A Flat Rigid Plate is a Universal Planar Manipulator *

Dan Reznik John Canny

EECS Dept., UC-Berkeley, CA 94720-1776 USA

Abstract

We consider the problem of parallel part manipulation, i.e., the simultaneous position and orientation control of one or more parts in a bounded region of the plane. We propose a novel, minimalist device, based on a single horizontally-vibrating flat plate. We show that a closed rigid motion of the plate, involving its 3 dofs, can be computed which produces desired average forces at a finite number of points, e.g., parts' locations. This implies that one or more parts can follow independent trajectories simultaneously, as they interact with a single vibrating plate. This is in sharp contrast with more complex designs such as massively-parallel actuator arrays and/or prehensile manipulation. Dynamic simulation is used to test the current method in two parallel part manipulation examples. A prototype of the device has been built with inexpensive parts; physical implementation of the proposed method is currently underway.

1 Introduction

We consider the problem of parallel part manipulation, i.e., the simultaneous position and orientation control of one or more parts in a bounded region of the plane. Difficulties with planning, execution speed, and end-effector clutter deem traditional pick-and-place methods inappropriate, especially if parts are numerous and their size/separation is small. Research on *active surfaces* addresses these issues by proposing that the surface on which parts rest should double as the actuation mechanism. One popular design concept is that of a massively-parallel array of actuators. A desired manipulation force/torque is synthesized by addressing a set of actuators lying in the vicinity of, or directly under, a part of interest. Many such devices have been recently proposed, differing essentially in actuator type [1, 2, 3, 4, 5, 6].

Array-based active surfaces require large actuator densities to achieve good force synthesis resolution. One such device contains an excess of 10,000 micro-actuators over an area of a few squared-cm [1]. The large number of tiny moving actuators can pose both fabrication and maintenance difficulties. In the spirit of minimalism [7], we address this issue by searching for a radically simpler active surface design which retains force synthesis capabilities similar to array-based devices. *Our main contribution is to show that a single horizontally-vibrating flat plate is such a device.* Owing to a special property of Coulomb friction (see below), we show that a closed horizontal motion of a flat plate can be computed which gives rise to desired frictional forces (averaged over the motion) at a finite number of points on the plate.

* Support provided in part by NSF grant CDA-9726389: "Challenges in CISE: Planning and Control for Massively Parallel Manipulation".

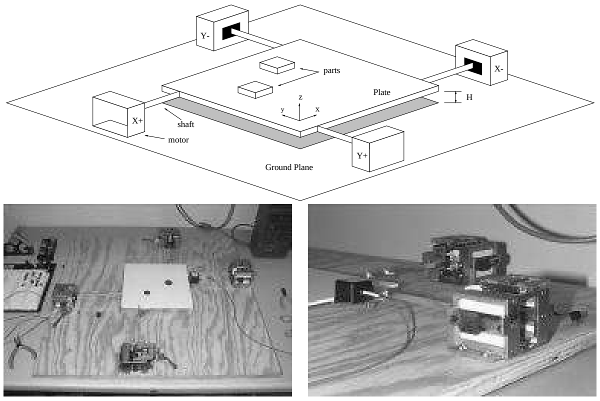


Figure 1: Top: Model of the proposed device – a flat horizontal plate, actuated by motors, is constrained to move horizontally. Parts are shown resting on the plate. Bottom left: a prototype of the device we've built with inexpensive parts: the plate is a thin square of smooth kitchen-top material; the motors are hard disk voice coils. Bottom right: a closeup of two motors, showing a motion sensor attached to the plate.

A model of our proposed device is shown in Figure 1. A horizontal, flat plate is actuated along its three degrees of freedom (two translations and one rotation) by linear motors, e.g., voice coils. The plate vibrates rapidly, causing parts to displace/rotate as they interact (via dynamic friction) with the plate. At the heart of our result is a sequence of 3 observations:

- The magnitude of dynamic friction is constant and independent of relative velocity.
- A rotation of the plate about a given point (as produced by any instantaneous rigid velocity) gives rise to a frictional force *field* in which forces at every point are constant and tangential with respect to the center of rotation.
- The family of such fields is not closed under addition (unlike the family of rigid velocity fields), i.e., the sum of N such fields is N-dimensional. In contrast, the family of rigid velocity fields is 3-dimensional.

We consider a closed motion of the plate composed of a finite sequence of small rotations, each about a known center. Centers of rotation are chosen away from the parts' locations (to avoid stiction effects). Assuming part motion is negligible throughout the sequence, the net force applied to the part after the plate completes the sequence is roughly the sum of each individual frictional force field, evaluated at the part's location. Owing to non-additivity of rotation-induced frictional force fields (above), we show that the problem of synthesizing desired average forces at parts' locations reduces to that of solving a linear system for the "intensity" (e.g., the duration) of each rotation. By specifying the *curl* of the net frictional force field (a simple addition to the linear system being solved), we can control part rotation at one or more points.

A limitation of the current approach is that the generated force fields are divergence-free, and therefore not appropriate for force-field-based sensorless manipulation [8]. In the setup envisioned, permanent sensing is required (e.g., low-level vision) which informs the algorithm about parts' positions.

Dynamic simulation is used to verify the algorithmic/numerical validity of our method. In one experiment, two parts follow user-defined, independent trajectories, simultaneously, as they interact with the plate. In a second experiment, a set of 6 black and 6 white parts, initially scattered at random over the plate, are sorted, in parallel. A prototype of the device has been built out of rather inexpensive parts, as shown in Figure 1. The plate is a thin slab of smooth kitchen-top material; the motors have been removed from old hard disk drives. Currently, PC-based control and sensing software is being developed which will be integrated with existing hardware. While many practical difficulties are expected, we feel the current approach is indeed feasible and will afford enormous simplifications to the current state of the art in active surface design.

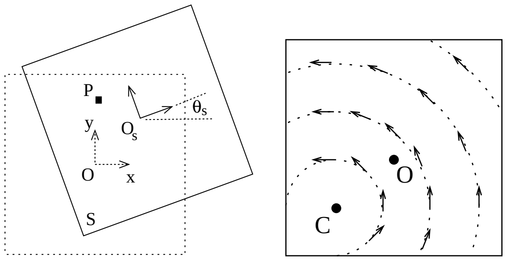
1.1 Related work

Manipulation via hardware minimalism is exploited by vibratory bowl feeders and certain vibrations-based devices such as the APOS feeder [9]. In the spirit of our work, Hayward et al. [10] have performed experiments with a horizontally/vertically vibrating plate in the context of automatic part orientation – they look for plate control waveforms which create interesting part energy minima. Böhringer et al. [11] document experiments with a transversally-vibrating plate in an attempt to automatically gather particles at vibration nodes.

This research is a continuation of our work on parts feeder design. In a companion paper [12], a parts feeder based on a 1d horizontally-vibrating plate is presented which also exploits the independence of Coulomb friction on relative velocity. In [13] we address the problems of motor frequency response and device resonance – an alternative motion computation method is proposed which prescribes a low-bandwidth, sinusoidal motion for the plate, involving its 3 dofs simultaneously.

The remainder of this paper is organized as follows: In Section 2 we describe device kinematics, the friction model, and the non-additivity of frictional force fields. In Section 3 we describe the plate motion computation method based on a sequence of rigid rotations. In Section 4 we present two dynamic simulation results for two parallel manipulation examples. Section 5 presents the conclusions and directions for future work.

2 Part Manipulation Principles



(a) (b)

Figure 2: (a) The dotted square represents S at its home position, with its center at the origin O; The solid square represents S translated by O_s and rotated by θ_s. The part is at point P. (b) A rigid motion of S about C yields a frictional force field of tangential vectors of constant magnitude.

Let S (for Surface) denote the plate; S is constrained to move in its own plane (3 dof). Let $O_s = (x_s, y_s)$, and θ_s denote S's translation and orientation relative to XY, Figure 2(a). Consider a smooth plate motion $S(t) = [O_s(t), \theta_s(t)]^t$ which is *closed*, i.e., $S(0) = S(T)$, for some $T > 0$. The plate's tangential velocity ν_s at a point P is:

$$\nu_s(P,t) \quad = \quad \dot{O}_s(t) + \dot{\theta}_s(t) \times [P - O_s(t)] \tag{1}$$

Let us consider points P much larger in magnitude than the plate's maximum translation, i.e., $||P|| \gg \max_t ||O_s(t)||$, so that $P - O_s(t) \simeq P$. Equation 1 reduces to:

$$\begin{aligned} \nu_s(P,t) &= \dot{O}_s(t) + \dot{\theta}_s(t) \times P \\ &= [u_1(t) - y u_3(t), u_2(t) + x u_3(t)]^t \end{aligned} \tag{2}$$

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 $\int_0^T u_i(t)\, dt = 0,\ i = 1 \ldots 3$.

Consider a part, modeled as a point mass lying at P on S. Let ν_s and ν_p denote, respectively, plate's and part's instantaneous velocities.

Assumption 1 *P is always sliding on S – relative accelerations at P give rise to inertial forces above the threshold of static friction [14], for most [1] $t \in [0, T)$.*

The Coulomb dynamic frictional force f_{2d} is of constant magnitude $\mu m g$ and opposite to the part's motion relative to the surface [14], i.e.:

$$f_{2d} = \mu m g \, \frac{(\nu_s - \nu_p)}{||\nu_s - \nu_p||} \tag{3}$$

[1] $S(t)$ 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; while we ignore stiction, 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 3 reduces to:

$$f_{2d}(P,t) = \mu m g \frac{\nu_s(P,t)}{||\nu_s(P,t)||} \tag{4}$$

Assumption 2 *The part's position P does not change appreciably during the plate's closed motion.*

An upper bound for the part's maximum displacement in T seconds given frictional forces is $||\Delta P||_{max} = \mu g T^2$. We assume this quantity negligible with respect to $||P||$.

So the average frictional force $\bar{f}_{2d}$ applied to the part (with its position presumed constant) is the integral of the instantaneous frictional forces at P divided by the length of the motion:

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

2.1 Instantaneous force fields

Chasles' theorem [15] 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:

$$[c_x, c_y, w] = [-u_2/u_3, u_1/u_3, u_3] \tag{6}$$

If the $u_i(t)$'s are directly controllable, we can choose 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 (c_x, c_y, w) of S. The tangential velocity ν_s at point $P = (x,y)^t$ is $w \times (P - C)$, or:

$$\nu_s(P) = w \begin{bmatrix} c_y - y \\ x - c_x \end{bmatrix} \tag{7}$$

With each point on the plate moving tangentially, a stationary part at point P will experience an instantaneous frictional force of the type $f_{2d}(P) = \mu m g \hat{\nu_s}(P)$, where $\hat{\nu_s}$ is the unit vector along the tangential velocity. So any rigid plate velocity gives rise to an *instantaneous force field* of the type illustrated in Figure 2(b): force at every point is perpendicular to the radial line, and of fixed magnitude $\mu m g$. Low tangential accelerations near C may cause the part to stick to the surface. We ignore this effect by assuming the rotation center is far enough from the part.

2.2 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:

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

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 <u>not</u> closed under addition: Let $\phi_1, \phi_2 \in \Phi$. Then $\phi_1 + \phi_2 \notin \Phi$. Namely, $\neg\exists k_3, C_3$ 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 3.

2.3 The parallel manipulation task

Consider M parts distributed over S at known locations P_i, $i = 1 \ldots M$. Let $\bar{f}_i(k)$ be a desired average force for part i at a given instant. The goal is to compute a closed motion $S(t)$ which yields $\bar{f}_{2d}(P_i) = \bar{f}_i$, $\forall i$, as shown in Figure 4.

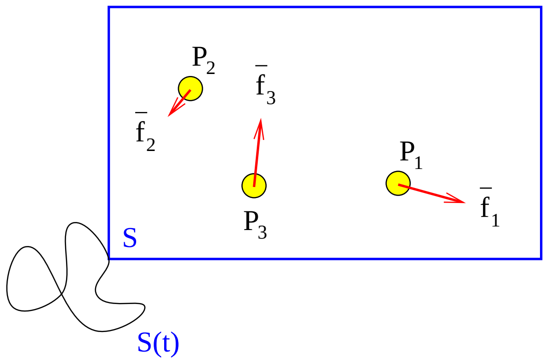


Figure 4: The parallel manipulation task consists in computing a closed motion $S(t)$ for the surface S which can generate desired frictional forces $\bar{f}_i$ (averaged over the motion) at each of the parts' locations P_i.

The setup envisioned is as shown in Figure 5. 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 $\bar{f}_i$ 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 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.

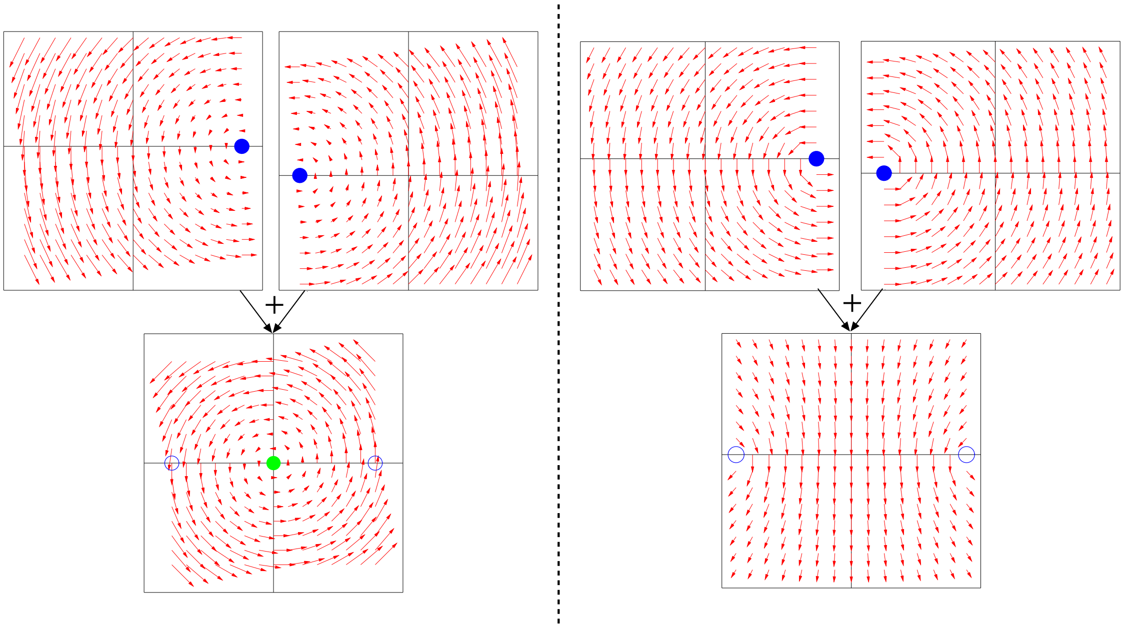


Figure 3: Left: the sum of two velocity fields is a velocity field. Right: the sum of two frictional force fields is not in the same family.

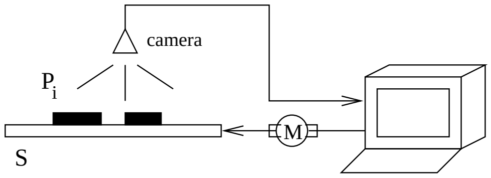


Figure 5: Closed-loop manipulation: a camera sends images to a computer, which sends commands to a motor which actuates the surface which moves the parts.

3 The Method of Sequencing Rotations

Consider N consecutive small-angle rotations of the surface, $N \geq 2M$, each about a distinct point C_j, $j = 1 \ldots N$. Let the duration of each rotation be an independent k_j, such that $\sum k_j = T$. Let $\phi_j = \mu mg \, \phi_{k_j, C_j}$. For simplicity, we assume the entire sequence of rotations is a closed motion of the surface. In reality, each motion in the sequence is a time-asymmetric rotation [12] (scaled by a design parameter) about the associated center.

Since parts move negligibly during the entire motion (Assumption 2), or, equivalently, that the ϕ_j's are slow-varying around the P_i's, we can state:

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:*

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

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 $K = (k_1 \ldots k_N)$ which satisfies the linear system $A.K = b$, with:

$$\begin{aligned} A_{ij} &= \begin{cases} -a(i,j) & , \quad i = 1, \cdots M \\ b(i-M, j) & , \quad i = M+1, \cdots 2M \end{cases} \\ & \quad j = 1 \cdots N \\ b_i &= \frac{N}{\mu mg} \begin{cases} \bar{f}_i^x & , \quad i = 1, \cdots M \\ \bar{f}_{i-M}^y & , \quad i = M+1, \cdots 2M \end{cases} \end{aligned} \tag{10}$$

and:

$$\begin{aligned} a(i,j) &= (p_i^y - c_j^y)/r_{ij} \\ b(i,j) &= (p_i^x - c_j^x)/r_{ij} \\ r_{ij} &= ||P_i - C_j|| \end{aligned}$$

3.1 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 $K = (k_1, k_2, k_3, k4)$ and execute 4 rotations consecutively about each of C_i's (Figure 6 with each such motion "scaled", in duration, by the computed k_j. The components of the 4×4 linear system become:

$$A_{4\times4} = \begin{bmatrix} -.316 & -.707 & .707 & .316 \\ -.949 & -.394 & .141 & .707 \\ .949 & -.707 & -.707 & .949 \\ .316 & -.919 & -.990 & .707 \end{bmatrix}$$

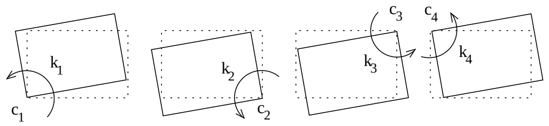


Figure 6: The surface executes 4 rotations about each of its corners, say, in CCW order. Each rotation is scaled (e.g., in duration) by a computed k_j.

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

By inverting A, compute $k = A^{-1}.b$ and obtain $K = (.58, -.39, .48, -.095)^t$. As shown in Figure 7, the resulting force field is as desired at the part's locations.

3.2 Rotation control

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 of an infinitesimal area element flowing along the field [16]. A simple expression yields the curl of a rotation-induced force field $\phi_{k,C}$:

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

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 ϕ_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:

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

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.

3.3 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 in no need for a fourth rotation about C_4. The resulting 3×3 linear system becomes:

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

Which yields:

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

Solving the above yields $K = (-.59, -.27, -.53)$. The resulting field, illustrated in Figure 7, 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.

3.4 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 rotation-induced 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 weaken the superposition assumption is to rotate about the C_j's in *descending* order of rotation priority, ρ_j, above. In Section 3.1, $(\rho_1 \cdots \rho_4)^t = (.16, .031, .43, .34)^t$, 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 [17].

3.5 Ill-conditioning

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

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

In 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. Consequently, in that region, A's rank will be limited to 2, the rank of the family of constant fields.

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. In our implementation, singular-value decomposition (SVD) is used to find the smallest-magnitude K, satisfying the constraints [18]. Conveniently, this algorithm allows us to place maximum bounds on the entries of A^{-1} [18].

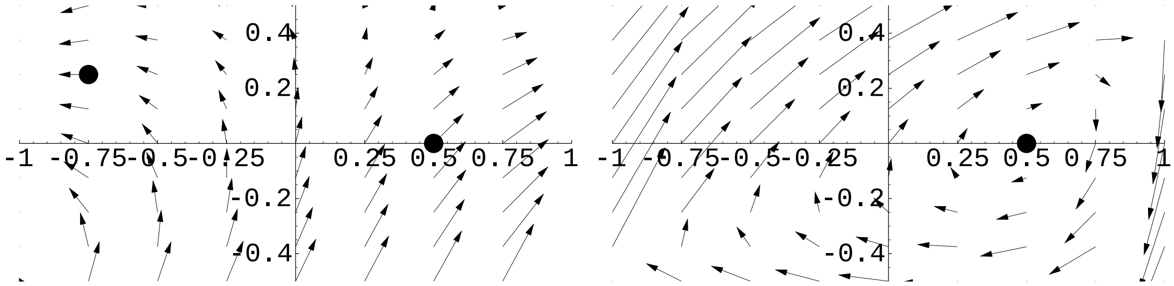


Figure 7: Left: average force field after 4 rotations about S's corners. The field points in the desired directions at the location of each part, shown as black disks. Right: average force field obtained after 3 rotations about C_1, C_2, and C_3 in sequence. Zero force and negative curl are desired at the part's location $(0.5, 0)$. The result is a CW whirlpool centered on the part.

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 specified. Typically, the set of points the surface can rotate about will be a function of the particular motor/suspension kinematics.

3.6 Sensorless manipulation

Divergence relates to the shrinkage or expansion of a unit area element flowing through the field [16]. 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 [8] for an example). Unfortunately, the divergence of a frictional force field $\nabla\phi = \phi_{x,x} + \phi_{y,y}$ is, by inspection, zero everywhere. Linearity implies that average force fields obtained by N rotations is also divergence-free, ruling out sensorless manipulation, hence the feedback loop setup illustrated in Figure 5.

4 Dynamic Simulation Examples

Dynamic simulation was used to informally verify the effectiveness of our method. Two parallel manipulation examples were considered, involving 2, and 10 parts, respectively. Part(s) are modeled as point-masses. Specific plate motions (as computed by our method) give rise to local tangential velocities which accelerate parts in given directions. Actual part motion is computed via numeric integration of frictional forces.

4.1 Trajectory following for 2 parts

Two polygonal trajectories are specified for two parts placed on the plate. Each part must follow their trajectory with constant speed. The motion is broken down into small steps. At each step, a simple PD controller (one per part) computes force needed to maintain the part in its trajectory and with the desired speed. This information is fed to the plate motion computation algorithm which then returns the required set of plate rotations. In this case, 9 rotations are used per closed motion, yielding 5 degrees of redundancy in the matrix inversion. Graphical output produced by the simulation is shown in Figure 8.

4.2 Parallel sorting of 10 parts

A set of 5 grey and 5 black parts is scattered at random over the plate. The goal is to move all black (resp. grey) parts to the plate's left (resp. right) edge. We use a very simple type of "sorting" algorithm: at every step, apply a force along $-x$ (resp. $+x$) to the black (resp. grey) parts. The 24 rotation centers are used per complete plate motion. 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. The resulting ill-conditioning causes SVD's solutions to produce net forces which deviate 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. Snapshots of a dynamic simulation of this example are shown in Figure 9.

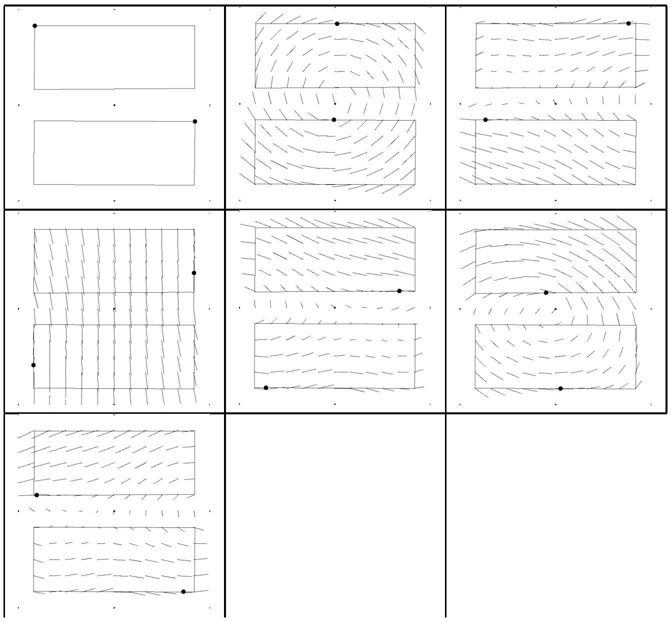


Figure 8: The trajectory following simulation. Read from left to right, top to bottom. At the top left snapshot, two parts are shown (dark disks) at the beginning of their planned rectangular trajectories. Net force field vectors (assuming superposition) are shown at each snapshot.

5 Conclusion

A novel design for a parallel, 2d part manipulator has been proposed which is mechanically simpler than actuator-array counterparts but which requires a more complex force synthesis algorithm. A prototype of the device has been built with inexpensive parts. Physical implementation of the current method is now underway. Low-level software routines are being written for both motor control and vision-based sensing of part position. Theory underlying our method has been presented in a rather informal way. A more formal treatment of order dependent effects and part dynamics is desirable. A generalization of the current method for manipulation in 3d is being worked on.

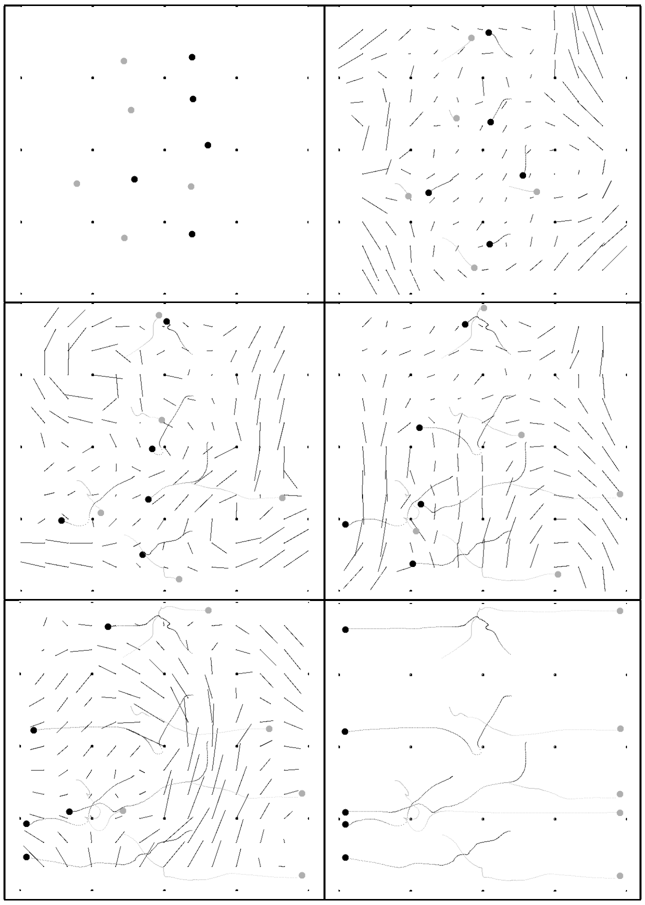


Figure 9: The part sorting simulation. The sequence should be read left to right, top to bottom. 5 dark and 5 grey disks appear in the first snapshot at their original locations. The goal is manipulate them in parallel so at the end all black go to the left and all grey go to the right. Force field vectors 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.

References

[1] K. Böhringer, B. Donald, R. Mihailovich, and N. MacDonald. Sensorless manipulation using massively parallel microfabricated actuator arrays. In *IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[2] 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.

[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] F. Moesner and T. Higushi. Devices for particle handling by an ac electric field. In *IEEE Micro-electromechanical Systems*, pages 66–71, January 1995.

[6] 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.

[7] 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.

[8] 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.

[9] G. Boothroyd. *Assembly automation and product design.* Marcel Dekker, Inc., New York, NY, 1991.

[10] N. Tran, K. Chan, and V. Hayward. Object behavior using a vibrating plate testbed for part presentation research. EE304-494 project writeup, McGill University, December 1995.

[11] 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.

[12] 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.

[13] D. Reznik and J. Canny. Universal part manipulation in the plane with a single horizontally-vibrating plate. In *International Workshop on Algorithmic Foundations of Robotics*, Houston, TX, March 1998.

[14] M. Mason. *Mechanics of Manipulation.* MIT Press, Cambridge, MA, 1997.

[15] J. Craig. *Introduction to robotics, mechanics and control.* Addison-Wesley, Reading, MA, 2nd edition, 1989.

[16] H. Schey. *Div, grad, curl, and all that.* W.W. Norton, New York, NY, 2nd edition, 1992.

[17] R. Murray, Z. Li, and S. Sastry. *A mathematical introduction to robotic manipulation.* CRC Press, Boca Raton, FL, 1994.

[18] W. Press et al. *Numerical recipes in C.* Cambridge University Press, New York, NY, 2nd edition, 1995.