

Universal Part Manipulation in the Plane with a Single Horizontally-Vibrating Plate

Dan S. Reznik, *UC-Berkeley, Berkeley, CA, USA*
John F. Canny, *UC-Berkeley, Berkeley, CA, USA*

Active surfaces can manipulate (i.e., translate, rotate) one or more parts in the plane by “displaying” arbitrary forces at a number of points. Many designs have been proposed based on a massively-parallel array of microactuators. For such “pixel-oriented” devices, a large density of microactuators per unit area is required for acceptable force field programmability; the large number and small size of moving parts renders these devices difficult to fabricate, test, and maintain. Here we take a minimalist approach and look for a mechanically simpler, sturdier design, which retains the same manipulation dexterity. Surprisingly, we show that a rigid flat plate is just such a device. Namely, a closed, horizontal motion of the plate can be computed which yields arbitrary frictional forces (averaged over the motion) at one or more points, so that one or more parts can be translated and/or rotated simultaneously and independently. Analysis for both part feeding and part manipulation cases, along with dynamic simulation and experimental results with a prototype of the device are included.

1 Introduction

Dextrous manipulation of parts in the plane is of key importance in the assembly, sorting, and feeding of industrial parts. Traditional pick-and-place methods are not suitable for small scale, many-part, and/or parallel manipulation applications due to end-effector clutter, planning/control complexity, and execution latencies. Research on active surfaces has addressed this issue by considering devices which incorporate the actuation mechanism into the surface on which parts rest.

A popular design concept is that of a massively-parallel array of micro-actuators. An arbitrary force can be generated at a specific location by addressing the appropriate “pixel”, or group of actuators.

Alternatively, their massive parallelism can be used to synthesize “sensorless” fields which automatically locate/orient parts [1, 2]. Existing devices include arrays of thermally-actuated cilia [3], micro-electro-mechanical resonators [1], orientable rollers [4], air-flow nozzles [5], and electric/magnetic elements [6, 7]. The large number and small size of moving elements in array-like devices pose fabrication difficulties and reduce device robustness.

Driven by the concept of minimalism in robotics [8], we search for a mechanically simpler and sturdier device which retains full manipulation dexterity. Our main contribution is to show that a single horizontally-vibrating rigid, flat plate is such a device. Our proposed device concept is shown in Figure 1. A rigid, flat plate is suspended parallel to ground by the shafts of four linear-travel (e.g., voice coil) motors. These can command arbitrary instantaneous accelerations (within bounds), along the plate’s three planar degrees of freedom. Parts placed on the plate’s surface are manipulated via sliding frictional forces.

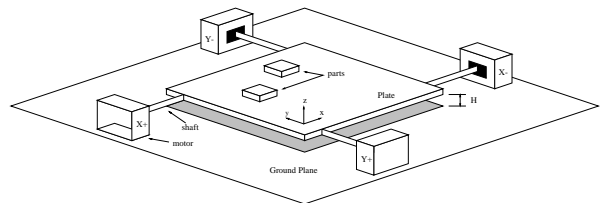


Figure 1: *Model of the proposed device – a rigid, flat horizontal plate and an actuation mechanism (made up of four linear-travel motors). The plate is constrained to vibrate horizontally. Two parts are shown resting on the plate.*

Our main result is to show that given desired forces at one or more part locations, a closed, horizontal motion of the plate – a vibration cycle – can be computed which yields the desired forces (averaged over the en-

tire motion) independently and simultaneously at each point. At the heart of this result is a sequence of five observations: (i) the magnitude of dynamic friction is constant and independent on the relative velocity; (ii) any instantaneous planar rigid motion of the plate corresponds to a rotation about a point; (iii) such a motion gives rise to a frictional force field (assuming dynamic friction at every point) of constant magnitude vectors, oriented tangentially with respect to the center of rotation; (iv) the family of fields of type (iii) is not closed under addition (unlike the family of rigid velocity fields), so that (v) the sum of N frictional force fields is N -dimensional (compare this to a maximum dimension of 3 for rigid velocity fields).

We propose two methods to computing the aforementioned closed motion. In the first method, the plate’s motion is composed of a finite sequence of small rotations about a known set of points (e.g., chosen conveniently given the actuation kinematics). Assume the entire sequence is executed fast enough and/or friction at the interface is rather small so that parts lying on the plate’s surface move little throughout the sequence. The average force applied to a part during the entire motion is the sum of frictional forces corresponding to each rotation. Owing to (v), the problem of generating desired average frictional forces under each part becomes that of solving for the duration of each rotation (i.e., its “intensity”, or “weight”), which under the present method reduces to inverting a linear system.

Switching abruptly between rotations can lead to resonance in the plate’s suspension. In the second method, vibration bandwidth is reduced by making the plate describe a smooth, closed motion involving its 3 degrees of freedom simultaneously. We fix the motion along each degree of freedom to be a sinusoid plus a double frequency sinusoid, each with controllable phase and amplitude, for a total of 12 free parameters. The map from parameter space to an average force at a given point is highly convoluted and not likely to be invertible; Given (v), above, we can expect the image of this map to be high-dimensional (at most 12-dimensional). By using iterative, numeric inversion methods we can compute the parameters which yield the desired forces.

1.1 Related work

The quintessential instance of a vibrations-based part manipulation is the vibratory bowl feeder [9]. In this case, off-plane vibration is used to feed parts along a track internal to the feeder. The APOS feeder [9] uses vibration to aid in massively parallel passive part entrapment applications. Designing feeder/part shapes for optimum entrapment has been the subject of some recent research [10]. More exotic experiments with vibration in the context of part manipulation include resonance-based manipulation [11] and part localization through node-formation in an oscillating surface [12].

Much of the work described here is a continuation of our research in exploiting horizontal vibrations for the purpose of part feeding and manipulation. In [13, 14] we describe a method for part feeding based on a 1d horizontally-vibrating plate. In [15] these ideas are generalized to the 2d manipulation case; the method of sequencing rotations is first introduced there; here this method is further described and contrasted with a second, lower-bandwidth method.

The remainder of this paper is organized as follows: In Section 2 we present manipulation preliminaries including plate kinematics, the concept of force field averaging, and the non-additivity of frictional force fields. In Section 3 we describe the parallel manipulation task along with two proposed methods (sequencing rotations and sinusoidal controls) for computing the required closed plate motions. In Section 4 we present dynamic simulation results for two parallel manipulation tasks. In Section 5 we describe some experiments we have performed with a prototype of the device. Conclusions are presented in Section 6.

2 Manipulation Preliminaries

2.1 Coulomb Friction in 2d

Let ν_s and ν_p denote, respectively, the instantaneous surface (i.e., the plate) and part velocities, relative to an inertial frame XY , Figure 2. Assuming the surfaces are sliding (i.e., the relative acceleration is sufficiently high), the frictional force f_{fric} will be of constant magnitude μmg and point opposite to the part’s motion relative to the surface, i.e.:

$$f_{fric,2d} = \mu mg \frac{(\nu_s - \nu_p)}{\|\nu_s - \nu_p\|} \quad (1)$$

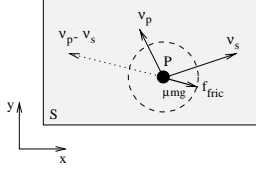


Figure 2: Coulomb friction in 2d: the frictional force lies on a circle of radius μmg centered on the part (P), and points opposite to P 's velocity relative S , i.e., $\nu_p - \nu_s$.

2.2 Force Field Averaging

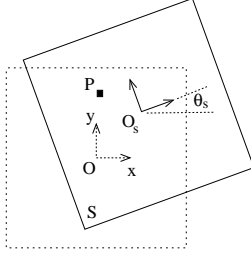


Figure 3: Dotted square: the plate S at its home position, where its center coincides with the origin O . Solid square: S with its center translated by O_s and rotated by θ_s . A part is shown lying on S at point P .

The surface S is constrained to move in its own plane. Let $O_s = (x_s, y_s)$, and θ_s denote S 's translation and orientation relative to XY , Figure 3. Consider a smooth motion $S(t) = [O_s(t), \theta_s(t)]^t$ of the surface which is *closed*, i.e., $S(0) = S(T)$, for some $T > 0$. The surface's tangential velocity ν_s at a point P is:

$$\nu_s(P, t) = \dot{O}_s(t) + \dot{\theta}_s(t) \times [P - O_s(t)] \quad (2)$$

Let us consider points P much larger in magnitude than the surface's maximum translation, i.e., $\|P\| \gg \max_t \|O_s(t)\|$, so that $P - O_s(t) \simeq P$. Equation 2 reduces to:

$$\begin{aligned} \nu_s(P, t) &= \dot{O}_s(t) + \dot{\theta}_s(t) \times P \\ &= [u_1(t) - y u_3(t), u_2(t) + x u_3(t)]^t \quad (3) \end{aligned}$$

Where symbols $u_1(t)$, $u_2(t)$, and $u_3(t)$ represent $\dot{x}_s(t)$, $\dot{y}_s(t)$, and $\dot{\theta}_s(t)$, respectively. Note $S(t)$ closed implies $\int_0^T u_i(t) dt = 0$, $i = 1 \dots 3$.

Assumption 1 P is always sliding on S – surface accelerations at P give rise to inertial forces above the threshold of static friction [16], for most $t \in [0, T)$.

Note: $S(t)$ closed implies that $\dot{\nu}_s(t)$ will have one or more zero-crossings within $[0, T)$ during which the part will tend to stick to the surface; this is why Assumption 1 can only be true for “most” t .

If the part's velocity $\nu_p(t)$ is negligible compared to $\nu_s(P, t)$ for most $t \in [0, T)$, Equation 1 reduces to:

$$f_{fric}(P, t) = \mu mg \frac{\nu_s(P, t)}{\|\nu_s(P, t)\|} \quad (4)$$

An upper bound for the part's maximum displacement in T seconds given frictional forces is $\|\Delta P\|_{max} = \mu g T^2$. Let us assume this quantity negligible with respect to $\|P\|$ so that the part's position can be considered nearly constant within one cycle. The average frictional force \bar{f}_{2d} applied to the part becomes simply the integral of the instantaneous frictional forces at a P divided by the length of the motion:

$$\bar{f}_{2d}(P) = \frac{1}{T} \int_0^T f_{fric}(P, t) dt \quad (5)$$

2.3 Instantaneous force fields

Chasles' theorem [17] in planar kinematics states that any instantaneous rigid velocity $[u_1, u_2, u_3](t)$ corresponds to a rigid rotation about a center $C = (c_x, c_y)$ at angular velocity w where:

$$\begin{aligned} c_x &= -u_2/u_3 \\ c_y &= +u_1/u_3 \\ w &= u_3 \end{aligned}$$

If the $u_i(t)$'s are directly controllable, we can chose them so S executes a rigid rotation about a fixed center C . Alternatively, the actuation kinematics maybe such that rotating the surface about one (or more) fixed points is a simple procedure. Consider an instantaneous rotation $\theta(t)$ of S about a point C , $\theta(0) = \theta(T)$.

With C stationary, S 's instantaneous tangential velocity $\nu_s(t)$ at point $P = (x, y)^t$ is $\dot{\theta}(t) \times (P - C)$, or:

$$\nu_s(t) = \dot{\theta}(t) \begin{bmatrix} c_y - y \\ x - c_x \end{bmatrix} \quad (6)$$

Define the unit tangential vector $\hat{\tau} = (P - C)/\|P - C\|$. From Equation 6, the direction of f_{fric} along $\hat{\tau}$ is given by $sgn(\dot{\theta})$, yielding the force field illustrated on Figure 4. So f_{fric} can be expressed succinctly as:

$$f_{fric}(t) = \mu mg \, sgn[\dot{\theta}(t)] \hat{\tau} \quad (7)$$

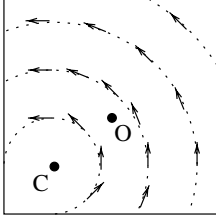


Figure 4: A rigid motion of S about C yields a frictional force field of tangential vectors of constant magnitude.

2.4 Non-additivity of force fields

Let $\psi_{k,C}(P)$ and $\phi_{k,C}(P)$ represent, respectively, the value of arbitrary velocity and average force fields at a point P , where k and C denote the fields' "strength" (a linear scaling parameter, explained below) and center of rotation, respectively. Algebraically:

$$\begin{aligned} \psi_{k,C}(P) &= k \, \mu mg \, R_{90}(P - C) \\ \phi_{k,C}(P) &= k \, \mu mg \, R_{90} \frac{P - C}{\|P - C\|} \\ R_{90} &= \text{rotate-by-90}^\circ\text{-matrix} \end{aligned} \quad (8)$$

Let Ψ (resp. Φ) denote the 3-dimensional family of ψ 's (resp. ϕ 's):

$$\begin{aligned} \Psi &= \{\psi_{k,C} | k \in \mathcal{R}, C \in \mathcal{R}^2\} \\ \Phi &= \{\phi_{k,C} | k \in \mathcal{R}, C \in \mathcal{R}^2\} \end{aligned}$$

It can be shown that for $\psi_1 = \psi_{k_1, C_1}$ and $\psi_2 = \psi_{k_2, C_2}$, $\psi_1 + \psi_2 = \psi_{k_3, C_3}$ with:

$$\begin{aligned} k_3 &= k_1 + k_2 \\ C_3 &= \frac{k_1 C_1 + k_2 C_2}{k_1 + k_2} \end{aligned}$$

i.e., Ψ is closed under addition. This property is however not true for Φ , namely:

Observation 1 Φ is not closed under addition: Let $\phi_1, \phi_2 \in \Phi$. Then $\phi_1 + \phi_2 \notin \Phi$. Namely, $\nexists k_3, C_3$ such that $\phi_1 + \phi_2 = \phi_{k_3, C_3}$.

In other words, the sum of two force fields does not lie in the original space, i.e., the resulting field will be part of a higher-dimensional family of vector fields, as depicted in Figure 5.

3 The parallel manipulation task

Consider M parts distributed over S at known locations P_i , $i = 1..M$. Let $f_i(k)$ be a desired average force for part i at a given instant. The goal is to compute a closed motion $S(t)$ which yields $\bar{f}_{2d}(P_i) = f_i$, $\forall i$, as shown in Figure 6.

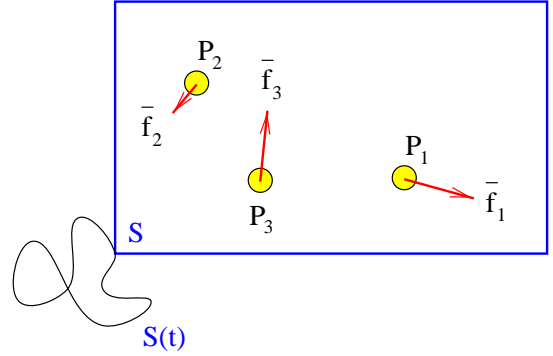


Figure 6: The parallel manipulation task consists in computing a closed motion $S(t)$ for the surface S which can generate desired frictional forces \bar{f}_i (averaged over the motion) at each of the parts' locations P_i .

The setup envisioned is as shown in Figure 7. A snapshot of S is obtained with a camera and sent to the computer. Image processing algorithms are used to determine the P_i 's. From the state within a manipulation task, desired forces f_i are computed for each part; this is followed by a solving phase in which a closed motion $S(t)$ is computed for the surface; this

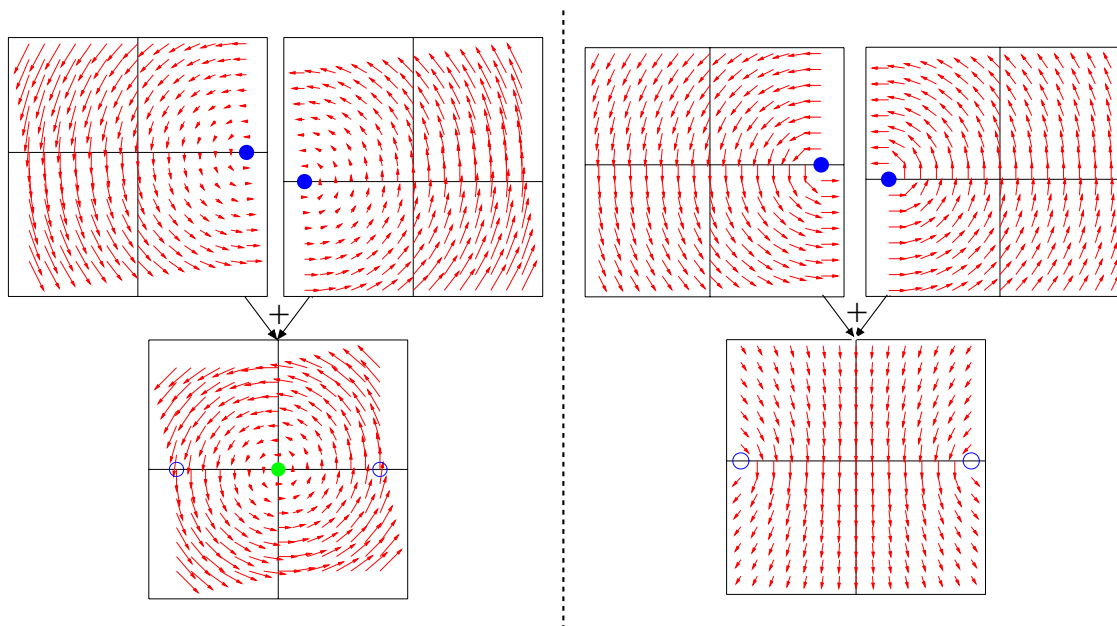


Figure 5: Left: the sum of two velocity fields is a velocity field. Right: the sum of two frictional force fields is not in the same family.

motion is then sent out as commands to motors driving the surface, and the loop closes. Feedback helps in compensating for deviations from the desired type of manipulation, caused by modeling errors, unevenness in friction, control inaccuracies, etc.

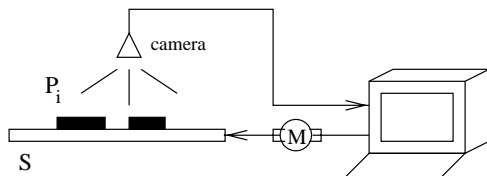


Figure 7: Closed-loop manipulation: a camera sends images to a computer, which sends commands to a motor which actuates the surface which moves the parts.

In what follows we present two methods to solving for a closed motion $S(t)$ which yields the \bar{f}_i 's.

3.1 Method A: Sequencing Rotations

Consider N consecutive small-angle rotations of the surface, $N \geq 2M$, each about a distinct point C_j , $j = 1 \dots N$. Let each such duration's rotation be scaled

by an independent k_j , such that $\sum k_j = T$. Let $\phi_j = \mu mg \phi_{k_j, C_j}$.

We assume that parts move negligibly during the entire motion, or, equivalently, that the ϕ_j 's are constant in a sufficiently large neighborhood of the P_i 's. This renders true the following observation.

Observation 2 Superposition: The net average force \bar{f}_i felt by P_i after N rotations is the weighted sum of the ϕ_j 's at the parts' starting positions P_i :

$$\bar{f}_i = \frac{1}{T} \sum_{j=1}^N \phi_j(P_i), i = 1 \dots M \quad (9)$$

Assume the C_j 's are pre-determined (e.g., by the manipulation kinematics), and that the \bar{f}_i are given by the manipulation task. Combining Equations 8 and 9 reduces the parallel part manipulation problem to finding a solution vector $K = (k_1 \dots k_N)$ which satisfies the linear system $A.K = b$, with:

$$A_{ij} = \begin{cases} -a(i, j) & , i = 1, \dots, M \\ b(i - M, j) & , i = M + 1, \dots, 2M \end{cases}$$

$$j = 1 \cdots N \quad (10)$$

$$b_i = \frac{N}{\mu mg} \begin{cases} \bar{f}_i^x & , i = 1, \dots, M \\ \bar{f}_{i-M}^y & , i = M + 1, \dots, 2M \end{cases}$$

and:

$$a(i, j) = (p_i^y - c_j^y)/r_{ij}$$

$$b(i, j) = (p_i^x - c_j^x)/r_{ij}$$

$$r_{ij} = \|P_i - C_j\|$$

3.1.1 Force specification example

Consider a rectangular surface rotatable about its 4 corners $C_1 = (-1, -0.5)^t, C_2 = (1, -.5)^t, C_3 = (1, .5)^t, C_4 = (-1, .5)^t$. Consider two parts $P_1 = (.5, 0)^t$ and $P_2 = (-.75, .25)$. Let $\mu = .25, g = 10, m = .04$ ($\mu mg = .1$) (note: all in SI units). Let $\bar{f}_1 = (.01, .01)^t$ and $\bar{f}_2 = (-.01, 0)^t$. The goal is to compute a 4-long vector $K = (k_1, k_2, k_3, k_4)$ and execute 4 rotations consecutively about each of C_i 's (Figure 8 with each such motion "scaled", in duration, by the computed k_j . The components of the 4×4 linear system become:

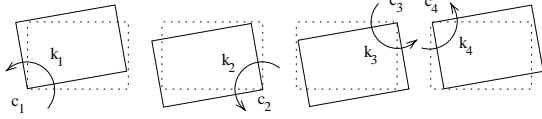


Figure 8: The surface executes 4 rotations about each of its corners, say, in CCW order. The duration of each rotation is scaled by a computed k_j .

$$A_{4 \times 4} = \begin{bmatrix} -.316 & -.707 & .707 & .316 \\ -.949 & -.394 & .141 & .707 \\ .949 & -.707 & -.707 & .949 \\ .316 & -.919 & -.990 & .707 \end{bmatrix}$$

$$b = (.4, -.4, .4, 0)^t$$

Inverting A and compute $k = A^{-1} \cdot b$ to obtain:

$$K = (.58, -.39, .48, -.095)^t$$

As shown in Figure 9, the resulting force field are as desired at the part's locations.

3.1.2 Rotating parts

The *curl* $\nabla \times f$ of a vector field $f = (f_x, f_y)^t$ is the scalar function $\frac{d}{dx}f_y - \frac{d}{dy}f_x$. It measures the rotation

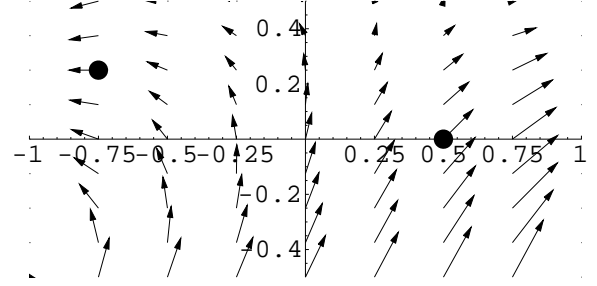


Figure 9: The average force vectors corresponding to a series of 4 rotations about S 's corners. The field points in the desired directions at the location of each part, shown as black disks.

of an infinitesimal area element flowing along the field [18]. A simple expression yields the curl of a rotation-induced force field $\phi_{k,C}$:

$$\nabla \times \phi_{k,C}(P) = k \frac{\mu mg}{\|P - C\|} \quad (11)$$

While the curl is measured at a point and the part's footprint will be over an area, one way to induce rotation is to specify a non-zero curl under the part's center of mass, while simultaneously specifying a zero-force for that point. Since the generated fields are continuous, this will create a "whirlpool" centered at the part.

The curl is a linear operator, so for a set of vector fields $\phi_i, \nabla \times \sum_i \phi_i = \sum_i \nabla \times \phi_i$. So the curl of the net average force field (Equation 9) can be written as:

$$\nabla \times f_i(P_i) = \frac{\mu mg}{N} \sum_{j=1}^N \frac{k_j}{\|P_i - C_j\|} \quad (12)$$

The above amounts to a single linear constraint, i.e., it becomes a row of the A matrix used in Equation 10. So besides specifying zero-part motion at M locations (amounting to $2M$ constraints), we can specify the curl, simultaneously, at Q locations (Q constraints). To avoid over-constraining the system, $2M + Q \leq N$, the number of rotation centers.

3.1.3 Curl specification example

Consider a single part P_1 placed at location $(.5, 0)$. We want to specify $\bar{f}_1 = (0, 0)^t$ (2 constraints) and

the curl at P_1 (1 constraint) to be $\delta_1 = -.05$ (negative curl means the field should be turning clockwise around that point). Since only 3 linear constraints are specified, there is no need for a fourth rotation about C_4 . The resulting 3×3 linear system becomes:

$$\begin{bmatrix} -s_{11} & -s_{12} & -s_{13} \\ c_{11} & c_{12} & c_{13} \\ r_{11}^{-1} & r_{12}^{-1} & r_{13}^{-1} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \frac{N}{\mu mg} \begin{bmatrix} f_1^x \\ f_1^y \\ \delta_1 \end{bmatrix}$$

Which yields:

$$\begin{bmatrix} -.316 & -.707 & .707 \\ .949 & -.707 & -.707 \\ .633 & 1.41 & 1.41 \end{bmatrix} \cdot \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1.5 \end{bmatrix}$$

Solving the above yields $K = (-.59, -.27, -.53)$. The resulting field, illustrated in Figure 10, contains a CW whirlpool at the part's position, as desired. The integral of the forces under the part generates negative torque, resulting in CW rotation.

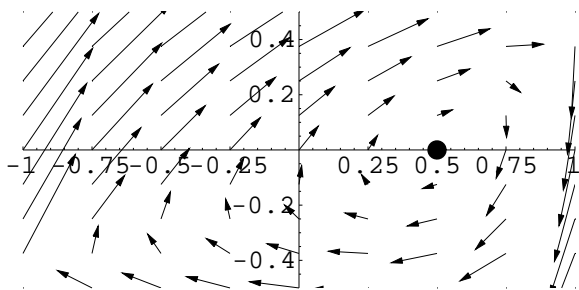


Figure 10: Average force field obtained after 3 rotations about C_1 , C_2 , and C_3 in sequence. Zero force and negative curl are desired at the part's location $(0.5, 0)$. The result is a CW whirlpool centered on the part

3.1.4 Ordering rotations

Superposition assumes the fields are slowly-varying around the parts' neighborhoods. However, field variability increases as one approaches the center of rotation (conversely, at infinity, an average force field looks like a constant field). Furthermore, field variability is proportional to the scaling k_j of a particular rotation. For every center of rotation C_j , define ρ_j , the center's rotation priority as follows:

$$\rho_j = \frac{k_j}{\min_i \|P_i - C_j\|}$$

A heuristic approach to reduce rotation-ordering effects which violate the superposition assumption is to rotate about the C_j 's in *descending* order of rotation priority, ρ_j , above. In Section 3.1.1, $(\rho_1 \cdots \rho_4)^t = (.16, .031, .43, .34)^t$, so the "optimal" rotation ordering is C_3, C_4, C_1 , and C_2 .

A more rigorous approach to understand order-dependent errors would have to take into account the component of each motion lying on the *lie-bracket* $[\phi_1, \phi_2]$ of two consecutive rotations [19].

3.1.5 Ill-conditioning

The inverse of A (Equation 10) becomes ill-conditioned when at least one the following is true:

- Distinct field values are specified for nearby locations (e.g., for parts approaching each other).
- The field value is specified too far from the centers of rotation.

The first case, specifying distinct field values in a small neighborhood forces high derivatives in the resulting (average) field. Since the latter is the sum of a set of smooth functions ϕ_i , this de-stabilizes the inversion.

For the second case, consider a point P very far from a center of rotation C . The resulting force field around P will be nearly constant and perpendicular to $P - C$ (tangential lines become straight lines). By superposition, the same effect will happen for a sequence of rotations about several distinct centers. Since the family of constant fields has rank 2 and is closed under addition, A 's rank will be limited to 2 in that region.

Ill-conditioning yields solution vectors K with unacceptably large entries. This can be addressed by introducing *redundancy*, i.e., additional centers of rotation, so that A becomes rectangular. Singular-value decomposition (SVD) is used to find the smallest-magnitude K , satisfying the constraints [20]. Conveniently, this algorithm allows us to place maximum bounds on the entries of A^{-1} [20].

Another method of dealing with ill-conditioning is to change the location of centers of rotations. If a given set of rotation centers produces an unstable matrix inversion, an alternative set can be attempted. Typically, the set of points the surface can rotate about will be a function of the particular motor/suspension kinematics.

3.2 Method B: Sinusoidal Controls

The previous manipulation method requires that the surface shift abruptly from one center of rotation to another. From a Fourier standpoint, the motion will contain significant harmonics, possibly causing resonance. One partial solution is to blend (low pass filter) the motion between consecutive rotations. That method requires that N rotations be executed before the correct average forces can be synthesized. Here we address these two issues by proposing an alternative method which (i) uses sinusoidal velocities on the surface's three degrees of freedom simultaneously (low spectral content) and (ii) achieves the desired average forces in a single, closed motion of the surface, involving its 3 degrees of freedom simultaneously.

Recall that Equation 3, 4, and 5 express the instantaneous surface velocity, frictional force, and average force (in one period) at location P , respectively, as a function of the surface's rigid velocities u_1 , u_2 , and u_3 . Here we consider these to be "sinusoids" of the form:

$$u_i(t) = \sum_{j=1}^N [a_{ij} \sin(jt) + b_{ij} \cos(jt)], \quad i = 1 \dots 3 \quad (13)$$

Where the a_{ij} and b_{ij} are $3N$ arbitrary constants. Assume, w.l.o.g., $T = 2\pi$.

3.2.1 The N=1 case

If $N = 1$, Equation 13 reduces to:

$$u_i(t) = a_i \sin(t) + b_i \cos(t), \quad i = 1 \dots 3 \quad (14)$$

with 6 free parameters $a_1, b_1, a_2, b_2, a_3, b_3$. From trigonometry:

$$a \sin(t) + b \cos(t) = \sqrt{a^2 + b^2} \cos(t + \tan^{-1} \frac{a}{b}) \quad (15)$$

i.e., each of the surface's degrees of freedom moves as a simple cosine with selectable amplitude and phase. We shown that in this case, $\bar{f}_{2d} = 0$ for any choice of the 6 free parameters. It is sufficient to show that

$$\int_{-\pi}^{\pi} \frac{\nu_s(P, t)}{|\nu_s(P, t)|} dt = 0$$

From Equation 3, the components of this integrand will be of the form:

$$g_1(t) = \frac{\alpha_1 s_1 + \alpha_2 c_1}{\sqrt{\alpha_3 + \alpha_4 s_2 + \alpha_5 c_2}} \quad (16)$$

where: $s_i = \sin(it)$
 $c_i = \cos(it)$

where $\alpha_i, i = 1 \dots 5$ are polynomials on $a_i, b_i, i = 1 \dots 3$, and the components of P . It can be shown that $g_1(t) = -g_1(t - \pi)$, so $g_1(t)$ integrates to zero in the $[-\pi, \pi]$ interval.

3.2.2 The N=2 case

If $N = 2$, Equation 13 reduces to:

$$u_i(t) = a_{i1} \sin(t) + b_{i1} \cos(t) + a_{i2} \sin(2t) + b_{i2} \cos(2t), \quad i = 1 \dots 3 \quad (17)$$

Equation 15 implies that each of the u_i 's is the sum of a sine to a double-frequency sine, with amplitudes and phases settable through a total of 12 free parameters.

Observation 3 *With $N = 2$, $\bar{f}_{2d} \neq 0$ in general.*

Consider the special case of $u_2 = u_3 = 0$, and $u_1(t) = \sin(t) + b \cos(2t)$. It can be shown that for some values of b this motion is time-asymmetric [14], i.e., it will produce non-zero forces in x direction.

With $N = 2$, the components of \bar{f}_{2d} will have the form of function $g_2(t)$ shown below:

$$g_2(t) = \frac{\beta_1 s_1 + \beta_2 c_1 + \beta_3 s_2 + \beta_4 c_2}{\sqrt{\beta_5 + r_2 + r_3 + r_4}} \quad (18)$$

$r_2 = \beta_6 s_2 + \beta_7 c_2$
 $r_3 = \beta_8 s_3 + \beta_9 c_3$
 $r_4 = \beta_{10} s_4 + \beta_{11} c_4$

where the β_i 's are polynomials on the 12 parameters and on P_x, P_y . The above expression is not integrable in closed-form, furthermore, the integrand has a pole at $a_{ij} = b_{ij} = 0$, making it inappropriate for taylor-series expansion. A clever alternative is to approximate the u_i sinusoids with triangular waves. Namely, consider the piecewise linear approximations $s^*(t) \simeq \sin(t)$ and $c^* \simeq \cos(t)$ defined as:

$$s^*(t) = \begin{cases} -2 - 2t/\pi & , \quad -\pi \leq t < -\pi/2 \\ 2t/\pi & , \quad -\pi/2 \leq t < \pi/2 \\ 2 - 2t/\pi & , \quad \pi/2 \leq t < \pi \end{cases}$$

$$c^*(t) = \begin{cases} 1 + 2t/\pi & , \quad -\pi \leq t < 0 \\ 1 - 2t/\pi & , \quad 0 \leq t < \pi \end{cases}$$

Scaling the triangular waveform by $12/\pi^2$ yields the best least-squares fit of a sine, reducing the the average absolute error from 15% to 8.5%, as shown in Figure 11. The spectrum of a triangular wave only contains odd harmonics, whose amplitudes are decay as $1/f^2$ [21], rendering it a “near” sine:

$$s^*(t) = s_1 + \frac{s_3}{9} + \frac{s_5}{25} + \dots$$

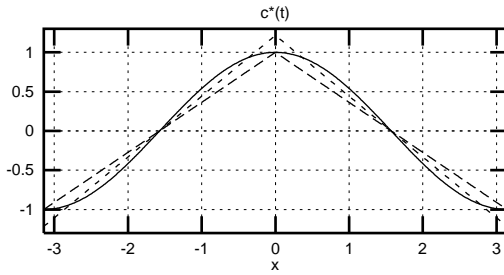


Figure 11: A plot of $\cos(t)$ and its its triangular approximation $c^*(t)$. Also shown is $c^*(t)$ scaled by the best least-squares fit amplitude $12/\pi^2$.

By defining similar triangular approximations s_2^* , and c_2^* for the double frequency sines and cosines, $g_2(t)$ above takes the following new form:

$$g_2^*(t) = \frac{k_1 + k_2 t}{\sqrt{k_3 + k_4 t + k_5 t^2}} \quad (19)$$

with k_i , $i = 1..5$ polynomials on the 12 free parameters and on P . From an integral table we obtain:

$$\int_0^1 g_2^*(t) dt = \frac{(k_1 k_5 - \frac{k_2 k_4}{2}) \log |c_3| + k_2 (c_1 - c_2)}{k_5^{3/2}}$$

$$c_1 = \sqrt{k_5 (k_3 + k_4 + k_5)}$$

$$c_2 = \sqrt{k_3 k_5} \quad (20)$$

There are 8 intervals in $[-\pi, \pi]$ where the linear coefficients for s^* , s_2^* , c^* , and c_2^* are constant, as shown in Figure 12. Separate expressions for the k_i 's must be computed for each interval, and Equation 20 must be broken down into a sum of integrals. Let $g_{2,j}^*$ denote g_2^* in interval j , $j = 1..8$. The following expression maps the 12 parameters and the coordinates of P to the average frictional force at P :

$$\bar{f}_{2d}(P, a_{ij}, b_{ij}) = \frac{\mu mg}{8} \sum_{j=1}^8 \int_0^1 g_{2,j}^*(t) dt \quad (21)$$

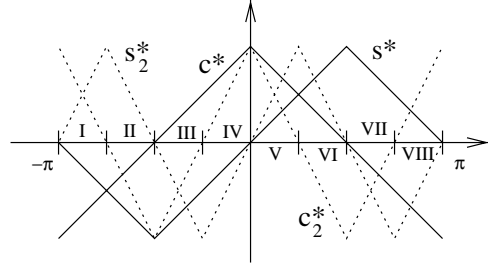


Figure 12: The s^* , c^* , s_2^* , and c_2^* triangular approximations in $[-\pi, \pi]$. Roman numerals I to VIII label the 8 distinct regions where the functions’ linear coefficients are constant.

The part manipulation problem requires that Equation 21 be inverted, namely, a set of $3N$ motion parameters must be computed which synthesize desired average forces at one or more parts’ locations. Let M be the number of parts. Then \bar{f}_{2d} is a $3N$ -to- $2M$ map. So long as the map’s rank is greater than $2M$, \bar{f}_{2d} can be inverted. The task of analyzing the map’s rank algebraically is too complex, especially given the form of Equation 20. Good invertibility however can be achieved with a large enough number of free parameters, i.e., redundancy. For the $N = 2$ case, 12 parameters allow for force to be specified in up to 6 distinct locations, though here we will consider it for just 2 (co-dimension 8). If more parts are involved, higher harmonics such as $\sin(3t)$ and $\cos(3t)$ are needed.

3.2.3 Iterative Inversion

Since a closed-form \bar{f}_{2d}^{-1} is out of the question, we will use a numeric method (gradient descent). We first construct a database of “initial guesses” as follows:

1. Generate a large number of random samples (P, a_{ij}, b_{ij}) , each with 14 entries (2 random part coordinates, and motion 12 parameters).
2. Pipe each sample through the forward map (Equation 21), and obtain the corresponding average force (\bar{f}_x, \bar{f}_y) .
3. Store the 12-tuple (a_{ij}, b_{ij}) in a search tree, with the 4-long vector (P, f_x, f_y) as the index.

the above process is illustrated in Figure 13. We use kd-trees [22] as the search/storage data-structure, for both its simplicity and efficiency. Namely, given a query (P', f'_x, f'_y) , the kd-tree can retrieve the data linked to the closest-matched index in logarithmic time.

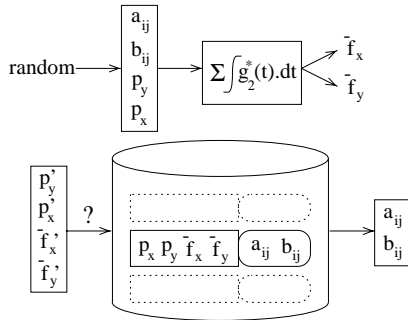


Figure 13: *Top:* a random 14-tuple (P, a_{ij}, b_{ij}) is piped through the forward map, yielding the components \bar{f}_x and \bar{f}_y of the average force. A new entry indexed by (f_x, f_y, P) and associated data (a_{ij}, b_{ij}) is inserted into a search structure (kd-tree). *Bottom:* A 4-long query vector is presented to the search structure; the which quickly locates the entry with the closest-matching index and retrieves the associated data.

For simplicity, we will consider the case of force specification at two locations P_1 and P_2 . Let the desired forces be f_1 and f_2 , respectively. To find the required 12 parameters a_{ij}, b_{ij} we proceed as follows:

1. Define \bar{P} as the mid-point between P_1 and P_2 , and \bar{f} as the average of f_1 and f_2 , Figure 14.
2. Query the kd-tree with (\bar{P}, \bar{f}) , and retrieve the closest “initial guess” vector $Q = (a_{ij}, b_{ij})$.
3. Numerically estimate the Jacobian $J = \partial \bar{f}_{2d} / \partial Q$, evaluated at (\bar{P}, Q) .
4. Let Δf represent the 4-long error vector with components $f_i - \bar{f}_{2d}(P_i, Q)$, $i \in \{1, 2\}$. Descend along the error gradient by setting $Q = Q - J^{-1} \Delta f$.

5. Repeat from (3) till Δf 's magnitude (the error in force) falls below a threshold.

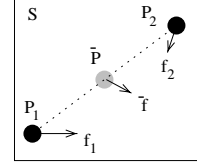


Figure 14: *Given 2 (or more parts), a query vector (\bar{P}, \bar{f}) as the average of all part positions and desired forces.*

3.3 Manipulability and Scalability

Given one or more part positions, what are the average force directions easiest to synthesize at each part, i.e., requiring parameters with the least magnitude. This notion is akin to that of a velocity ellipsoid at the tip of a serial manipulator. The complex dependency of the forward map on the 12 parameters makes this expression opaque to any analysis. Linearization is also impossible since the Jacobian of \bar{f}_{2d} is not defined at the origin (zero parameter vector). In practice, the onset of non-invertibility is signaled by an increase in the number of iterations taken by the numerical solver per step. The most ill-conditioned force direction can be estimated from the forward map in the following fashion: pipe a large number of randomly-chosen parameter vectors through the forward map. The worst directions are those which yield the highest input/output magnitude ratios (shrinkage of output signals loss of rank in that direction).

The 2nd manipulation method can produce desired forces with a single, sinusoidal motion, featuring a shorter, more band-limited motion with respect to the 1st method. This comes with the price of no good theory for the system's invertibility and difficult scalability. Namely, with the 1st method, incorporating new centers of rotation (adding redundancy) does not affect the structure of the solver. Here, adding more free parameters is obtained with higher order harmonics. For example, adding the 3rd harmonic would increase the number of free parameters from 12 to 18. New expressions for the k_i 's (Equation 19) need to be derived, e.g., using a symbolic engine, for each of the constant coefficient regions in the triangular approximations. This number jumps from 8 to 16, and is in general 2^r , where r is the order of the highest harmonic used.

3.4 Sensorless Manipulation

Divergence relates to the shrinkage or expansion of a unit area element flowing through the field [18]. In manipulating parts with force fields, non-zero divergence can be used to automatically reduce the uncertainty about a part’s location and/or orientation (see [2] for an example). Unfortunately, the divergence of a frictional force field $\nabla\phi = \phi_{x,x} + \phi_{y,y}$ is zero everywhere. Linearity implies that average force fields obtained by N rotations and/or by a continuous (sinusoidal) motion will be also divergence-free, ruling out sensorless manipulation.

4 Dynamic Simulation

4.1 1st method: parallel sorting of 10 parts

A set of 5 grey and 5 black parts is scattered at random over the surface. The goal is to move all black parts to the left edge and all grey parts to the right edge of the surface. We use a very simple type of motion planning. At every step, apply a force along $-x$ to the black parts and along $+x$ to grey ones. We will use the method of sequencing rotations, we 24 rotation centers. With a total of 20 constraints per step, this gives us 4 degrees of redundancy. SVD is used to compute a bounded solution vector. As parts are pushed to their respective sides, they come inevitably close to each other. Such situations ill condition the system, causing SVD’s solution to produce actual forces which deviate (considerably) from the desired ones. Interestingly, this produces a convenient effect – parts coming close to each other will deviate from their horizontal paths and “dodge” each other, avoiding collisions. Since this is a side effect of SVD, there are no motion planning guarantees here... Snapshots of a dynamic simulation of this example are shown in Figure 15.

4.2 2nd method: trajectory following for 2 parts

Consider two parts P_1 and P_2 which must move with constant speed along distinct pre-specified polygonal trajectories on S . A simple PD controller is used for each part to compute the force needed to reduce the error (deviation in trajectory and desired speed). We use the 2nd method (sinusoidal control) to synthesize the forces computed by the PDs. Snapshots of a dynamic

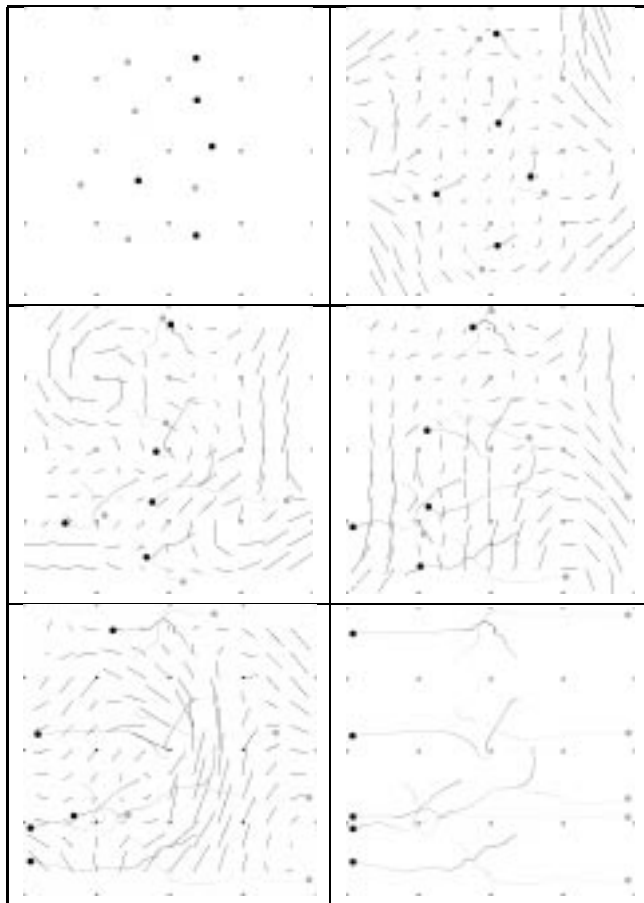


Figure 15: *The part sorting simulation. The sequence should be read left to right, top to bottom. 5 dark and 5 grey disks appear in the first snapshot at their original locations. The goal is manipulate them in parallel so at the end all black go to the left and all grey go to the right. Force field vectors are shown for each snapshot. Also shown are the trails left by the motion of each part. Note that only strictly horizontal force is applied to the parts – the deviations are caused by ill-conditioning in the linear system.*

simulation of this example are shown in Figure 16. Numeric inversion typically iterates less than 5 times per step, allowing for real-time visualization.

5 Hardware Experiments

A prototype of the device shown in Figure 17 has been built out of rather inexpensive parts – the motors are

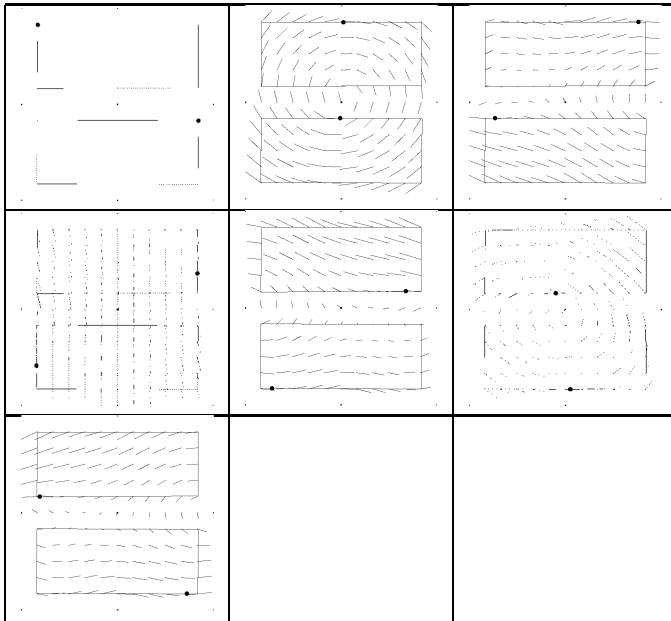


Figure 16: *The trajectory following simulation. Read from left to right, top to bottom. At the top left snapshot, two parts are shown (dark disks) at the beginning of their planned rectangular trajectories. Force field vectors are shown at each snapshot.*

recycled disk-drive voice coils; the plate is a thin slab of kitchentop material.

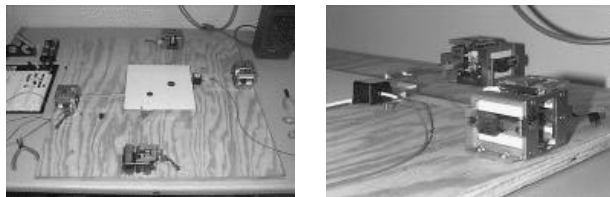


Figure 17: *Left: a prototype of the device built from inexpensive parts. Right: a closeup of two motors; also shown is an accelerometer attached to the plate.*

A simple experiment was performed with an earlier version of the the shaker table, for which the surface was an aluminum baking tray. The experiment consisted in driving the surface to a periodic, time-asymmetric [14] rotation about its center, at a few 10s of Hz. We placed a 1 cent coin on the surface, and observed it describe a constant-velocity motion along a

circular trajectory, demonstrating part feeding circles around a circular path. Centrifugal effects slowly increase the trajectory’s radius causing the coin to eventually touch (or fall off) the surface’s boundary. Snapshots of this experiment are shown in Figure 18.



Figure 18: *Snapshots of the coin rotation experiment: an early version of the shaker table is shown, with a baking tray used as the surface. As the tray rotates time-asymmetrically about its center, a 1 cent coin moves with constant speed along a circular trajectory.*

6 Conclusion

We have presented a novel approach to part manipulation in the plane based on the time-averaging of coulomb friction at one or more points under a rigid, closed motion of the surface. The main feature of the current approach is its mechanical simplicity. Two motion computation methods have been proposed: one involves the execution of a finite number of rotations about a fixed set of points. The other involves computing the parameters for a smooth motion of the surface, where each of its degrees of freedom moves in sinusoidal fashion. The second method has the advantage of much lower bandwidth, being less likely to induce suspension resonance; however its solution requires a rather complex, non-scalable iterative inversion machinery. Dynamic simulation results for two parallel manipulation instances have been presented, illustrating the effectiveness of both methods. Simple experiments with an existing prototype have been presented. The next major step in this research involves developing the motor control and image processing code so as to integrate the device with a supervising PC.

Acknowledgements

This work was supported by NSF grant CDA-9726389: “Challenges in CISE: Planning and Control for Massively Parallel Manipulation”. Thanks to Ken Goldberg, Karl Böhringer, Wolfgang Zesch, and Yan Zhuang for useful discussions, suggestions, and comments.

References

- [1] K. Böhringer, B. Donald, R. Mihailovich, and N. MacDonald. Sensorless manipulation using massively parallel micro-fabricated actuator arrays. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.
- [2] L. Kavraki. Part orientation with programmable vector fields: Two stable equilibria for most parts. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- [3] J. Suh, S. Glander, R. Darling, C. Storment, and G. Kovacs. Combined organic thermal and electrostatic omnidirectional ciliary microactuator array for object positioning and inspection. In *Proc. Solid State Sensor and Actuator Workshop*, Hilton Head, NC, June 1996.
- [4] J. Luntz, W. Messner, and H. Choset. Parcel manipulation and dynamics with a distributed actuator array: the virtual vehicle. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- [5] S. Konishi and H. Fujita. A conveyance system using air flow based on the concept of distributed micro motion systems. *Journal of Micro-electromechanical Systems*, 3(2):54–58, June 1994.
- [6] F. Moesner and T. Higushi. Devices for particle handling by an ac electric field. In *IEEE Micro-electromechanical Systems*, pages 66–71, January 1995.
- [7] C. Liu, T. Tsao, P. Will, Y. Tai, and W. Liu. A micro-machined magnetic actuator array for microrobotics assembly systems. In *International Conference on Solid-State Sensors and Actuators*, Stockholm, Sweden, 1995.
- [8] J. Canny and K. Goldberg. RISC for industrial robots: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.
- [9] G. Boothroyd. *Assembly automation and product design*. Marcel Dekker, Inc., New York, NY, 1991.
- [10] J. Krishnasamy, M. Jakiela, and D. Whitney. Mechanics of vibration – assisted entrapment with application to design. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, May 1995.
- [11] Vincent Hayward. Part manipulation with a vibratory surface. private communication.
- [12] K. Böhringer, V. Bhatt, and K. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 1995.
- [13] D. Reznik, J. Canny, and K. Goldberg. Analysis of part motion on a longitudinally vibrating plate. In *International Workshop On Intelligent Robots and Systems*, Grenoble, France, September 1997.
- [14] D. Reznik and J. Canny. The coulomb pump: A novel parts feeding method using a horizontally-vibrating surface. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.
- [15] D. Reznik and J. Canny. A flat rigid plate is a universal planar manipulator. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998.
- [16] M. Mason. *Mechanics of Manipulation*. MIT Press, Cambridge, MA, 1997.
- [17] J. Craig. *Introduction to robotics, mechanics and control*. Addison-Wesley, Reading, MA, 2nd edition, 1989.
- [18] H. Schey. *Div, grad, curl, and all that*. W.W. Norton, New York, NY, 2nd edition, 1992.
- [19] R. Murray, Z. Li, and S. Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, Boca Raton, FL, 1994.
- [20] W. Press et al. *Numerical recipes in C*. Cambridge University Press, New York, NY, 2nd edition, 1995.
- [21] S. Haykin. *An introduction to analog and digital communications*. John Wiley & Sons, Inc., New York, NY, 1989.
- [22] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.