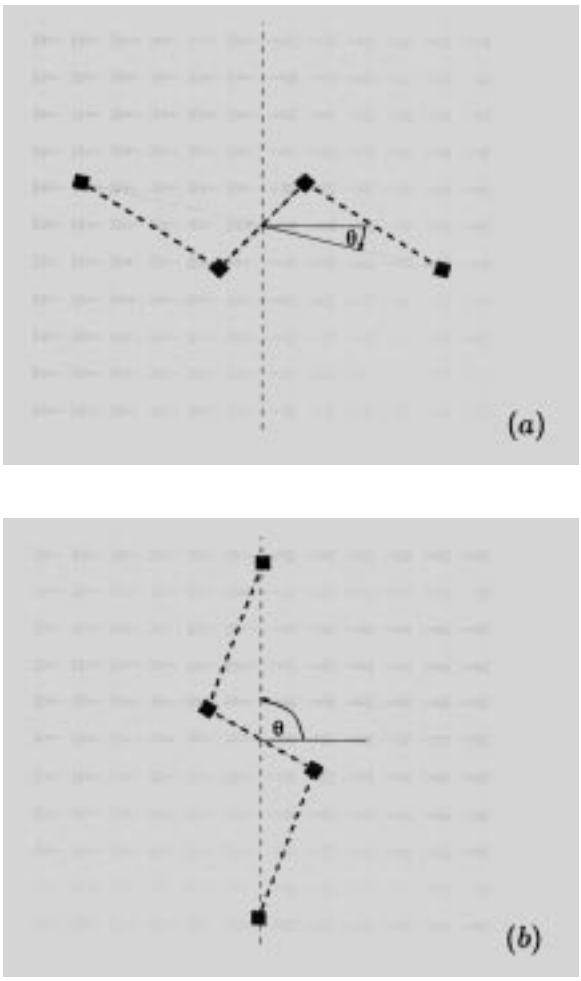

Fig. 19. Total equilibria of an S-shaped part with area contacts in a squeeze field. (a) Maximum potential, $\mathbf{z}_{\max} = (0, 0, \theta_{\max})$, such that $r_A \sin\theta_{\max} = -r_B \sin(\theta_{\max} + \phi)$; $\theta_{\max} \approx -0.24$. (b) Minimum potential, $\mathbf{z}_{\min} = (0, 0, \theta_{\min})$; $\theta_{\min} \approx \pi/2$.

We conclude that parts with point contacts can exhibit pathological behavior even in very simple and otherwise well-behaved potential fields: this example shows that for such parts, it is possible that the generalized force is not zero in a pose that minimizes the potential of the part.

This pathology cannot occur when only parts with finite area contact are allowed. From Corollary 3, we know that the (lifted) potential of a part with area contact is C^1; hence its gradient exists everywhere. In particular, the gradient is zero at the minimum of the potential. This means that in a pose with minimum potential, the generalized force must be zero. Let us summarize these results.

Table 1.

		Complexity		
Task	Field(s)	Fields	Planning	Plan Steps
Translate	Constant	Constant magnitude and direction	–	1
Center	Radial	Constant magnitude, continuous directions	–	1
	Orthogonal squeezes magnitude and direction	Piecewise constant	$O(1)$	$O(1)$
Uniquely orient	Sequence of squeezes	Piecewise constant magnitude and direction	$O(k^2n^4)$	$O(kn^2)$
	Inertial	Smooth magnitude piecewise-constant direction	$O(1)$	$O(1)$
Uniquely pose	Manipulation grammar	m arbitrary fields, at most E stable equations	$O(m^2 2^E)$	$O(m2^E)$ (not complete)
	Sequence of radial + squeeze	Piecewise-continuous magnitude and direction	$O(k^2n^2)$	$O(kn)$
	Elliptic	Smooth magnitude and direction	$O(1)$	$O(1)$
	UFO	Continuous magnitude and direction	$--$	1

- Magnitude control. Consider an array in which the *magnitude* of the actuator forces cannot be controlled. Does there exist an array with constant magnitude in which all parts reach one unique equilibrium? Or can one prove that, without magnitude control, the number of distinct equilibria is always greater than one?

- Geometric filters. This paper focuses mainly on sensorless manipulation strategies for *unique positioning* of parts. Another important application of programmable vector fields are *geometric filters*, which would be useful for the sorting and singulation of parts. Figure 1 shows a simple filter that separates smaller and larger parts. We are interested in the question, Given n parts, does there exist a vector field that will separate them into specific equivalence classes? For example, does there exist a field that moves small and large rectangles to the left, and triangles to the right? In particular, it would be interesting to know whether for any two different parts there exists a sequence of force fields that will separate them.

- Force-field computers. In this paper, we have demonstrated that even with a rather limited vocabulary of simple force fields, useful and quite complex tasks such as sensorless posing or sorting of parts can be performed. It might be possible that force fields could be used to solve certain classes of problems, by encoding them in particular force fields, part shapes, and initial and goal poses, resulting in a "force-field computer" that provides a physical implementation of the problem. Identifying the class of encodable problems might yield deeper insights into the complexity of parts manipulation with force-vector fields.

- Performance measures. Are there performance measures for how fast (in real time) an array will orient a part? In some sense, the actuators are fighting each other (as we have observed experimentally) when the part approaches equilibrium. For squeeze grasps, one measure of "efficiency," albeit crude, might be the integral of the magnitude of the moment function, i.e., $\int_0^{2\pi} |M(\theta)|d\theta$. The issue is that if, for many poses, $|M(\theta)|$ is very small, then the orientation process will be slow. Better measures are also desirable.

- Uncertainty. In practice, neither the force-vector field nor the part geometry will be exact, and both can only be characterized up to tolerances (Donald 1989). This is particularly important at the microscopic scale. Within the framework of potential fields, we can express this uncertainty by considering not one single potential function U_P, but rather *families of potentials* that correspond to different values within the uncertainty range. Bounds on part and force tolerances will correspond to limits on the variation within these function families. An investigation of these limits will allow us to obtain upper error bounds for manipulation tasks under which a specific strategy will still achieve its goal.

  A family of potential functions is a set $\{U_\alpha : \mathcal{C} \rightarrow \mathbb{R}\}_{\alpha \in J}$ where J is an index set. For example, we may start with a single potential function $U : \mathcal{C} \rightarrow \mathbb{R}$ and define a family of potential functions $\mathcal{F}(U, \epsilon, z)$ as $\{\{U_\alpha : \mathcal{C} \rightarrow \mathbb{R} \mid ||U_\alpha(p) - U(p)||_z < \epsilon\}$ for some ϵ and