BUILDING A UNIVERSAL PLANAR MANIPULATOR

Dan Reznik
Emil Moshkovich
John Canny

Abstract Distributed manipulation devices make use of a large number of actuators, organized in array fashion, to manipulate a small number of parts. Inspired by minimalism we look at a complementary question: can a device with few degrees of actuation freedom be used to flexibly manipulate a large number of parts? In previous publications we have shown that a single horizontally-vibrating plate is just such a device. This suggests that actuator count can be traded for control complexity. In this paper we review our theory of minimalist manipulation and describe implementation solutions towards a working prototype.

1 INTRODUCTION

Distributed manipulation devices make use of a large number of actuators, organized in array fashion, to manipulate a small number of parts (Luntz et al., 1998; Kavraki, 1997; Böhringer et al., 1998). Inspired by minimalism in robotics (Canny and Goldberg, 1994), in our own research we have looked at a complementary question: can a device with few degrees of actuation freedom be used to independently manipulate a large number of parts? The well-known bowl feeder (Boothroyd, 1991) achieves just that at the expense of non-programmability, i.e., its function – e.g., part presentation at known orientation – is determined once and for all by its design.

In previous publications (Reznik and Canny, 1998a; Reznik and Canny, 1998b; Reznik and Canny, 1998c) we have shown that, surprisingly, a *programmable* parallel manipulation device – a Universal Planar Manipulator (UPM) – can be built out of a single flat plate. In the approach proposed, a horizontally-vibrating plate manipulates (i.e., translates and rotates) parts via frictional interactions (of the sliding type) with the latter. Perhaps the simplest form of this type of manipulation is rectilinear *part feeding*. In (Reznik and Canny, 1998a) we show that by introducing pump-like asymmetries on horizontal vibrations of a flat plate, parts placed on its surface are propelled forward at a well-known speed. Here we use similar friction-based actuation principles to achieve parallel part manipulation.

The *parallel manipulation problem* we consider is that of translating and/or rotating multiple parts along independent trajectories, e.g., as required by a high-level task such as part presentation, sorting, or assembly/mating. Here we ignore part rotation, focusing on translation only (part orientation can be achieved under the same method, see (Reznik and Canny, 1998a)). The basic problem solved is to compute a suitable *closed motion* of the plate which creates "correct" frictional forces under each part. I.e., friction averaged over the entire motion causes each part to move a discrete step along the part's independent trajectory. If this procedure is iterated over quickly, smooth parallel manipulation is achieved.

An important contribution has been to show that a sequence of plate rotations about a known set of centers is just such a desired closed motion (Reznik and Canny, 1998c). Each iteration of the manipulation algorithm reduces to (i) locating parts, (ii) obtaining the desired steps, (iii) computing the *duration* of each rotation, and (iv) executing the rotation.

An important issue is that the current approach requires that parts' positions be known at all times, e.g., through image sensing. Indeed, this precludes open-loop, sensorless manipulation, which has been recently investigated as an application for distributed-manipulation devices.

The main contribution of this paper to the field of distributed manipulation is to show that device complexity (indeed actuator count) can be dramatically reduced and traded for more sophisticated control.

1.1 RELATED WORK

A number of researchers have studied manipulation devices based on a single horizontal plate. In (Hayward et al., 1995), vertical oscillations of a plate are used to trigger resonances on coins so as to make them stay upright. In (Böhringer et al., 1995), the nodes of a vibrating plate are used to gather pellets and orient polygonal objects. In (Swanson et al., 1995), closed horizontal motions of a flat plate are used to "ratchet" a part of known shape to a desired final orientation.

One researcher has reported experimental results utilizing our method in the simpler case of parts feeding (Quaid, 1999). Another researcher has designed a distributed manipulation device (Frei and Wiesendanger, 1999) based on sliding friction actuators.

This paper is organized as follows: in Section 2 we concisely review the theory behind our manipulation algorithm. In Section 3 we address practical implementation issues in building a prototype of our device. In Section 4 we present preliminary experimental results. Conclusions are presented in Section 5.

2 REVIEW

2.1 THE MANIPULATION ALGORITHM

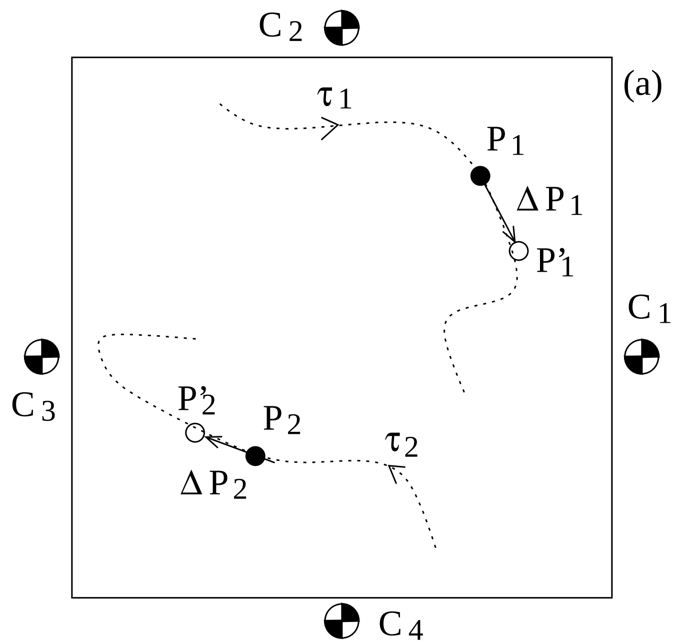
The manipulation problem being addressed is illustrated in Figure 1(a). N parts lie at known positions P_i, $i = 1 \ldots N$ within a bounded area of the plane. A desired small, straight motion $\Delta P_i = (\Delta P_i^x, \Delta P_i^y)^T$ is prescribed for each part, e.g., along a trajectory associated with a high-level task such as part mating, sorting, etc. Define a set of points C_j, $j = \ldots M$, $M \geq 2N$ in the plane, called *centers of rotation* (see below).

Through friction (see below), our manipulation algorithm can alter parts' positions via a special motion primitive: parts can be "told" to rotate a *constant* distance d about any of the C_j's. This is unlike a rigid rotation for which part's displacements would be proportional to their distance from C. We consider d sufficiently small so the primitive rotation is approximately straight and along the tangent, as shown in Figure 1(b). This primitive causes parts to *flow* along a vector field $\phi_C = (\phi_C^x, \phi_C^y)^T$ defined as:

$$\phi_C = \frac{(P - C)^\perp}{||P - C||} \tag{1}$$

Note that at any point P, ϕ_C is unit and perpendicular to $P - C$. It can be shown that the family of these fields is not closed under addition, i.e., $\{\phi_{C_j}\}, j = 1 \ldots M$ will, in general, span an M-dimensional space. Compare this with the linear space of rigid rotations which is closed at dimension 3. (Reznik and Canny, 1998b).

Define a set of scalars $d_j, j = 1 \ldots M$. Define $\Delta P_i'$ as part i's net displacement after it has flowed a distance d_j along ϕ_{C_j}, sequentially, for $j = 1 \ldots M$. With the d_j's small, the concatenation of flows is approximately equal to their sum (i.e., we ignore second- and higher-order terms of the Taylor expansion), and write a linear expression for the $\Delta P_i'$:



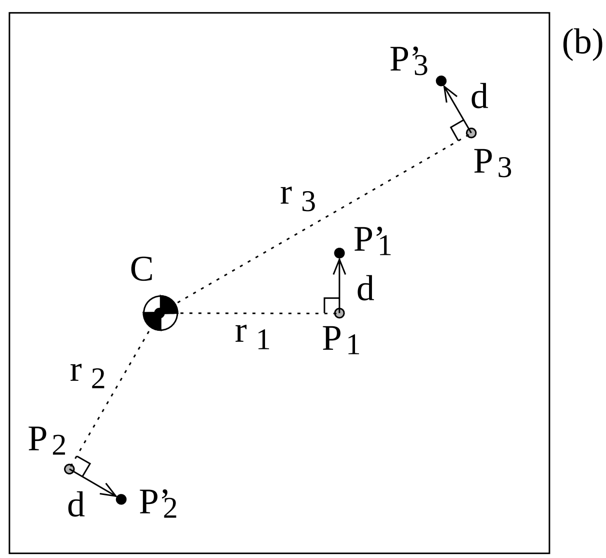


Figure 1 (a) The planar manipulation problem: N parts P_i need to execute a motion ΔP_i, e.g., along trajectories τ_i specified by some high-level task (assembly, sorting, etc.). A set of M points C_j is pre-specified about which the parts can execute a special type of rotation (see below). In the picture, $N = 2$ and $M = 4$. (b) The non-linear rotation primitive used by the manipulation algorithm: all parts P_i flow tangentially with respect to C by a specified d.

$$\Delta P_i' = \sum_{j=1}^{M} d_j \phi_{ij}, \; i = 1 \ldots N \tag{2}$$

where ϕ_{ij} is simply ϕ_{C_j} evaluated at P_i. The above can be expressed succinctly as the following linear system:

$$\Delta P = \Phi . d \tag{3}$$

With:

$$\begin{aligned} \Delta P &= \begin{bmatrix} \Delta P_i^x \\ \text{-----} \\ \Delta P_i^y \end{bmatrix}_{2N \times 1} \\ \Phi &= \begin{bmatrix} \phi_{ij}^x \\ \text{-----} \\ \phi_{ij}^y \end{bmatrix}_{2N \times M} \\ d &= [d_j]_{M \times 1} \end{aligned}$$

The manipulation algorithm can be summarized as follows:

1. Obtain (e.g., from sensors) current part positions P_i
2. Obtain (e.g., from task) the desired part translations ΔP_i
3. Solve Equation 3 for d, i.e., compute $\Phi^{-1}.\Delta P$
4. Rotate parts d_j about C_j, sequentially, for $j = 1 \ldots M$
5. Repeat

After each sequence of rotations, we expect:

$$\Delta P_i' \;\cong\; \Delta P_i, \; i = 1 \ldots N$$

Visualization of the concepts discussed in this Section is provided in Figures 2 and 3.

2.2 TIME-ASYMMETRIC MOTION

Consider a horizontal surface S constrained to move along x. Let the surface's motion be periodic, with velocity profile $\nu_s(t)$, $\nu_s(t) = \nu_s(t+T)$. Consider a part P of mass m lying on S, with velocity ν_p. Assume S's

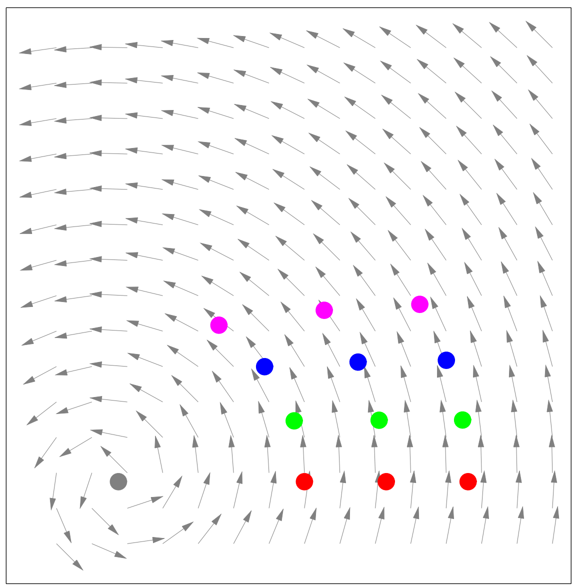


Figure 2 Race Track Experiment: three parts are allowed to "race" simultaneously (i.e., flow) along a non-linear rotation field. Their initial positions are all along a line directly to the right of the center of rotation, which is located at the lower left corner of the field. Four consecutive snapshots of the motion are shown. As expected, the inner parts advance more rapidly than the outer ones.

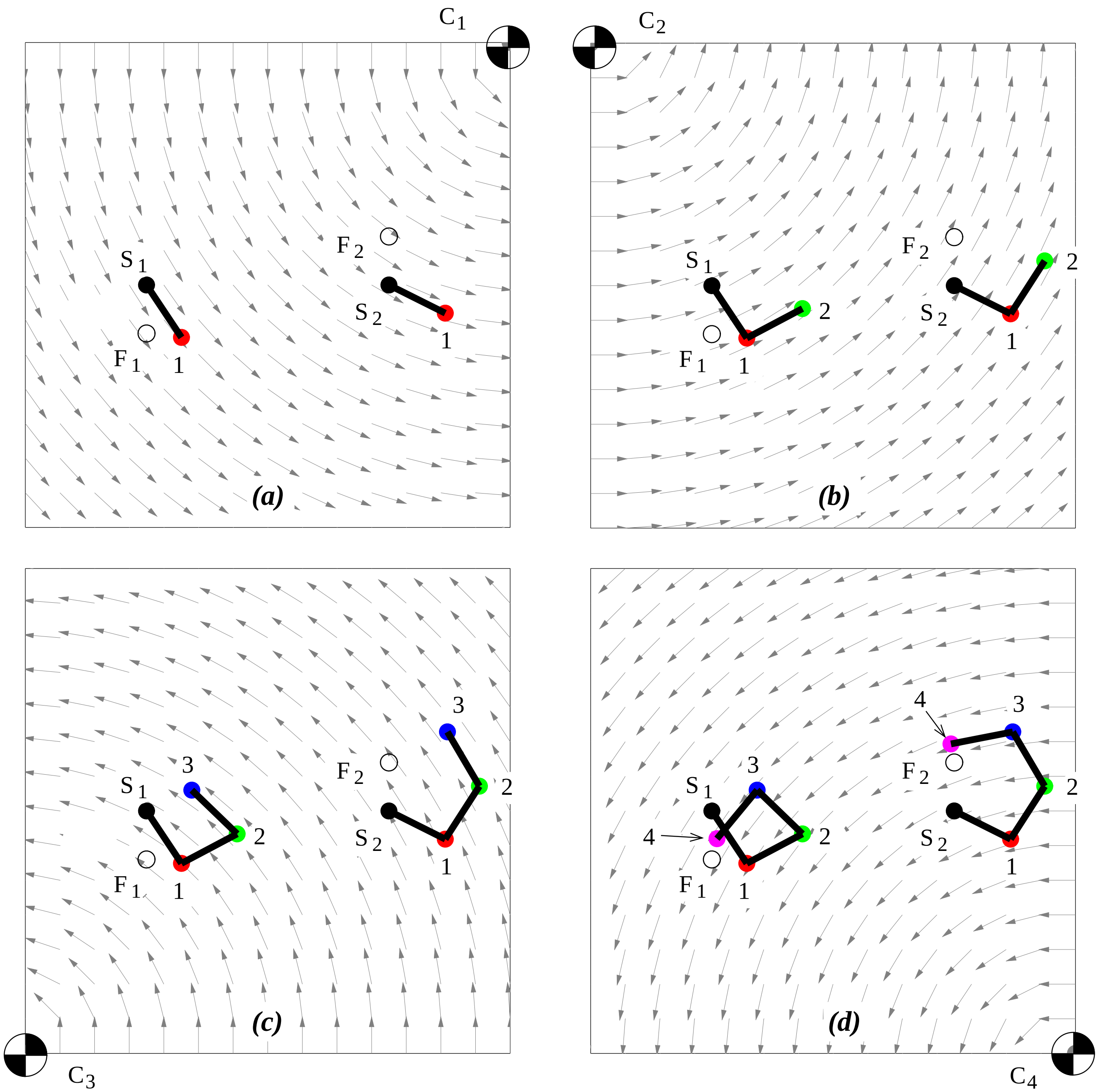


Figure 3 Four snapshots of a 2-part parallel manipulation problem: in (a) two parts are shown lying at starting locations S_1 and S_2; the goal is to move them to final locations F_1 and F_2. Four centers of rotation C_j, $j = 1 \ldots 4$ are specified, each at the corners of a square workspace. The rotations will take place starting with C_1, in counterclockwise order. Snapshots (a) through (d) show the parts' motions incrementally, after each rotation. Intermediate positions are labeled 1 through 4, and connected by a polygonal line. (d) Part's final positions (labeled 4) deviate from the intended destinations F_1 and F_2. This error was made intentionally large by prescribing large desired steps for each part.

acceleration relative to P is high enough so that (i) the part is always sliding on S and (ii) the part's speed ν_p is constant within one cycle, i.e., frictional forces are negligible compared to inertia. The average Coulomb friction $\bar{f}_{1d}$ applied to the part per cycle is given by:

$$\bar{f}_{1d} = \frac{\mu m g}{T} \int_0^T sgn[\nu_s(t) - \nu_p]\ dt \tag{4}$$

Define t^+ as the duration of positive $\nu_s(t) - \nu_p$ within one cycle. It can be shown (Reznik and Canny, 1998a) that:

$$\bar{f}_{1d} = \mu m g(\frac{2t^+}{T} - 1) \tag{5}$$

With $t^+ > T/2$, $\bar{f}_{1d}$ is positive, and the part will feed. In (Reznik and Canny, 1998a), we considered $\nu_s(t)$ of the form:

$$\nu_s(t) = \cos(wt) - \frac{1}{2}\cos(2wt) \tag{6}$$

This particular velocity waveform was picked because it contains only two harmonics and delivers a large $\bar{f}_{1d}$ relative to its peak acceleration (Reznik and Canny, 1998a). In particular, for ν_p small, it can be shown that $\bar{f}_{1d} \cong 0.24\mu m g$, denoted $\bar{f}_0$.

Consider now a surface S which is constrained to *rotate* about a fixed point C. Let $w_s(t)$ represent the periodic angular velocity of S about C. Let $w_s(t)$ be of the form of Equation 6. Then for a part resting ($||\nu_p|| = 0$) at position P on S, the surface will apply $\bar{f}_0$ average force along $(P-C)^{\perp}$. Assuming the Coulomb model of sliding friction applies, over a time Δt, the part will displace $d \propto \Delta t^2$, regardless of its distance from C (in fact, near C tangential accelerations are too small and the part won't slide). The moral is: vibratory rotation can be used to synthesize the "non-linear rotation primitive" described in Section 2.1.

3 PRACTICAL CHALLENGES & SOLUTIONS

3.1 ACTUATION KINEMATICS

One way to accomplish the oscillatory surface motion prescribed in Section 2.2 is to have the surface's three dof's (x, y, and θ) move in phase with velocities as in Equation 6. Note that the instantaneous velocity of a rigid body in the plane is related to its instantaneous center of rotation by the following map (Craig, 1989):

$$\begin{bmatrix} c_x \\ c_y \\ w \end{bmatrix} = \begin{bmatrix} -\dot{y}/\dot{\theta} \\ \dot{x}/\dot{\theta} \\ \dot{\theta} \end{bmatrix} \tag{7}$$

The actuation kinematics illustrated in Figure 4 is designed to apply forces along the table's 3 dof's so that C can be easily chosen. As shown, the plate is positioned at the center of a working area. Four linear actuators are used to apply forces to the each of the plate's sides. Shafts connect the table to the motor, allowing the latter to both push and pull on the former. Shafts are stiff along the actuation direction and compliant perpendicularly.

Let X_1, X_2, Y_1, Y_2 denote the force applied to the table the motor positioned to the left, right, bottom, and top of the table, respectively, as shown in Figure 4(a). At the operating frequencies, overall table displacements will be small, so we can decouple cross-talk between dof's. Namely, the table will tend to rotate clockwise if motors at opposite sides push (or pull) *in tandem*, while the table will tend to translate if a given motor pushes while the one on the opposite side pulls (or vice versa). This can be expressed by the following set of equations which relates applied forces to the resultants along the plate's 3 dof's:

$$\begin{aligned} f_x &= X_1 - X_2 \\ f_y &= Y_1 - Y_2 \\ \tau_\theta &= r[(Y_1 + Y_2) - (X_1 + X_2)] \end{aligned} \tag{8}$$

where r denotes the table's center distance to the actuation point on each side. In Figure 4(b), a more space efficient (and kinematically equivalent) arrangement of motors and table is shown, in which the position (and force signs) of X_2 and Y_2 are changed.

We will model the off-axis shaft compliances as linear damped springs. If input forces are well above resonance, inertial forces dominate both spring and damping forces, so that the velocity along each axis is simply the time integral of the applied external force. So let each motor apply a force of the type:

$$f(t) = \cos(wt) - \sin(2wt) \tag{9}$$

scaled by chosen constants X_1, X_2, Y_1, Y_2. Then, because the map in (8) is linear, the force applied to the table along each of its dofs will also be of this form, so that the resulting integrated velocity will be as desired:

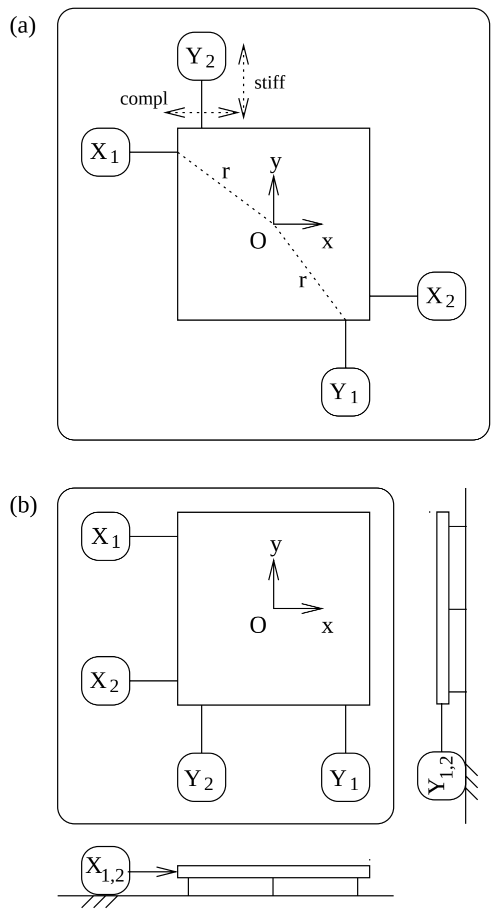


Figure 4 The actuation kinematics: (a) Four linear actuators, labeled $X_i, Y_i, i = 1, 2$, apply force to an individual side of the table through a shaft, attached at a distance r from the center of the table. Shafts are stiff along the driving direction and compliant perpendicularly. In the figure, Y_1's shaft is shown stiff along y and compliant x. (b) A more space-efficient arrangement of motors is shown, along with side views of the table; these show weight-supporting flexible rods under the table (also present in -a-).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} (t) = \begin{bmatrix} \frac{1}{M} f_x \\ \frac{1}{M} f_y \\ \frac{r}{I} \tau_\theta \end{bmatrix} \frac{1}{w} [\sin(wt) + \frac{1}{2} \cos(2wt)] \quad (10)$$

Using Equations 7 and 8, we can choose X_1, X_2, Y_1, Y_2 to *place* C at a desired spot and scale the angular velocity about it.

3.2 SIGNAL GENERATION AND COR VISUALIZATION

We use *voice coils* (BEI Kimco Magnetic Systems, 1999) for each linear actuator. These devices respond with force along the driving axis proportionally to the current flowing through them. We built a dedicated circuit to generate the motor waveforms as defined in Equation 9; a block diagram of the signal-generation hardware is shown in Figure 5. Two microcontrollers (Microchip Catalog, 1996) running appropriate firmware produce a total of four independent analog signals; each signal is power amplified and sent to a motor. A host PC communicates with the board via the parallel port. The microcontroller firmware allows for the flexible calibration of relative phase and amplitude between the 1st and 2nd harmonic components in Equation 9, and for the turning on and off of signals sent to motors, with chosen scaling amplitudes X_1, X_2, Y_1, and Y_2.

Instead of calculating C based on a set of known dynamic parameters (input forces, plate mass and geometry, motors' force constants), we took a reverse-engineering approach. We installed accelerometers (Analog Devices Catalog, 1999) at two opposite corners of the plate (actually glued underneath). Each sensor provides two analog measurements corresponding to the acceleration at two perpendicular axes.

In Appendix 5 we show that by knowing the rigid velocities v_1 and v_2 at two distinct points p_1 and p_2 of a moving plate (e.g., two opposite corners, $p_1 = -p_2$) we can determine the plate's instantaneous center of rotation and angular velocity:

$$|w| = \frac{||v_2 - v_1||}{2||r_1||} \quad (11)$$

$$C = \frac{(v_1 + v_2)^\perp}{2w} \quad (12)$$

There are two problems with the above: (1) the sensors recover acceleration, and not velocity; (2) sensor data is noisy. To address (1) we simply state that under sinusoidal excitation, the RMS velocity will be proportional to the RMS acceleration, independently for the 1st and 2nd harmonic components. Speaking of RMS, this suggests a solution for (2), i.e., rather than computing C and w based on instantaneous acceleration readings we do that based on average amplitudes over a large number of sampled cycles.

Figure 5 shows the 11-bit A/D converter used to sample the four accelerometer signals simultaneously. This is currently done at a rate of 5 KHz. Samples are passed to the PC via the parallel port in real-time. One such sequence of samples is shown in Figure 6(a). Since the force frequency w is known, the least-squares amplitude and phase of the signal are recovered by dotting the sensor samples with the four orthogonal functions $\cos(wt)$, $\sin(wt)$, $\cos(2wt)$ and $\sin(2wt)$ (essentially a DFT (Haykin, 1989)), yielding coefficients c_1, s_1, c_2, s_2, i.e., we fit the following "model" acceleration $a(t)$ to our data:

$$\begin{aligned} a(t) &= c_1 \cos(wt) + s_1 \sin(wt) \\ &+ c_2 \cos(2wt) + s_2 \sin(2wt) \end{aligned} \tag{13}$$

which we express succinctly as $a(t) = [c_1, s_1, c_2, s_2]$. A well-registered least-squares fit to the data in Figure 6(a) is shown in Figure 6(b). To visualize the least squares-fit velocity waveform, we simply integrate Equation 13, obtaining an identical waveform expressed as:

$$v(t) = \frac{1}{w}[-s1, c_1, -\frac{s_2}{2}, \frac{c_2}{2}] \tag{14}$$

This is used to generate the velocity waveform shown in Figure 13. Real-time visualization of $v(t)$ allows the user to fine tune relative phase and amplitude parameters between first and second harmonic to compensate for frequency dependent phase and amplitude response (ideally, phase is flat and amplitude roll-off is as $1/w$ when $w >> w_0$, however a bit of pre-compensation is always needed).

The least-squares fit recover unsigned amplitudes to the sinusoidal accelerations along each of the four probed axes. To assign directions to each of these vectors we need to consider the phase relationship between the first and second harmonic of each acceleration signal. For a 1d acceleration profile of the form $\cos(t) + \cos(2t + \phi)$, the average force will be positive iff $\phi \in (0, \pi)$ (Reznik and Canny, 1998a). In Appendix 1, we shown that this corresponds to the following expression in terms of the four free parameters in Equation 13:

$$sgn[\text{force}] = sgn[2 s_1 c_1 c_2 + s_2(s_1^2 - c_1^2)] \tag{15}$$

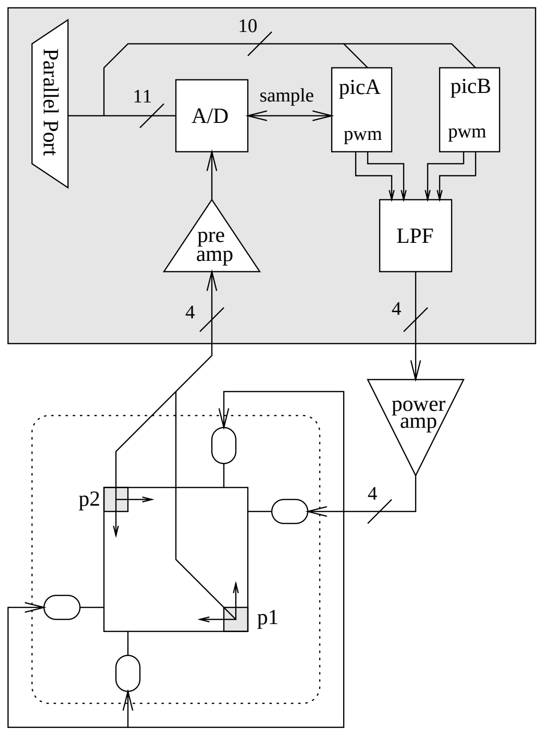
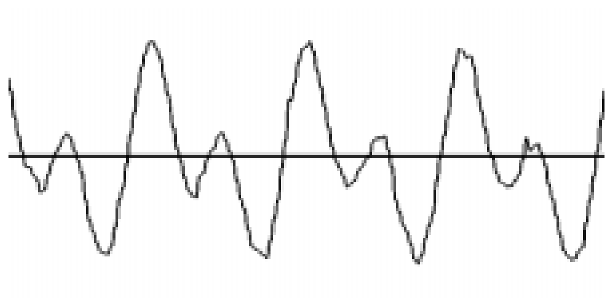
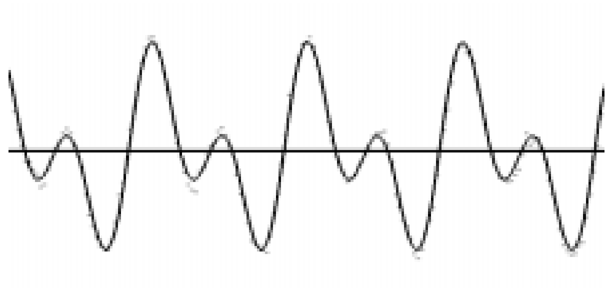


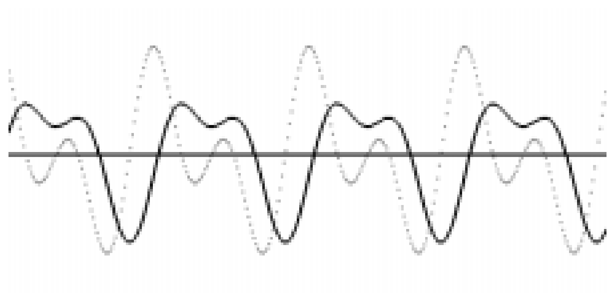
Figure 5 Block diagram of the signal generation and acceleration acquisition hardware: two microcontrollers (picA and picB) generate four independent PWM signals. These are low-pass filtered and power-amplified, and then applied to each motor. Two 2-axis accelerometers are glued under opposite corners of the table. The four acceleration readings are pre-amplified and input to a 4-channel, 11-bit A/D, whose sampling is controlled by one of the PICs. The PC can send commands and/or read samples from the A/D via a parallel port interface.



(a)



(b)



(c)

Figure 6 The waveform fitting process: (a) Samples coming from one sensed axis; (b) Least-squares fit (shown solid) and original samples (shown dotted); (c) Least-squares fit (shown dotted) and closed-form integral, i.e., the fitted velocity signal (shown solid).

3.3 SYNTHESIZING SCALED DISPLACEMENT FIELDS

Consider a surface rotating periodically about point C with angular velocity $w(t)$ as in Equation 6. Consider N parts P_i lying on S at rest. With enough motor power the amplitude of $w(t)$ can be made high enough so that parts velocities ν_p are negligible compared to the peak tangential velocity at the part's locations, call it ν_{max}.Under sliding Coulomb friction, parts will experience and average tangential force per cycle of $\bar{f}_{1d} = \bar{f}_0$, as mentioned above.

To simplify control, we make the following key assumptions: (i) A desired displacement field will be generated by a finite-duration pulse. (ii) At the beginning of the pulse all parts will have zero velocity. To ensure this, each pulse will be preceded by a sufficiently long rest phase. (iii) By keeping all parts' velocities negligible with respect to the peak of $w(t)\times r_{min}$, where $r_{min} = \min_i\{P_i - C\}$, all parts will accelerate by the exact same amount, and that amount will be linearly proportional to the pulse's length.

To avoid impulse-response ringing, we will initiate (resp. terminate) the pulse with smooth attack (resp. decay) phases, of identical duration. The pulse's middle part, called its *sustain phase* will be of a much higher duration S. These concepts are illustrated in Figure 8. The final desired displacement d for all parts P_i will be proportional to S^2, i.e.:

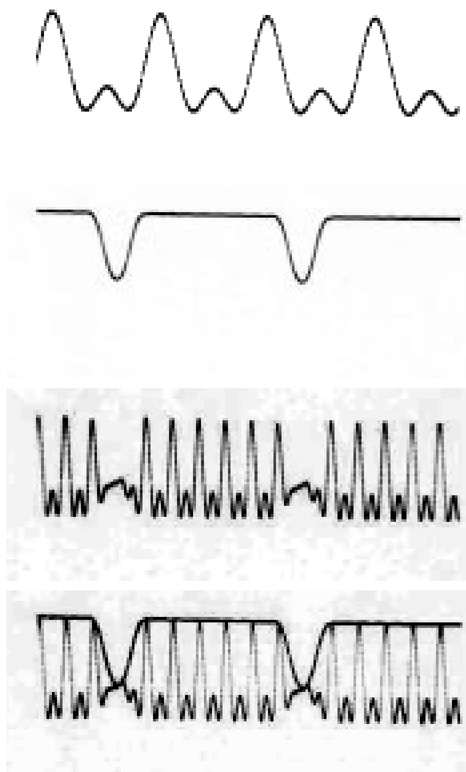
$$S \propto \sqrt{d}$$

The signal-generation hardware allows for the easy tuning of attack/decay and sustain durations shaping of the output waveform. Oscilloscope photographs showing actual output are reproduced in Figure 7.

3.4 TRACKING PARTS

A camera is placed a few feet above the table pointing downward at the latter's center. The ground is black, the table is white and the parts are pennies painted black.

The first step is to determine the table's rotation and translation relative to the image. This is a one-time operation, done prior to the task, given that the table itself moves negligibly when it vibrates. We compute the table's edge map using standard procedures (Russell and Norvig, 1995). Each edge in the image is then hashed by its distance to the image's center and angle onto a 2d Hough-vote array (Russell and Norvig, 1995). Edges making up the table's four sides will cluster at four locations on the Hough-array. Each of the Hough peaks gives rise



(a)

(b)

(c)

(d)

Figure 7 Waveform shaping: (a) four cycles of the original velocity waveform $\sin(t) + \cos(2t)/2$; (b) the attack/sustain/decay envelope; (c) the shaped waveform, i.e., -a- multiplied by -b-; (d) the envelope superimposed on the shaped wave, showing registration. These pictures were taken from an actual oscilloscope (the sweeping rate for -a is four times faster than for the rest).

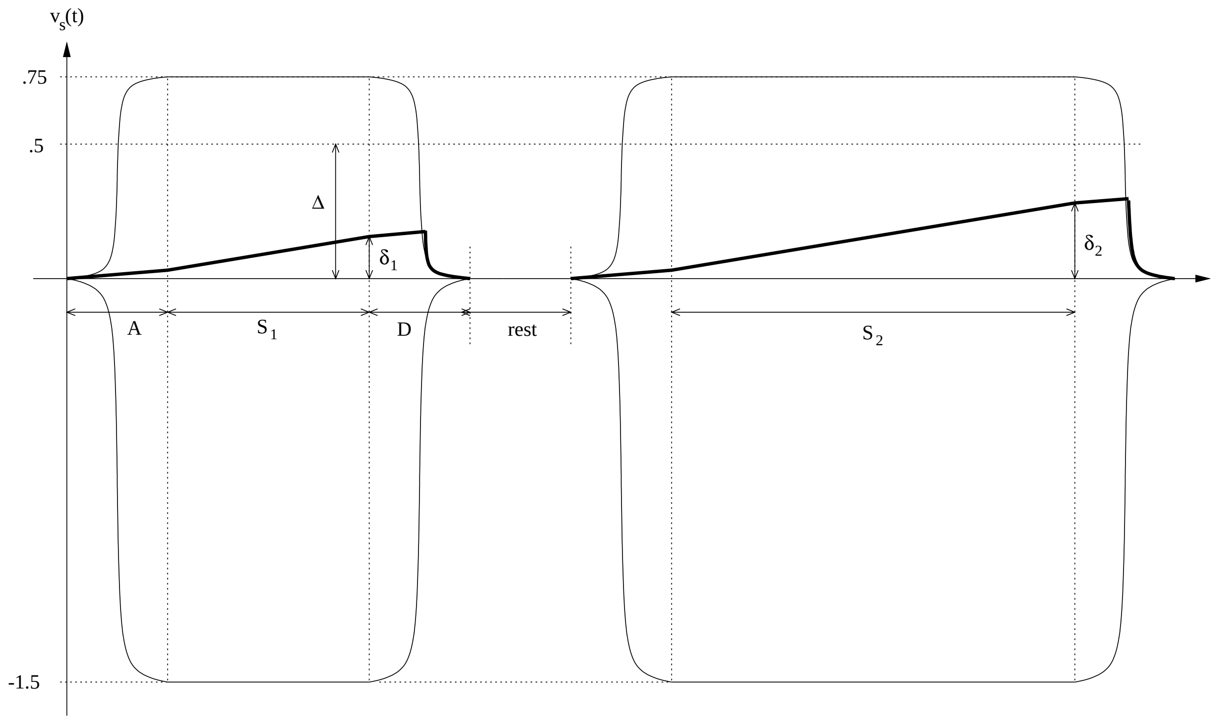


Figure 8 Emulating rotation intensity through pulse duration: two shaped pulses are shown. The pulse is represented by the *outline* of a normalized tangential velocity, covering $[-1.5, 0.75]$ along y (i.e., the range of $\cos(t) - \cos(2t)/2$). Each pulse contains 10s-100s of cycles of the basic driving waveform (not drawn). Each pulse starts (resp. finishes) with a smooth attack (resp. decay) phase lasting A (resp. $D = A$) seconds. The first (resp. second) pulse *sustain* duration is S_1 (resp. S_2). For convenient visualization, $S_2 = 2S_1$. The part's velocity is shown in plotted with a thicker line. Pulses are preceded and followed by a *rest* phase which ensures part velocity is null at the beginning of each pulse. Though not drawn to scale, assume the attack/decay phases are very short compared to the sustain; in this fashion, part speed will increase steadily so that at the end of the pulse, its value (shown as δ_1 and δ_2) is *proportional* to S_1, S_2, i.e., part displacement will be proportional to δ_i^2. To ensure this, the dynamic parameters must be tuned so that δ_i is negligible compared to the peak of the envelope.

to a line. Sorting these lines by angle and then intersecting consecutive line pairs, we obtain the 4 corners of the table and its coordinate frame.

The second step is to locate the coins' initial locations. Having previously determined the table's sides (and their lengths as they appear in the image) we compute a circular kernel (a solid disk) with a pixel-radius proportional to the penny/table-side ratio known a priori. This kernel is convolved with all points in the image interior to the table outline, computed above. The convolved image will contain peaks corresponding to the center of each coin.

Determining coins' initial locations is done once prior to the task. The actual *tracking* of coins is a much cheaper operation. Once they start moving, once must simply convolve the aforementioned disk-shaped kernel over a 1 or 2 pixel neighborhood of a part's current location; the peak in the convolved neighborhood determines the coin's new position.

3.5 THE CONTROL LOOP

The sequence of steps suggested in Section 2.1 is slightly modified to incorporate the practical solutions described in this Section:

- Use vision to obtain parts' coordinates P_i
- From task trajectories, specify new motion subgoals ΔP_i
- Given a set of M feasible COR's, solve for rotation scaling d_j, $j = 1 \ldots M$.
- Actuate table so it rotates $\sqrt{d_j}$ seconds about C_j (using a shaped pulse), sequentially, for $j = 1 \ldots M$.
- Compare with desired steps (report error), repeat

4 EXPERIMENTS

4.1 COR STEERING AND CALIBRATION

Given the actuation kinematics in Figure 4(a), Equations 10 and 7 give rise to the following proportionality laws:

$$\begin{aligned} w &\propto (Y_1 + Y_2) - (X_1 + X_2) \\ c_x &\propto (Y_2 - Y_1)/w \\ c_y &\propto (X_1 - X_2)/w \end{aligned} \tag{16}$$

We used the signal generation hardware to test the table's vibration under six distinct choices for amplitudes $X_i, Y_i, i = 1, 2$, as shown in Table 1. As it is apparent, in all combinations the w control $(Y_1 + Y_2) - (X_1 + X_2)$ is kept constant. By varying the other components, the idea is to "steer" the COR away from its original position in fixed

	X_1	X_2	Y_1	Y_2	$Y_1 - Y_2$	$X_1 - X_2$	c_x	c_y	Δc_x	Δc_y
(a)	-128	-128	128	128	0	0	$-.22$	.02		
(b)	-256	0	128	128	-256	0	2.92	.04	3.14	.02
(c)	-256	0	0	256	-256	256	2.72	2.52	$-.20$	2.48
(d)	-128	-128	0	256	0	256	$-.18$	2.66	-2.90	.14
(e)	-128	-128	-128	384	0	512	$-.14$	5.34	.02	2.68
(f)	0	-256	-128	384	256	512	-3.25	5.54	-3.11	.2

Table 1 Motor amplitudes $X_i, Y_i, i = 1, 2$ as they were passed to the hardware waveform generator. Real-time accelerometer output was used to compute the coordinates c_x and c_y of the associated center of rotation, displayed in inches with respect to the table's center (the table is an 8"x8" square). Notice that the last two CORs lie outside the table's surface. The $\Delta c_{x,y}$ show the COR displacement with respect to its location given the controls in the preceding row. As seen, the device is fairly "balanced" on both axis, responding linearly to changes in the control as predicted by Equation 16.

steps along the following axes: $+x$, $+y$, $-x$, $+y$, and $-x$. A program was written which performs real-time acquisition of acceleration data and the simultaneous computation/visualization of the COR's. Figure 9 shows the CORs placement for each of the amplitude combinations sent to the motors; as shown, the COR does get placed at the intended locations. The actual coordinates for C calculated in real-time from the accelerometers' outputs are shown in the last two columns of Table 1.

With this machinery, one can tweak waveform amplitudes input to the four motors until the COR is steered to a convenient location. Repeating this process for enough distinct locations and recording the required amplitudes gives rise to a "COR library" which can then be used by our parallel manipulation algorithm.

4.2 ONE-PART TRAJECTORY-FOLLOWING

In order to test the integrity of key parts of the system, namely, the image-processing/part tracking, the interfacing with the signal generation hardware, and the mechanical functionality of our prototype, we designed a simple automated, visually-servoed task involving a single part (a penny painted black). The experimental setup is shown in Figure 10.

(a) The penny is placed at a random location on the table. (b) The image processing system locates it. (c) The penny is brought to the exact center of the table via translations along x and y. (d) The penny will traverse clockwise and indefinitely, the four branches of an imaginary "plus" sign laid over the table. It starts out traversing in the $-x$ direction

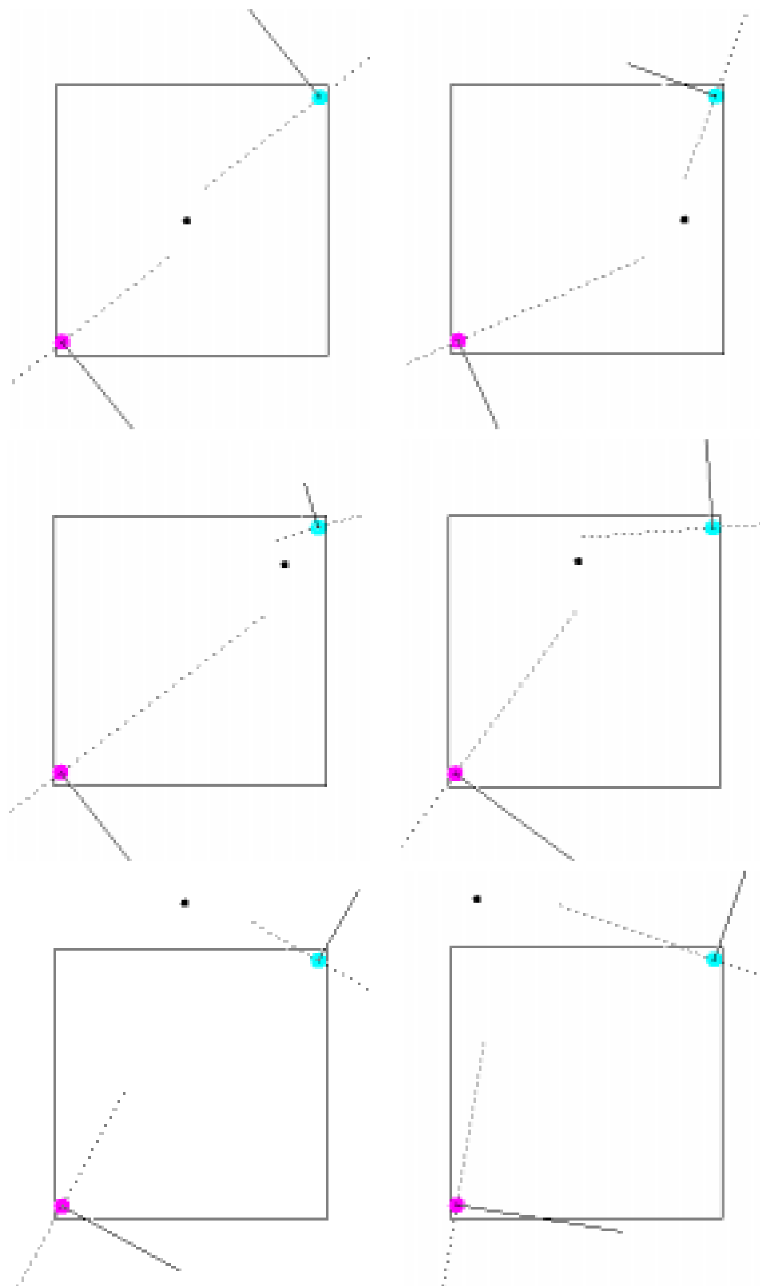


Figure 9 Steering the COR with the 6 amplitude combinations shown in Table 1. The table is drawn as an outline; the accelerometers are drawn centered at their actual locations near the lower-left and upper-right corners of the table. The actual magnitude of acceleration measured by each two-axis accelerometer is shown along with the perpendicular (the COR is supposed to fall at the intersection of these). The actual computed COR is shown as a black dot. Snapshots should be read left-to-right, top-to-bottom; in the first four, the COR lies inside the table's surface; in the remaining two, it falls outside.

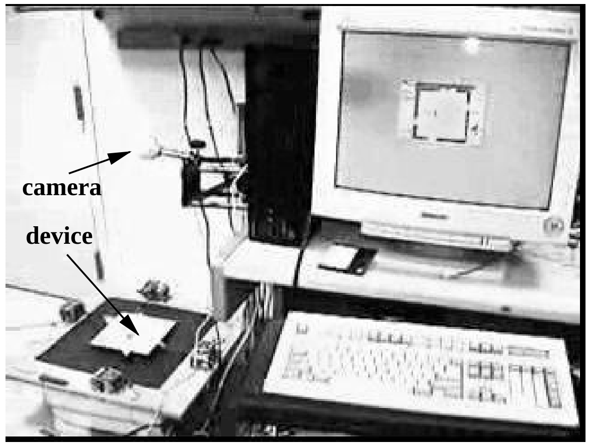


Figure 10 Experimental Setup for the 1-coin experiment: the computer, shaker table, and camera are visible.

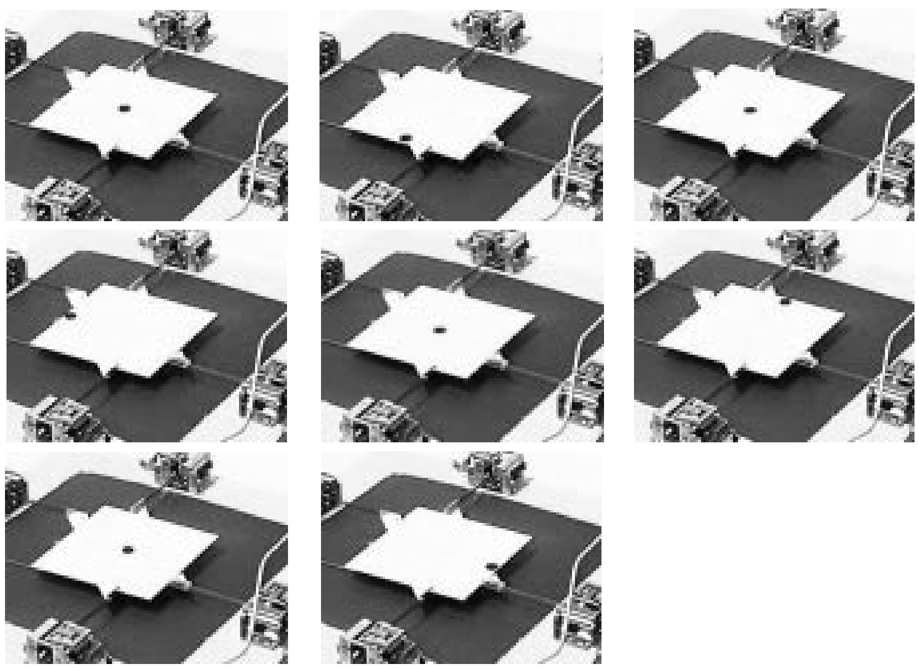


Figure 11 Eight consecutive snapshots (to be read left-to-right, top-to-bottom) of a simple visually-servoed trajectory-following task involving a single part (black penny). The plate is vibrated along x and y to steer the coin along the branches of an imaginary "plus" sign centered on the board. It does so in clockwise order, starting with the $-y$ branch. For each branch, the coin advances from the table's center to its edge at which point visual-servoing commands the motors to reverse feeding direction.

until it hits the table's edge, at which point it switches directions and returns to the center. After that, the $+y$ branch is explored, and so forth. For this simple task, the system performed robustly and consistently. Eight consecutive snapshots of this experiment, are shown in Figure 11.

5 CONCLUSION & FUTURE WORK

We have described a minimalist approach to parallel part manipulation which is dual to the standard array-based device in distributed manipulation in the sense that a small (indeed a single) number of actuators is used to manipulate a large number of parts. This is achieved through a more complex manipulation scheme. Additionally, our algorithm requires that parts' positions be known, precluding sensorless manipulation, a direction which is of much interest in array-based distributed manipulators.

Implementation of the device is underway; important hurdles already cleared include the design, mechanical tuning, and control of the actuation kinematics, the ability to flexibly generate signals to the actuators, visualization and calibration of centers-of-rotation, and part localization through image processing.

Appendix: COR Calculation

Assume the table is a rigid square with center O. Assume the instantaneous velocities v_1 and v_2 at points r_1 and r_2 are known. These quantities are illustrated in Figure 1. The goal is to compute the table's instantaneous center of rotation c and the associated instantaneous angular velocity w measured about c. We can write:

$$\begin{aligned} v_1 &= w(r_1 - c)^{\perp} && \text{(A.1)} \\ v_2 &= w(r_2 - c)^{\perp} && \text{(A.2)} \end{aligned}$$

Taking the difference (A.1)-(A.2) eliminates c, i.e.:

$$v_2 - v1 = w(r_2 - r_1)^{\perp} = -2wr_1 \qquad \text{(A.3)}$$

Which implies:

$$\begin{aligned} |w| &= \frac{||v_2 - v_1||}{2||r_1||} && \text{(A.4)} \\ sgn(w) &= sgn[(v_2 - v_1) \times r_1] \end{aligned}$$

Taking the sum (A.1)+(A.2) yields:

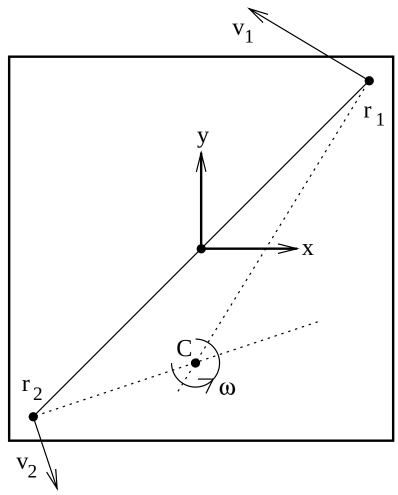


Figure 1 The shaker table is shown with two accelerometers placed at r_1 and r_2, with $r_2 = -r_1$. The instantaneous velocities at these points are v_1 and v_2, respectively. Lines L_1 and L_2 pass through r_1 and r_2, and are perpendicular to v_1 and v_2, respectively. The instantaneous center of rotation C and angular velocity w are also shown. Notice that C will lie at $L_1 \cap L_2$.

$$v_1 + v_2 = w(r_1 + r_2)^\perp - 2wc^\perp$$

Since $r_1 + r2$ vanishes in the above, we proceed with:

$$\begin{aligned} v_1 + v_2 &= -2wc^\perp \\ (v_1 + v_2)^\perp &= 2wc \\ c &= \frac{(v_1 + v_2)^\perp}{2w} \end{aligned} \tag{A.5}$$

With w computed as in (A.4). Equations A.4 and A.5 are then the final results. An alternative method to compute c is to find the intersection of infinite lines L_1, L_2 passing thru r_1, r_2, which are perpendicular to v_1, v_2, respectively (see Figure 1). This method is inconvenient since the intersection is ill-defined with nearly parallel v_1 and v_2.

Appendix: COR Calculation

Assume plate's acceleration relative to part is of the form:

$$a_p(t) = \cos(t) + 2b\cos(2t + \phi) \tag{B.1}$$

In (Reznik and Canny, 1998a) we show that under the above plate motion, the part's equilibrium velocity is:

$$\nu_{eq} = b\sin(\phi), \ \ |b| < 1/2 \tag{B.2}$$

Though a closed-form expression was not derived for the average force applied to the part per cycle (assuming zero part velocity) in terms of b and ϕ, Equation B.2 implies that the *sign* of the average force is given by $sgn[b\sin(\phi)]$. An alternative representation for Equation B.1 is:

$$\begin{aligned} a_p(t) &= c_1\cos(t) + s_1\sin(t) + \\ &\quad c_2\cos(2t) + s_2\sin(2t) && \text{(B.3)} \\ &= m_1\cos(t-\alpha_1) + m_2\cos(2t-\alpha_2) && \text{(B.4)} \\ (m_i, \alpha_i) &= (\sqrt{c_i^2 + s_i^2}\ ,\ \tan^{-1}\frac{s_i}{c_i}),\ i = 1, 2 \end{aligned}$$

Let $t' = t - \alpha_1$, then Equation B.4 can be rewritten as:

$$a_p(t) = m_1[\cos(t') + \frac{m_2}{m_1}\cos(2t' + 2\alpha_1 - \alpha_2)] \tag{B.5}$$

Modulo the m_1 scaling factor, Equation B.5 is in the form of Equation B.1, with $\phi = 2\alpha_1 - \alpha_2$ and $b = \frac{m_2}{2m_1} > 0$. So the force will be positive when $\sin(2\alpha_1 - \alpha_2) > 0$, i.e.:

$$2\alpha_1 - \alpha_2 \in (0, \pi) \tag{B.6}$$

Define complex numbers $z_i = c_i + js_i, i = 1, 2$. Then $2\alpha_1$ and α_2 are the angles under $z_1^2 = c_1^2 - s_1^2 + 2jc_1s_1$ and z_2, respectively. So Equation B.6 is equivalent to stating $z_1^2 \times z_2 > 0$, or equivalently:

$$2s_1c_1c_2 + s_2(s_1^2 - c_1^2) > 0$$

References

Analog Devices Catalog (1999). Winter 1999 short form designer's guide.

BEI Kimco Magnetic Systems (1999). Voice coil actuators, an applications guide. Information booklet.

Böhringer, K., Bhatt, V., and Goldberg, K. (1995). Sensorless manipulation using transverse vibrations of a plate. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan.

Böhringer, K., Goldberg, K., Cohn, M., Howe, R., and A. (1998). Parallel microassembly using electrostatic force fields. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium.

Boothroyd, G. (1991). *Assembly automation and product design*. Marcel Dekker, Inc., New York, NY.

Canny, J. and Goldberg, K. (1994). RISC for industrial robots: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, San Diego, CA.

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

Frei, P. and Wiesendanger, M. (1999). Simultaneous planar transport of multiple objects on individual trajectories using friction forces. In *Workshop on Distributed Manipulation, IEEE International Conference on Robotics and Automation*, Detroit, MI.

Haykin, S. (1989). *An introduction to analog and digital communications*. John Wiley & Sons, Inc., New York, NY.

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

Kavraki, L. (1997). Part orientation with programmable vector fields: Two stable equilibria for most parts. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM.

Luntz, J., Messner, W., and Choset, H. (1998). Velocity field design for the modular distributed manipulator system (MDMS). In Agarwal, P., Kavraki, L., and Mason, M., editors, *3rd Workshop on Algorithmic Foundations of Robotics: Robotics: the algorithmic perspective*. A. K. Peters, Natick, MA.

Microchip Catalog (1996). Pic16/17 microcontroller data book.

Quaid, A. (1999). A miniature mobile parts feeder: operating principles and simulation results. In *IEEE International Conference on Robotics and Automation*, Detroit, MI.

Reznik, D. and Canny, J. (1998a). The coulomb pump: A novel parts feeding method using a horizontally-vibrating surface. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium.

Reznik, D. and Canny, J. (1998b). A flat rigid plate is a universal planar manipulator. In *IEEE International Conference on Robotics and Automation*, Leuven, Belgium.

Reznik, D. and Canny, J. (1998c). Universal part manipulation in the plane with a single horizontally-vibrating plate. In Agarwal, P., Kavraki, L., and Mason, M., editors, *3rd Workshop on Algorithmic Foundations of Robotics: Robotics: the algorithmic perspective*. A. K. Peters, Natick, MA.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence, a Modern Approach*. Prentice Hall, New Jersey, NJ.

Swanson, P., Buurridge, R., and Koditchek, D. (1995). Global asymptotic stability of a passive juggler: A parts feeding strategy. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan.