The Universal Planar Manipulator

by

Dan S. Reznik

B.Sc. (University of Kansas at Lawrence) 1990
M.Sc. (University of Wisconsin at Madison) 1993

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Prof. John F. Canny, Chair
Prof. Ken Goldberg
Prof. Kris Pister

Fall 2000

The dissertation of Dan S. Reznik is approved:

Chair Date

Date

Date

University of California at Berkeley

Fall 2000

The Universal Planar Manipulator

by

Dan S. Reznik

Abstract

The Universal Planar Manipulator

by

Dan S. Reznik

Doctor of Philosophy in Computer Science

University of California at Berkeley

Prof. John F. Canny, Chair

This dissertation is about a novel technology for planar part manipulation. A device has been developed, called the Universal Planar Manipulator (UPM), based on a single actuator: a rigid, horizontal plate which vibrates in its own plane (3 degrees of freedom). Generic parts (bottles, tools, etc.) placed on the UPM's plate move as a result of friction. Our main contribution has been to develop control algorithms for the UPM which allow it to manipulate several parts in parallel. The UPM's actuation simplicity combined with its open workspace (no grippers) renders it appealing to existing industrial applications and novel interactive devices such as toys and active desks.

Prof. John F. Canny
Dissertation Committee Chair

To my parents Hania and Alberto Reznik and to my grandmother Hedwig Friedländer.

Contents

List of Figures

Acknowledgements

My first thanks goes to my advisor, Professor John F. Canny, for his guidance, creative impetus (read: unstoppable genius), pragmatism, and friendship. I thank Professors Kris Pister, Ron Fearing, and Ken Goldberg, for their advice and support as members of my Ph.D. committee. I thank my friends Professor Sanjay Tiwari and Christian Laugier for their support at a difficult moment and for encouraging me to pursue my Ph.D. no matter what. I thank my officemates Eric Paulos, Francesca Barrientos, Yan Zhuang, Scott Klemmer, and Danyel Fisher for their technical help and companionship during those late nights in Soda Hall. I thank Emil Moshkovich for his help and advice in the project. I thank Lydia Kavraki, Karl Böhringer, and Howie Choset for their collaboration and friendship. I thank Winnie Wang, Katrhyn Crabtree, Tammy White, and Teresita Sarracino for their help with departmental matters. I thank Luís Cascão-Pereira, Heather Yaros, Paulo Ney de Souza, Henrique Bursztyn, Léo Tenenblatt, Jean-Philippe Boubli, Manish Chakrabarti, and Chi-Chao Chang (and others not appearing) for their long-lasting friendship and support. Finally, I thank my family in Brazil for their unconditional support in this 13-year long journey as a foreign student in the U.S.

Support for this research was provided in part by NSF grants FD93-19412, CDA-9726389 (Challenges in CISE: Planning and Control for Massively Parallel Manipulation)

Chapter 1

Introduction

In this dissertation we describe a novel device called the Universal Planar Manipulator (UPM), whose function is to manipulate parts in the plane. Part manipulation is the controlled displacement – translation, rotation – of generic objects (tools, bottles, coins, etc.) in a bounded planar workspace. The simplest type of manipulation is *parts feeding*: one or more objects must displace as a group somewhere, e.g., in transferring luggage from plane to passengers in an airport, moving pharmaceutical products on a production line, etc. The simplest parts feeder is the conveyor belt which uses static friction of objects with a continuously rolling belt to produce forward motion. Part feeding is important in applications where inspection of objects is necessary, e.g., rejecting defective products. Another important feature commonly associated with parts feeding is part presentation: a part needs to be transferred from point A to point B and be presented at point B at a known pose, so as to prepare it to the next manufacturing stage. This is akin to a relay runner who must hand his baton to the next runner at an expected orientation. This leads to the notion of manipulation.

Manipulation is as a selective type of parts feeding: specific objects need to execute specific motions, e.g., a defective object must be moved out of a group currently being conveyed, or oriented to a desired pose. A typical solution for manipulation is an anthropomorphic, ”pick-and-place” one: a robotic end effector (e.g., a gripper) picks up an object at a starting location, transports it to a destination, and places it there. Disadvantages of this method include the fact that robotic arms are expensive, hard to design, and hard to control. If several objects need to be manipulated simultaneously – an operation called *parallel manipulation* – motion control must take into account potential collisions between the various arm/end-effector devices.

The shortcomings of pick-and-place manipulation have motivated research on non-prehensile (gripperless) manipulation devices, grouped in a field called Distributed Manipulation [7]. Devices have been proposed based on 100s to 1000s of actuators tiled over a supporting surface. Each actuator produces a local force of some kind, e.g., magnetic, mechanic, aerodynamic, etc. An actuator array can thus display an arbitrary planar force field, and thus achieve arbitrary parallel manipulation of parts sitting on its surface. This design eliminates robotic arm clutter simplifying control and inspection tasks (parts are directly visible). However, it has the disadvantage of a large number of moving parts, anyone of which is a candidate for failure.

In this thesis, we have looked at a complementary question: can a device be designed which retains the active-surface design of an actuator array and which contains dramatically less moving actuators? We have developed a device, called the Universal Planar Manipulator (UPM), for general planar manipulation in the plane which is based on a single moving actuator: a rigid, horizontal plate with only three degrees of freedom. Our main contribution has been to show that such a minimalist device is sufficient to manipulate several parts sitting on its surface in parallel. We describe motion control algorithms for the UPM's horizontal plate which achieve this goal.

Due to its simplicity (and reduced costs) the UPM could penetrate existing industrial manipulation applications and inspire novel devices such as toys and interactive desks.

1.1 Related Work

1.1.1 Dynamic Simulation as a Design Tool

Our thesis research started with the analysis of parts-feeding performance for an existing actuator-array design. A dynamic simulation tool called Impulse [29] was used to study the dependence of feeding throughput on actuator geometry. The device in question, called the M-Chip, was developed by Böhringer et al. [9, 8]. It contains approximately 10,000 micro-fabricated actuators, called resonators, tiled over a few square-cm of silicon substrate. Resonators vibrate in see-saw fashion at several kHz; local forces are imparted on resting objects through repeated impacts coming from the oscillating resonators. No attempt had yet been made to simulate part-array interactions quantitatively at the detailed mechanical level, owing mainly to the lack of efficient, accurate dynamic simulation tools. The Impulse tool [29] overcomes these problems by providing an accurate model for collision detection and resolution, which is particularly well suited for the types of rigid vertex-face interactions likely to occur between the part and the resonators.

Previous work done using Impulse to collect statistical data to characterize a complex mechanical process was done in the context of estimating pose statistics for polyhedral parts dropped from random orientations on a flat surface [30]. From a design optimization standpoint, the work closest to ours is that of Berkowitz and Canny [4, 5] who used Impulse to optimize the design of a passive parts-orienting device. From a modeling standpoint, Boothroyd [11] has worked on abstracting the performance of vibratory bowl feeders with respect to part mass, part geometry, friction, and oscillating frequency of the device. A related and emerging line of work involves simulation-based optimization of part shapes targeted for thermodynamic self- assembly [13, 22].

The end-result of this exploration was (i) we determined geometric parameters which optimized parts feeding and (ii) we found a bug on the original design which was solved by a new resonator control motion scheme.

1.1.2 Vibrations-Based Parts Feeding

The M-Chip is based on nearly vertical impacts of resonators against supported parts. However the feeding motion is typically horizontal. That suggests a more appropriate design where actuator motion is mostly horizontal. This presupposes a sliding frictional interaction between horizontally vibrating actuator and parts. The nature of sliding friction is such that forward feeding forces result if an asymmetry in the feeder's horizontal motion is imposed. Zesch et al. have developed a micro-positioning device which steps over a stationary flat plate by contracting/extending its body [45]. Similar to inch-worm motion, this device exploits both stiction and sliding frictional modes. Reznik and Canny [36] have shown that asymmetry in the compression-decompression phases of part-actuator impact is the primary cause for forward part feeding for the M-Chip. Reznik and Canny have consider the problem of a motion waveform with fewest sinusoidal components [39]. Experimental results utilizing this type of motion have been reported in [34]. Other examples of array-based feeders which exploit asymmetry include [20, 43].

1.1.3 Vibrations-Based Planar Manipulation

The big challenge then becomes to upgrade a simple vibrations-based feeding motion to a full (and possibly parallel) type of planar manipulation. The bowl feeder [11] is the canonical example of a vibrations-based manipulation device. Though stark in its simplicity, it is not a programmable device: its function is tied to part shape and its own internal track design. The APOS feeder [11] uses vibrations of a grooved surface to orient objects as they fall into the grooves. Hayward et al. [18] 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. [6] document experiments with a transversely-vibrating plate in an attempt to automatically gather particles at vibration nodes. In [44], closed horizontal motions of a flat plate are used to "ratchet" a part of known shape to a desired final orientation. Frei and Wiesendanger [15] have designed a distributed manipulation device based on an array of 1-dof plates (actuated vertically). The array moves as a whole along a planar circle; by careful phase control they can achieve pixel-wise force control with just one dof per actuator. Using these designs as inspiration, we took on the challenge to develop a novel device, which with a single moving actuator retains full parallel, arbitrary planar manipulation dexterity. This is explained in the following chapters.

1.2 Thesis Organization

This thesis is organized as follows: in Chapter 2 we present our work on Dynamic Simulation as a tool for designing parts feeders. In Chapter 3 we present a novel design for a one-dimensional parts feeder based on a horizontal plate. In Chapter 4 we extend the vibrations-based feeder design to a device called the Universal Planar Manipulator (UPM) capable of handling generic parallel planar manipulation. In Chapter 5 we discuss implementation challenges and solutions for the UPM. In Chapter 6 we describe a motion primitive, called the "jet", which renders manipulation with the UPM robust and practical. Conclusions and future work are presented in Chapter 7.

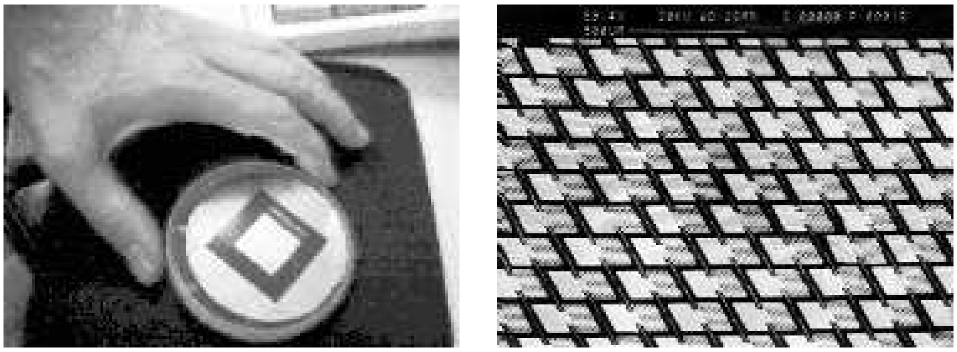
Chapter 2

Dynamic Simulation as a Design Tool for a Microactuator Array

A recent trend in robotics has been the design and fabrication of micro-electromechanical (MEMs) active surfaces intended for part manipulation at the sub-mm scale. Böhringer et al. [8] have fabricated one such device called the *M-Chip* ("M" stands for manipulation), shown in Figure 2.1(a). This device consists of an array of approximately 10,000 micro-actuators, called *resonators*, tiled over a few square-cm of silicon substrate. Resonators are rectangular slabs of silicon supported a few μm above the substrate by torsional *rods*. Electrostatic actuation cause resonators to oscillate about the rods at a few kHz. One end of the resonator is equipped with a *ridge* of several vertical *poles*. The asymmetric design generates anisotropic impact forces when a small *part* is placed over the array, inducing a motion bias towards a specific direction [8].

In one version [1] of the device, called the *unidirectional array*, resonators are all oriented

[1] Another version of the device, called the *squeeze array*, consists of two opposing unidirectional arrays connected to each other along a center line. This device has been proposed for sensorless part orientation [9].



(a) (b)

Figure 2.1: (a) A photo of Böhringer's hand holding his M-Chip, containing approximately 10,000 resonators. (b) Electron micrograph of a portion of the array showing the interleaved tiling of resonators.

towards a single direction and tiled in interleaved fashion, as shown by the electron micrograph in Figure 2.1(b). This configuration acts as a microscale parts feeder for light objects dropped on its surface – these are transferred from one end of the array to the other at a constant speed ν called the *feed rate*.

One of the main difficulties in testing and re-engineering the M-Chip is obtaining meaningful experimental data given the small dimensions and the sheer number of on-chip moving actuators. The fact that the dynamic state of the device and/or part is practically inaccessible makes it difficult for one to identify design changes which could lead to better performance. This type of problem is analogous to that faced by a semiconductors designer who needs detailed measurements on the operation of a new type of device with the intent of improving its performance. Such measurements will be often too time consuming and/or noisy, so an alternative is simulation tools such as FEM or SPICE. Here we propose a similar solution. Our contribution is to utilize a powerful dynamic simulation tool called *Impulse* [29] to extract detailed, noise-free dynamic information from the array over a variety of experiments. In designing dynamic simulations for the M-Chip, we have as principal goals:

1. Design verification and debugging: does the device correctly feed parts placed on its surface and at which rate? If not, can problems be identified and solutions proposed which are successful at least in simulation?

2. Performance optimization: for a working prototype, are there changes in the design which could lead to better performance?

3. Device modeling and critique: can the chaotic part-array interactions be reduced to a simplified model which captures the bulk behavior of the device's dynamics?

In doing (1) we discovered a bug in the existing design which causes the part to jam against a row of resonators, leading to an interruption on the feeding process. The designers confirmed this problem from experimental tests, reinforcing the reliability of our simulated models. We proposed and tested a solution, called *row-strobing*, which eliminates jamming for all simulated experiments. Still in this category, we identified a phenomenon called *double bouncing*: low speed secondary resonator-part collisions occurring shortly after a higher speed collision cause frictional losses, reducing the feed rate. This effect was eliminated with a reduction of the resonator's duty cycle, resulting in a 30% gain in feed rate.

For (2), we leveraged on the simulator's flexibility to perform a "blind search" over a two-parameter family of resonator shapes and part masses; the combination leading to the best feed rate was subsequently found.

To address (3), we collected statistical information on the part's dynamic parameters. We found that the part's main motion modes are (i) forward feeding and (ii) vertical bouncing. In particular, (ii) is corresponds the motion of a mass supported by a vertical spring, with measurable spring constant. We found that most of the energy of the system is “wasted” on this non-feeding hopping mode. By applying external forces during the feeding process, we also found that the friction experienced by the part is of *viscous* type. Our own parallel work on the dynamics of vibratory part feeders [40] has provided some answers as to why this is so.

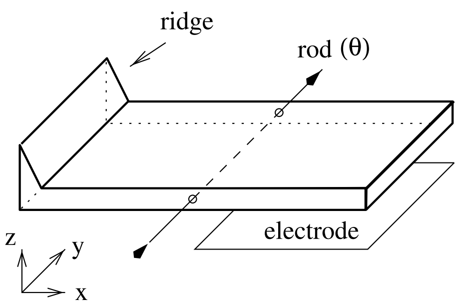
The remainder of this Chapter is organized as follows: in Section 2.1 the simulation setup is explained. In Section 2.2 we describe how both the jamming and double bounce problems were eliminated, and also performance studies which identify an optimal resonator shape. In Section 2.3 we develop an abstract model for the device and study the energetics of part motion. Section 2.4 concludes with a summary and directions for future work.

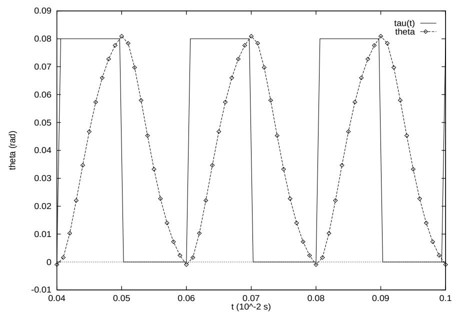
2.1 Simulation Setup

In modeling the M-Chip with Impulse, the focus was to preserve original nominal parameters and dimensions as closely as possible. We used the original resonator mass, geometry, and oscillation frequency as given to us by the designers. We started out by creating a 280x180x5 μm geometric model for the resonator, shown in Figure 2.2(a). To simplify collision detection, we modeled the set of poles installed on one extreme of the resonator collectively as a single 5 μ-high ridge. The resonator body[2] was modeled as a rectangular slab. In the actual device, resonator motion is induced by applying a voltage between the resonator and an electrode underneath it, as shown in Figure 2.2(a). Electrostatic actuation was replaced by (i) modeling the supporting rods as a single spring-loaded revolute joint, and by (ii) a torque control-law acting at that joint. The joint’s spring

[2]Fabricated resonators posses a grating of holes, but these have no effect on collisions with the part.

and damping constants were chosen according to the torsional elasticity of silicon. The torque wave applied to the joint is a 5 kHz positive square wave, whose amplitude was chosen[3] to produce positive oscillations of approximately 5^o (0.08 rad). Figure 2.2(b) shows the driving torque law superimposed on θ, the free oscillation angle, which is roughly a 5 kHz positive sinewave (the resonator acts as a mechanical low pass filter which attenuates the higher frequency harmonics of the driving square-wave).





(a) (b)

Figure 2.2: (a) The resonator is modeled as a rectangular slab with a ridge at one end. The supporting rods are modeled as a spring-loaded, damped revolute joint. The electrostatic actuation (provided by the electrode) is replaced by a torque control law at the joint. (b) The torque applied to the resonator is a 5 kHz positive square wave. The torque (not shown in scale) is superimposed with $\theta(t)$, the resonator angle. The resulting oscillation is a positive 5 kHz sinewave.

The next modeling step was to assemble the resonators in interleaved fashion, as in Figure 2.1(b). Though the actual device contains over 10,000 resonators, simulating this many moving parts is impractical with Impulse. However, by considering an 8x5 sub-array of resonators, and a rectangular array-aligned part with silicon's density, we are can remain physically consistent with the full size experiment. The final Impulse model shown as a 3D rendering is depicted in Figure 2.3. Notice that the part's footprint covers approximately 10 resonators, while in actuality it would cover

[3] 1 kHz and 2^o are the values nominally used by the designers – w.l.o.g., we chose 5 kHz and 5^o for better numerical stability with Impulse.

100s.

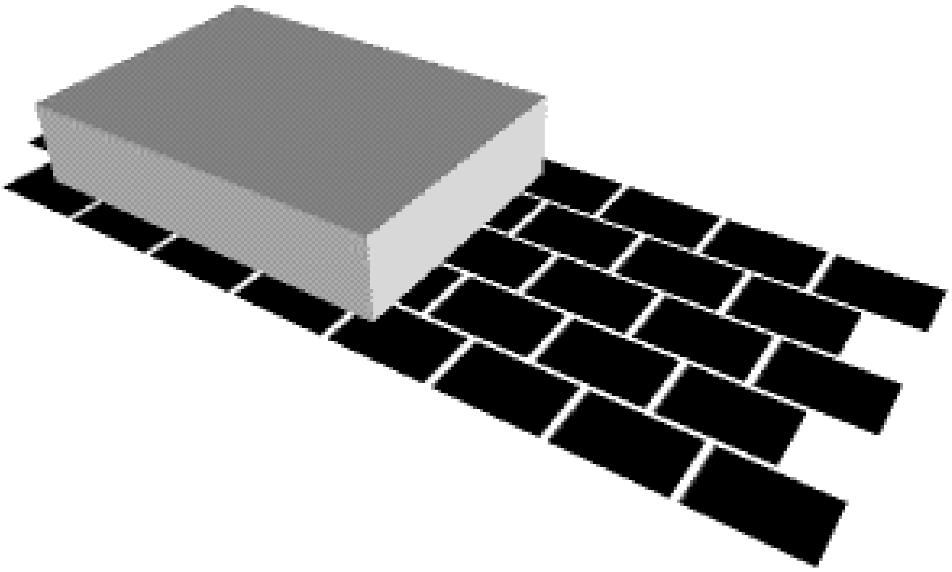


Figure 2.3: 3D rendering of the complete model showing a 1.15x0.4x0.25 μm, 550 μg part resting at its initial position over an 8x5 resonator array.

The Impulse simulator resolves collisions between rigid bodies based on a sophisticated discrete-event impulse model. Two global parameters need to be set: the friction coefficient μ, and the restitution coefficient ϵ. These were set to the physically reasonable (and numerically-stable) values of 1.0 and 0.5, respectively. Experiments to evaluate how these parameters affect array performance have not yet been tried, though some theoretical results have been derived for vibratory feeders [11].

2.2 Design Optimization

The first simulated experiment performed with the array was to drive the resonators and simply drop the part on it. Figure 2.3 shows the part at its starting position for this experiment.

2.2.1 Part jamming

Once dropped on the array, the part is propelled forward at a constant speed of about 0.8 mm/sec, a value in close match with experimental results performed by the designers. A problem quickly discovered was that the part would jam (i.e., stop its forward motion) as soon as it encountered a new row of resonators, as shown in Figure 2.4(a). Jamming occurs since the part's vertical hopping is not high enough to allow it to skip over the ridges of the next row of resonators. As the part is driven up, the next row of resonators is also doing so, and the part bounces back.

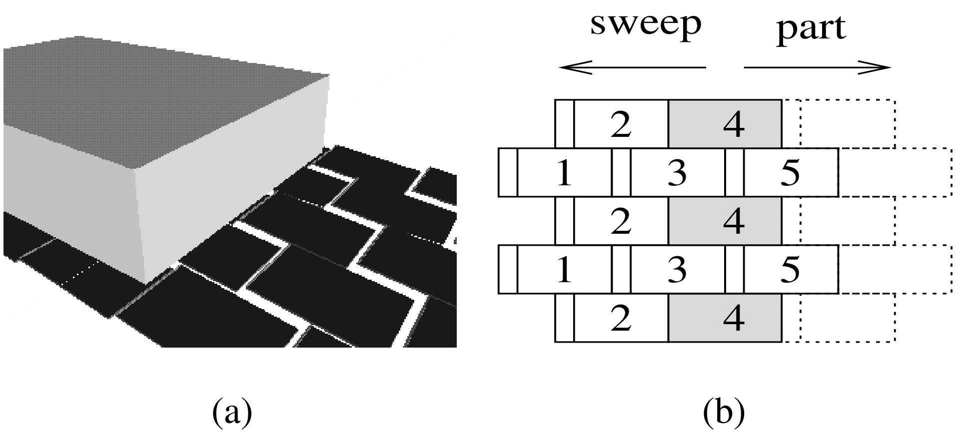


Figure 2.4: The jamming problem: in (a) the part is shown unable to skip over the ridges of the encountered row of resonators. In (b) the row-strobing method is illustrated. Consecutive rows in the array are labeled from 1-8 in the direction of the part's motion (only the first 5 rows are shown). The process involves selectively turning off consecutive rows of resonators for a few oscillation cycles, in the direction opposite to the part's motion.

The first attempted solution to eliminate jamming was to drive consecutive resonator rows at different phase offsets, so that ridges in the blocking row would be going down while other resonators would be driving the part up. This idea did not work since the out-of-phase impacts occurring under the part cause lower part hops which in turn aggravate the problem of skipping a set of blocking ridges. A solution which proved successful is called *row-strobing*, illustrated in

Figure 2.4(b). The idea is to propagate a wave of off resonators in the direction opposite to the part's motion. Label all rows from 1 to N along the array's feeding direction. The first row to be turned off is row N. That row is left in the off state during γ complete oscillation cycles, at which point it is turned back on. The process continues with row $N-1$ being off for γ cycles, and so on until row 1 is reached. At this point, the wave wraps around and the process restarts at row N. Since part position is not known, any resonator row is potentially causing jamming, thus the need for a sweeping wave (this could be alleviated with sensing). To ensure that the jamming row is located faster, the wave is propagated against the feeding direction so the relative speed between the wave and the part is higher. Choosing too small a γ may not give the part enough time to be pushed over the blocking ridges, however the average jam clearing time is proportional to γ. We found that $\gamma = 50$ oscillation cycles gave the best results. In the case of a very long array (as in the real device) the blocking row can be found faster by propagating several off wavefronts separated by a constant number λ of resonator rows. If λ is too small, too many off rows will lie under the part at any given time, reducing the feed rate considerably. In our case we set $\lambda = 4$ rows, i.e., in our 8x5 array there will be, at any given time, two off rows moving in the $-x$ direction. Since the part is about 4 resonators long, the part receives $1/4$ less impacts at any given time.

As shown in Figure 2.5, the row-strobing method results in a motion of the part characterized by periods of constant forward feed rate (of approximately 0.8 mm/sec) interleaved with short jamming phases, where the feed rate is null.

2.2.2 Double bouncing

The simulator treats collisions as discrete events; each collision is followed by a calculation which computes an impulse force applied at the collision point and in opposite direction to the

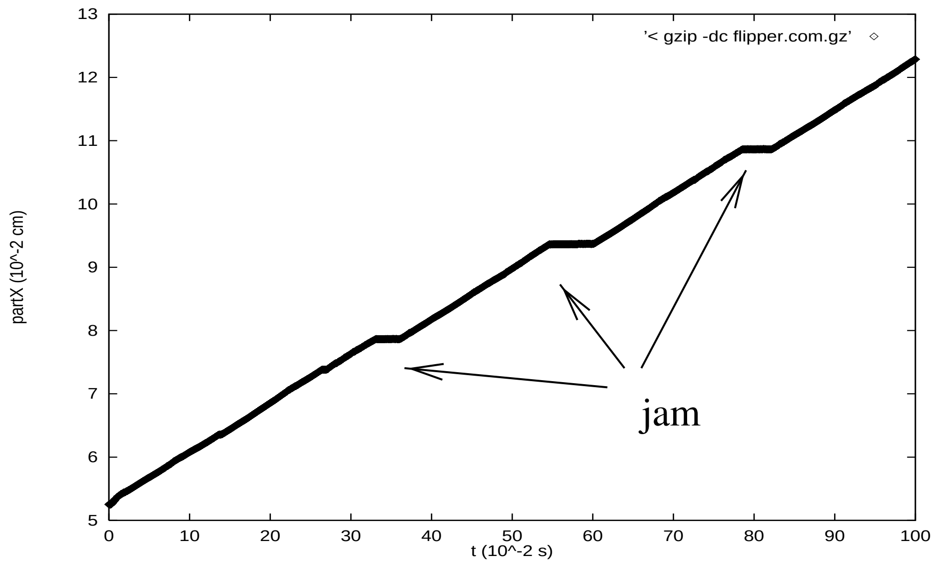


Figure 2.5: Plot of the part's forward displacement as a function of time. As shown, the part's motion alternates between constant forward motion and short periods of jamming.

bodies colliding. We looked at the stream of collisions between the part and the resonator located on the 2nd row and 3rd column of the array during the first non-jamming period of Figure 2.5, namely, for $0.05 < t < 0.2$ sec. For every such collision we recorded the associated (i) resonator angle θ, (ii) angular speed $\dot{\theta}$, and (iii) F_x, the x component (i.e., along the feeding direction) of the impulse force calculated by the simulator. Figure 2.6 shows a scatter plot of the $(\theta, \dot{\theta})$ pairs gathered. These pairs cluster into two separate clouds pointed to by the arrow labeled *loaded*. As expected, one of the clouds (signaled by the *free* arrow) follows quite closely the θ vs. $\dot{\theta}$ relation for the resonator's free oscillation – this is an ellipse since the free oscillation is roughly sinusoidal. However, the second cluster of points is anomalous.

What phenomenon could be generating the lower cloud of points? That cluster indicates that many collisions are occurring at an angle θ with a much slower $\dot{\theta}$ than that of free oscillations. We found that these collisions were being caused by secondary bounces of the resonator on the part

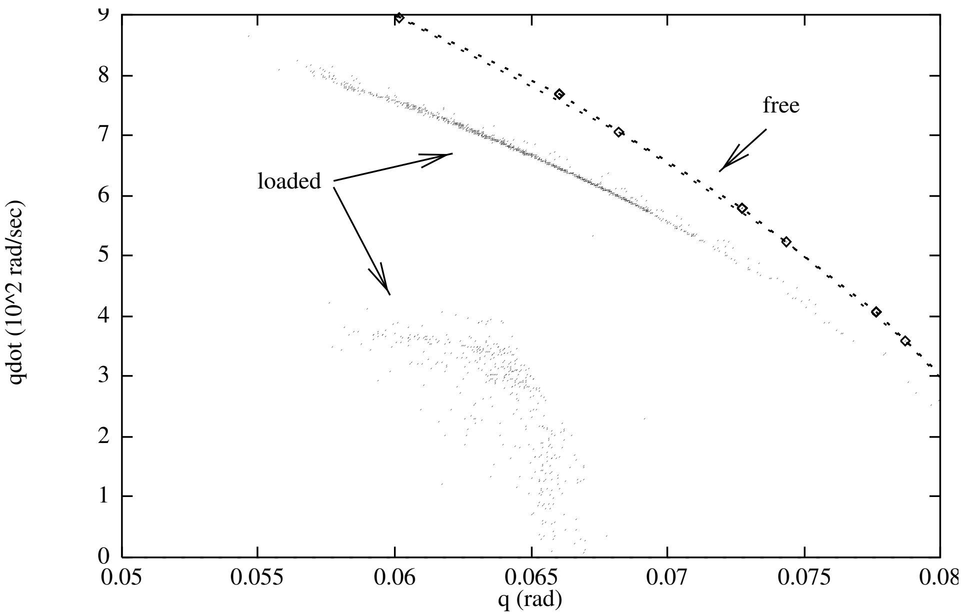


Figure 2.6: Scatter plot correlating resonator angle θ (x-axis) w/ the angular speed $\dot{\theta}$ (y axis). The *free* arrow shows this relationship for free oscillation – roughly an ellipsoid. The *loaded* arrow points to $(\theta, \dot{\theta})$ pairs generated by individual collisions when the part is over a chosen resonator. Some of the loaded points follow the free oscillation curve, while others lie in an anomalous region of low values of $\dot{\theta}$.

occurring shortly after a normal 5 kHz bounce. To show that, we correlated θ, $\dot{\theta}$, and F_x with the collision's *inter-arrival time*, which measures the time elapsed between the current collision and the one last occurring in the simulation. This is shown in Figure 2.7(a,b,c). Though one expects collisions with a single resonator to be spaced by $1/5$ kHz $= 0.2$ ms (i.e., the part receives one impact per resonator cycle), the graphs show that collisions cluster over .02, .18, and .2 ms interarrivals. The collisions occurring at .02 ms after the normal .2 ms ones are *double bounces*; after the first collision, the ridge is still being driven upward by the torque control law, causing the resonator to ricochet one or more times against the part. After the double bounce, the resonator will tend to re-synchronize with the driving square wave, so that the next collision occurs within $.18$ sec: $0.18 + 0.02 = 0.2$ ms. The (a) plot shows that the .02 collisions occur at lower values of θ than the normal .2 ms ones. This indicates that a downward motion of the part favors the double bounces. The (b) plot shows that double bounces occur at much slower $\dot{\theta}$ than normal bounces, implying that the former transmit less impact energy to the part than the latter. The (c) plot shows that most double bounces are associated with an impulse pointing in the negative feeding direction, i.e., they act as brakes!

Double bounces were eliminated by reducing the *duty cycle* of the square wave torque driving the resonators from 50% to 36%. This shuts off the driving torque law just before a double bounce is likely to occur (i.e., .02 ms after the average time normal collisions occur). The results of this change are shown in Figure 2.7(d,e,f), which side by side with the previous plots illustrate how the cloud of .02 and .18 ms collisions coalesce into a single cluster over 0.2 ms inter-arrivals. These graphs also show that the new impulses occur in average at a higher value of $\dot{\theta}$, i.e., they transfer more momentum at every collision. This simple reduction in duty cycle increases the feed rate from 0.8 to 1.0 mm/sec, i.e., a 30% improvement. As an interesting note, the points in Figure 2.7(f) are split evenly in the positive and negative F_x range, indicating that the total force applied to the part over the period considered is zero – the part is feeding forward at a constant speed.

2.2.3 Optimizing the resonator's shape

The feed rate ν was measured against three parameters: (i) the part's mass M, (ii) the ridge's distance L from the resonator's midpoint, and (iii) the ridge height H. The last two parameters are illustrated in Figure 2.8.

The plots in Figure 2.9 show the results of these experiments. Plots (a,b,c) show, respectively, results from the mass, ridge distance, and ridge height experiments. The x-axis labels the parameter being varied; the feed rate is plotted along y. The collision angle θ, angular velocity $\dot{\theta}$, and the height of the part's center of mass PartZ (averaged over an entire feeding task) are superimposed over the feed rate; their numeric values have been omitted for the sake of clarity.

Figure 2.9(a)'s optimal ν at $M = 55\ \mu$g is reached since (i) θ is monotonically decreasing (as it gets heavier, the part "sinks" into the array, tracking PartZ) and (ii) $\dot{\theta}$ tends to level off (both these effects contribute to less momentum transferred on the $+x$ direction).

For Figure 2.9(b), θ increases with L by a simple lever-like effect (ridge gets closer to revolute joint). This effect explains a monotonically decreasing PartZ, since as the ridge gets closer to the center, it transfers less momentum at every impact. These two opposing trends cause the $\dot{\theta}$ curve to go through a maximum at approximately $L = -125\ \mu$m, which in turn causes ν to reach its highest value shortly after $\dot{\theta}$'s maximum.

The first obvious fact in Figure 2.9(c) is that PartZ moves linearly up with an increase in the ridge height. For the lower region of H values, θ remains constant while $\dot{\theta}$ increases monotonically, and so does ν. At $H = 16\ \mu$m both θ and $\dot{\theta}$ decrease sharply, also decreasing ν. At present we haven't been able to justify what are the geometric/dynamic reasons for this effect.

2.3 Device Modeling

2.3.1 Array ballistics

The plot in Figure 2.9(a) shows a linear relationship between M and the PartZ, i.e., the array acts as a linear spring over which the part bounces. This linear relationship yields a spring constant $k = 294$ kdyn/cm. For $M = 55\ \mu$g, this mass-spring system resonates at 370 Hz. We computed the Discrete Fourier Transform (DFT) of PartZ for $0.05 < t < 0.2$ sec, and found two major frequency components: one at 360 and one at 5 kHz, corresponding, respectively, to the spring-like oscillations, and the normal resonator impacts. The DFT also revealed that the 360 Hz component was 5 times larger than the 5 kHz one, suggesting that too much energy is being spent in the upward vibrational mode (more on this below).

2.3.2 Friction modeling

To understand the type of friction experienced by the part as it "slides" on the array, we performed the following experiment. A 55 μg part is dropped on the array as in Figure 2.3. Enough time is waited so that the part achieves its equilibrium feed rate of 1.1 mm/sec. At that point ($t = 0.05$ sec), an external force F_{ext} in the $-x$ direction is applied to the part's center of mass. The experiment consists in observing the resulting feed rate ν' for different values of F_{ext}. Figure 2.10 shows the part's x position vs. time, before and after F_{ext} is applied – this graph reveals a linear relationship between the new feed rate and F_{ext}, i.e., the array acts as a *viscous* medium, akin to a fluid! This effect motivated us to look at this problem in depth, but for lack of space we direct the reader to another publication [40]. In a nutshell, it is caused by the velocity-independent Coulomb frictional force combined with the temporally asymmetric stream of part-resonator impacts.

The viscous model prescribes a frictional force proportional to (i) the part's weight, and (ii) the difference between part velocity and normal feed rate, i.e., $F_{fric} = \mu M g(\nu' - \nu)$, where μ is the coefficient of static friction. This relation was used to compute values for μ for various externally applied forces, noting that at equilibrium $F_{fric} = F_{ext}$. The results are tabulated in Table 2.1. As shown, the model assumption explain the data quite well, with μ nearly independent of F_{ext} and approximately equal to Impulse's global coefficient of friction.

F_{ext} (dyn)	ν' (mm/sec)	μ
0	1.1	N/A
.025	.6	1.16
.05	.06	.99
.06	0	0.93
.075	-.36	1.02
.1	-.88	.99

DOF	E_{avg} ($10^{-}5$ dyn cm)	Stdev
LinX	1.78	.71
LinY	.1	.15
LinZ	1.38	1.86
AngX	.65	.86
AngY	.63	1.28
AngZ	.0081	.045
PotZ	50.1	14.3

Table 2.1: Left: Friction coefficient (μ) computed using a viscous friction model, for various externally applied forces. Right: Energies and standard deviations present in each of the part's independent DOF's, over $0.05 < t < 0.2$ sec.

2.3.3 Energetics of part motion

We measured the average energy present in each of the part's degrees of freedom: three translational kinetic energies along X, Y, and Z, denoted LinX, LinY, and LinZ; three angular kinetic energies about the X, Y, and Z axes, denoted AngX, AngY, and AngZ; the part's potential energy PotZ (with $g = 981$ cm/sec^2) with respect to its resting height. The values for these energies averaged over an $0.05 < t < 0.2$ sec (and the corresponding standard deviations) are shown in Table 2.1.

The above data shows that this type of array consumes a disproportionate amount of energy to keep the part at an average potential energy, rather than for forward motion. This suggests that an array with longitudinal rather than vertical actuators would be more energy efficient (e.g., see [43] for a novel type of design based on thermally-actuated cilia). Notice also that a small portion of the energy is *equipartitioned* between AngX and AngY, implying that part motion along these DOF's is pretty much chaotic. The low value of AngZ shows that the array (as expected) is unable to accelerate the part about the Z axis.

2.3.4 An abstract model

From the above studies, the following simplified model of part/array dynamics can be derived: the array acts as a springy conveyor belt over which the part hops. The average part height (how much it sinks into the array), denoted d, is a function of the part's mass M and the supporting spring's stiffness k. Friction with the conveyor belt is viscous with coefficient μ. The belt feeds at a rate ν, also a function of d. This model is depicted in Figure 2.11.

2.4 Summary

In this chapter we show how dynamic simulation can be used as an effective tool in the characterization an further design optimization of an existing MEMs device called the M-Chip. We are planning to incorporate into the M-Chip's design the various improvements suggested by this work, and find out whether the gains in performance are indeed possible.

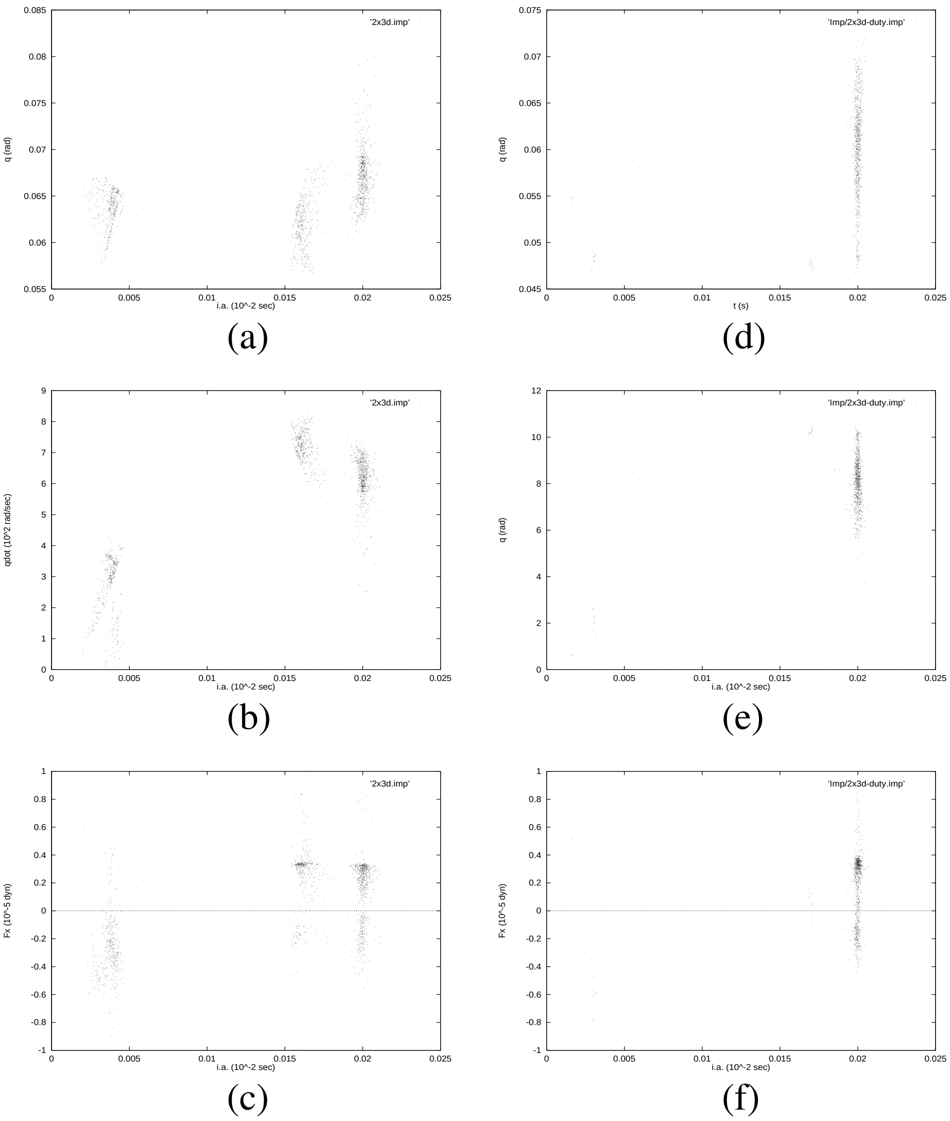


Figure 2.7: Graphs (a,b,c) show θ, $\dot{\theta}$, and F_x (plotted along y) clustering over three specific collision inter-arrivals (plotted along x): .02,.18, and .2 ms. Graphs (d,e,f) show the re-clustering of the same quantities, this time solely over 0.2 ms inter-arrivals, when a shorter duty cycle is used.

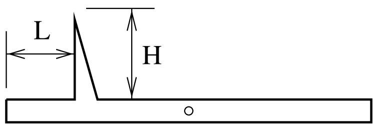


Figure 2.8: The shape of the resonator is parameterized according to ridge distance L, and ridge height H.

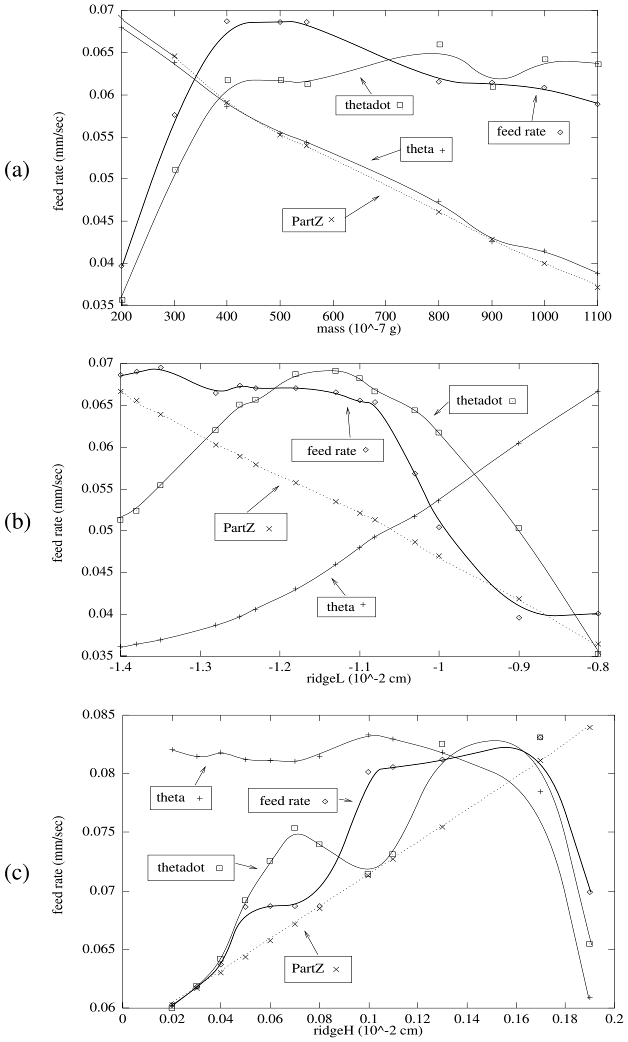


Figure 2.9: Results of the performance studies. Each graph's y-axis presents simultaneously ν, θ, $\dot{\theta}$, and PartZ (averaged over a feeding experiment) plotted against the parameter being varied. For (a), (b), and (c), the parameters are, respectively, M, L, and H. The y-axis is labeled in feed rate units, numeric values for the other quantities have been omitted for simplicity.

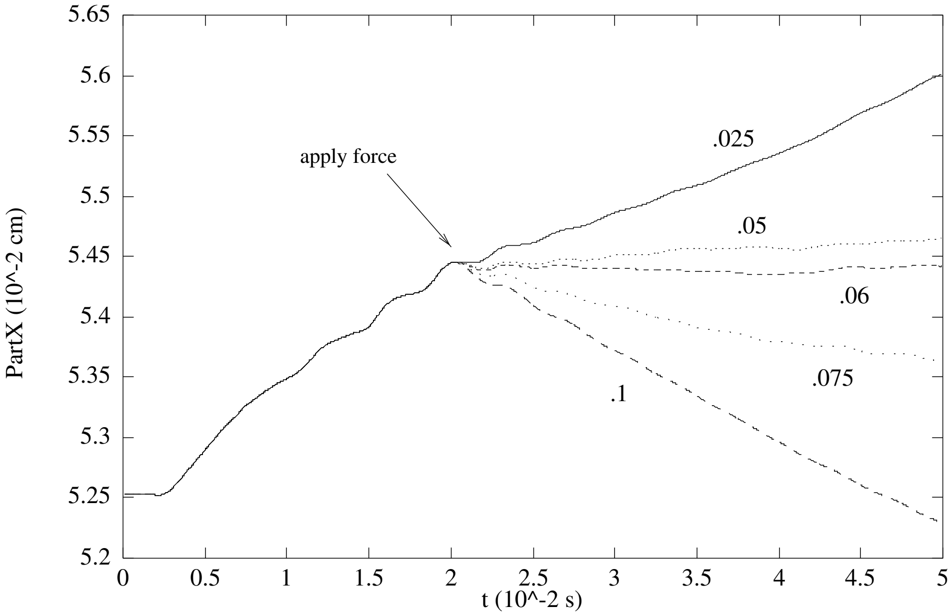


Figure 2.10: The change in ν for different values of F_{ext}. After the force is applied, the part reverts to a new constant velocity ν' linearly related to F_{ext}, as in a viscous fluid.

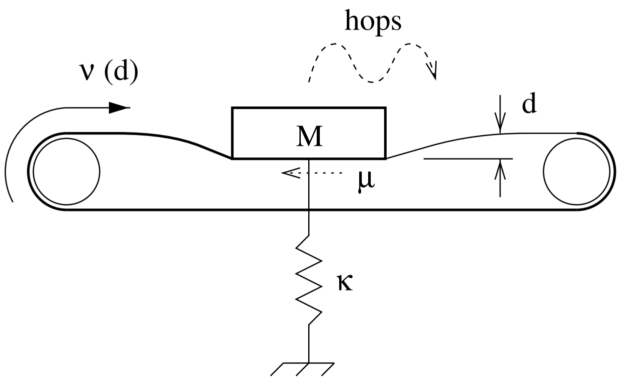


Figure 2.11: The array acts as a spring-loaded conveyor belt over which the part hops. The friction is viscous with coefficient μ. The average part height d is a function of the part's mass M and the supporting spring's stiffness k. The belt's feed rate ν is a function of d.

Chapter 3

The Coulomb Pump: a Novel Parts Feeding Method using a Horizontally-Vibrating Surface

Part feeders, which singulate and orient parts prior to packing and insertion, are critical components of an assembly line and one of the biggest obstacles to flexible assembly. Vibratory bowl feeders use 3d vibrations of a rigid bowl to feed industrial parts along a helical track [11]. In each vibration cycle, a part undergoes sticking, hopping, landing, and sliding motion modes. In such applications one is not concerned with part positioning accuracy nor with premature part wear due to repeated collisions with the feeder.

In the spirit of minimalism [12], we consider a simpler vibrations-based parts feeder design based on a horizontally vibrating flat plate, as shown in Figure 3.1. A linear motor (e.g., a voice coil) accelerates the plate along a single degree of freedom. Parts lay flush with the plate;

force is transferred to the part via dynamic friction (assuming the plate vibrates fast enough). This reduces part wear (assuming sliding friction is better than rigid collisions), and improves part motion controllability.

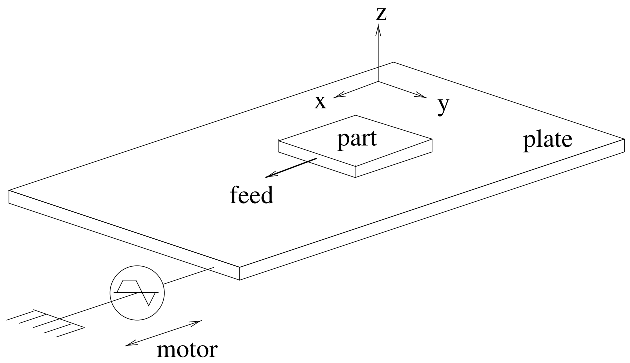


Figure 3.1: Our proposed feeder design: a flat plate, actuated by a linear motor, vibrates along a single degree of freedom. Parts lay flush with the plate. The plate's pump-like motion results in positive net dynamic frictional force applied to the part per cycle, causing the part to feed forward.

Each plate vibration is an asymmetric, pump-like closed motion along a single degree of freedom: the plate spends more time moving forward than backward. Since dynamic frictional forces are independent of the relative velocity at the part-plate interface, this type of motion results in positive net force applied to the part per cycle, resulting in forward feeding.

In designing plate vibration profiles, i.e., the waveform input to the motor, we consider issues of wave simplicity (ease of synthesis) and low bandwidth (anti-resonance). Two "canonical" waveforms types are analyzed: bang-bang and sinusoidal. In the former, the motor is driven at either full or zero throttle; in the latter, input to the motor consists a low frequency sine plus its first harmonic. Analytic expressions for both feeding forces and feed rates are derived. A feed rate performance measure is defined which allows for some quantitative comparison of the two methods. Results for the sinusoidal method are informally verified with dynamic simulation.

A prototype of the feeder has been built with inexpensive parts. A simple part feeding experiment is presented.

The remainder of this Chapter is organized as follows: In Section 3.1 we present the part feeder's mechanical model and the principle of pump-like motion, or "time-asymmetry". In Section 3.2, we consider motion waveform design and compare two possible controls: bang-bang and sinusoidal. In Section 3.3 we present a simple experiment performed with a prototype of the feeder. A Summary is presented in Section 3.4.

3.1 Part Feeding Principles

3.1.1 Feeder model

The parts feeder we consider is illustrated in Figure 3.2. A waveform generator W produces a periodic signal which is power-amplified (A) and fed to a linear motor M. The motor converts input energy into output force f_{ext}, applied along x to a flat *plate* S. The plate is constrained to move along a single horizontal dimension (e.g., x), with gravity acting perpendicularly (along $-z$). The plate's x compliance (suspension mechanism) is modeled as a damped-spring (k, b). One or more *parts* P are placed over the plate's surface.

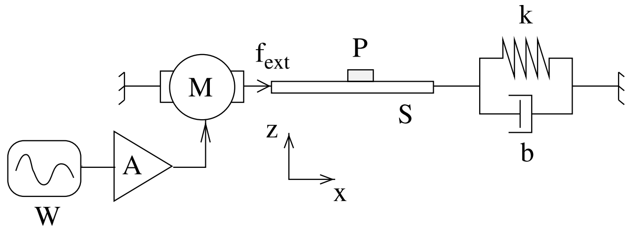


Figure 3.2: Parts feeder model: a flat plate S is actuated along x by a linear motor M; the plate's compliance/suspension is modeled as a damped spring (k, b). M converts the amplified signal generated by W into output force f_{ext} applied to S along x.

Let $a_s(t) = f_{ext}/m_s$ denote the plate's instantaneous acceleration, where m_s is the plate's mass (parts' masses are negligible). In what follows we will ignore amplification and motor transfer function issues and assume $a_s(t)$ can be specified directly. Let $a_s(t)$ be a periodic function with period T. The feeder is a mass-spring system with resonance w_0 occurring at [14]:

$$w_0 = \sqrt{\frac{k}{m_s} - \left(\frac{b}{4km_s}\right)^2} \tag{3.1}$$

Imposing $T \ll 2\pi/w_0$, i.e., $a_s(t)$ well above resonance, affords us the following nice properties:

- Plate oscillations are bounded.
- The plate's *velocity profile* $\nu_s(t)$ is the perfect time-integral of $a_s(t)$ and has zero steady-state DC level [14].
- Surface-part relative accelerations are above the threshold of sticking[1] [26], i.e., P is always sliding on S.

The plate's surface is assumed smooth and with uniform coefficient of dynamic friction μ. Let $\nu_s(t)$, $\nu_p(t)$ denote the plate's and part's instantaneous velocities, respectively. The Coulomb model for dynamic friction [26] states that the instantaneous frictional force acting on P will be of fixed magnitude μmg and act opposite to the relative velocity, i.e.:

$$f_{fric}(t) = \mu mg \; sgn[\nu_s(t) - \nu_p(t)] \tag{3.2}$$

[1] Actually, short periods of sticking, ignored here, will occur near the zero-crossings of relative acceleration

3.1.2 Coulomb Sliding Friction

When two contacting surfaces slide one with respect to another (assume no lubricants are used, i.e., consider "dry" friction), a dissipative frictional force develops, opposite to the relative motion. Over a wide band of velocities (3 orders of magnitude, see below) the opposing force is nearly constant with respect to velocity, namely, it is equal to $\mu\vec{N}$, where $\vec{N}$ is the contact force (the normal) and μ is the constant of sliding friction. This widely used model is known as the Coulomb model for sliding friction [35].

In [2], it is shown that μ actually falls as relative speed increases. Measurements are given for a mild steel brake shoe interacting with a railway steel wheel: at wheel speeds (relative to the brake) of $\{.1, 1, 10, 100\}$ inches per second, μ varies as $\{.23, .19, .18\}$, i.e., 3 orders of magnitude in speed change decrease μ by a mere 20%. Because in our experiments the speeds considered fall in a much smaller range, it is appropriate to consider μ constant.

In [21] light is shed on the molecular-level reasons for which sliding friction is largely independent on relative velocity. The authors argue that (i) frictional forces are a result of the making and breaking of weak molecular bonds between two relatively moving interfaces; (ii) the number of such bonds is proportional to contact area; this effect is cancelled out by (iii) the fact that contact area is inversely proportional to velocity, a non-trivial result, based on how the enmeshing of interface molecules is altered by speed.

3.1.3 Time asymmetry

In order for a part to feed forward, it must perceive a positive net frictional force as interacts with the plate over one oscillation cycle. The non-linear dependency of frictional force on relative velocity (we view independence as a non-linear law) suggests a simple approach to achieve part feeding: choose $a_s(t)$ such that $\nu_s(t)$ is positive for a longer time than it is negative. We term such a $\nu_s(t)$ *time-asymmetric*. For simplicity's sake, we make the following *quasi-static assumption*: the maximum change in part's speed in one cycle $\mu g T$ is negligible compared to the plate's peak velocity, $\nu_{s,max}$. Since the latter is bounded by $a_{max}T$, where a_{max} is the peak acceleration in one cycle, this requirement corresponds loosely to imposing $\mu g/a_{max} \ll 1$, which is facilitated by either one of: a slippery surface, a lightweight plate, and/or a powerful motor.

Let ν_p denote the part's "constant" velocity in one cycle. From Equation 3.2, we obtain an expression for the average force f_{1d} applied to the part over one cycle:

$$\bar{f}_{1d} = \frac{\mu m g}{T} \int_0^T sgn[\nu_s(t) - \nu_p]\, dt \tag{3.3}$$

The effect of the sgn function above is to "saturate" $\nu_s(t) - \nu_p$, i.e., transform it into a square wave. Define t^+ as the duration of the positive portion of $\nu_s(t) - \nu_p$. The average force will be linked to the *asymmetry* in the $sgn[\nu_s(t) - \nu_p]$ square wave, i.e., to t^+'s deviation from $T/2$. It can be shown that:

$$\bar{f}_{1d} = \mu m g(\frac{2t^+}{T} - 1) \tag{3.4}$$

If looked at the scale of several cycles, the part's velocity will increase if the force applied to it per cycle is non-zero, i.e., $t^+ > T/2$. This process will converge when the part reaches an equilibrium speed ν_{eq}, called the *feed rate*, such that $\bar{f}_{1d} = 0$. The concepts in this Section are illustrated in Figure 3.3.

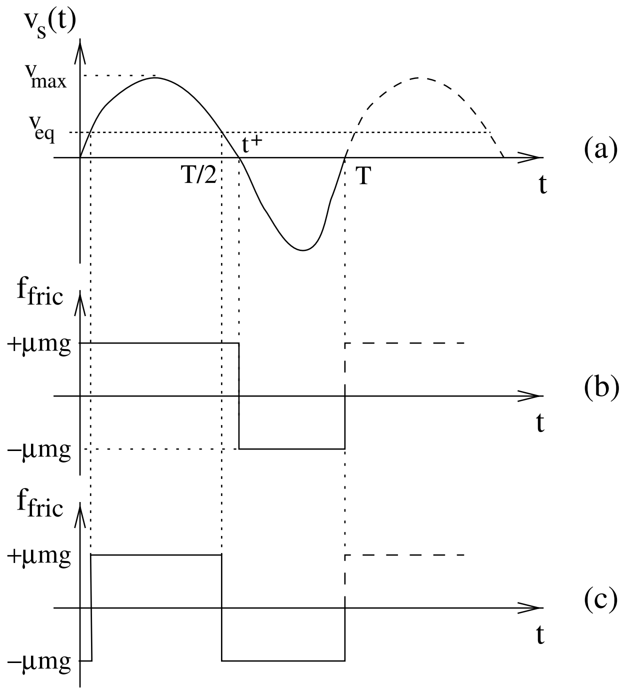


Figure 3.3: (a) A time-asymmetric velocity profile $\nu_s(t)$: the length of its positive semi-cycle is greater than $T/2$; $\nu_s(t)$'s peak value is ν_{max}. If the part's velocity is zero, $t^+ > T/2$. (b) Instantaneous frictional forces for a null part velocity: $\nu_s(t) - \nu_p$ is "saturated" into an asymmetric square wave. (c) The frictional force at equilibrium is a square wave with 50% duty cycle. The corresponding part velocity ν_{eq} is shown in (a) as the horizontal line which divides $\nu_s(t)$ into positive and negative semi-cycles of equal length $T/2$.

3.2 Designing the Control Waveform

In designing an $a_s(t)$ which achieves a time-asymmetric $\nu_s(t)$ we look for the following properties:

- Simplicity: a functionally simple $a_s(t)$ is easy to synthesize and parameterize.
- Bandwidth: low harmonic content reduces the chance of feeder resonance. Also, the motor will greatly attenuate $a_s(t)$'s high frequency components.
- Performance: one measure of performance is the ratio of the feed rate ν_{eq} by $\nu_s(t)$'s peak

value. Since this is bounded by $a_{max}T$, and will typically correspond to $\nu_s(t)$'s negative extremum $|\nu_{min}|$, we define the following two performance measures:

$$\begin{aligned} \lambda_1 &= \nu_{eq}/(a_{max}T) \\ \lambda_2 &= \nu_{eq}/|\nu_{min}| \end{aligned} \tag{3.5}$$

In what follows we present two $a_s(t)$ waveforms. The first focuses on waveform simplicity, while the second attempts to minimize spectral content. We then compare their $\lambda_{1,2}$ performances.

3.2.1 Bang-bang acceleration

A simplifying assumption is to expect that the motor can only deliver three types of forces: full throttle forward, zero force, and full throttle backward. As it turns out, if these three force application modes are repeated in sequence, the plate will move in a time-asymmetric manner. The corresponding $a_s(t)$ is the "bang-bang" waveform shown in Figure 3.4(a), and defined below:

$$\begin{aligned} a_s(t) &= \begin{cases} a_{max} & 0 \le t < t_1 \\ 0 & t_1 \le t < t_2 \\ -a_{max} & t_2 \le t < T \end{cases} \qquad (3.6) \\ t_1 &= (1-z)T/2 \\ t_2 &= (1+z)T/2 \end{aligned}$$

Where $z \in [0,1)$ is a parameter governing the zero-force phase duration. Namely, $a_s(t)$ 3 phases are: (i) positive constant acceleration, (ii) zero-acceleration, and (iii) negative constant acceleration. For simplicity's sake, we let phases (i) and (iii) be of equal duration; phase (ii)'s length is zT. a_{max} is the (bounded) acceleration desired in phases (i) and (iii). Integration (and removal of the DC term) yields a piecewise linear expression for the velocity profile:

$$\begin{aligned} \nu_s(t) &= \begin{cases} a_{max}[\frac{T}{4}(z^2-1)+t] & 0 \le t < t_1 \\ a_{max}\frac{T}{4}(z-1)^2 & t_1 \le t < t_2 \\ a_{max}[\frac{T}{4}(z^2+3)-t] & t_2 \le t < T \end{cases} \\ t_1, t_2 &= \text{as in above.} \end{aligned} \tag{3.7}$$

The above function is illustrated in Figure 3.4.

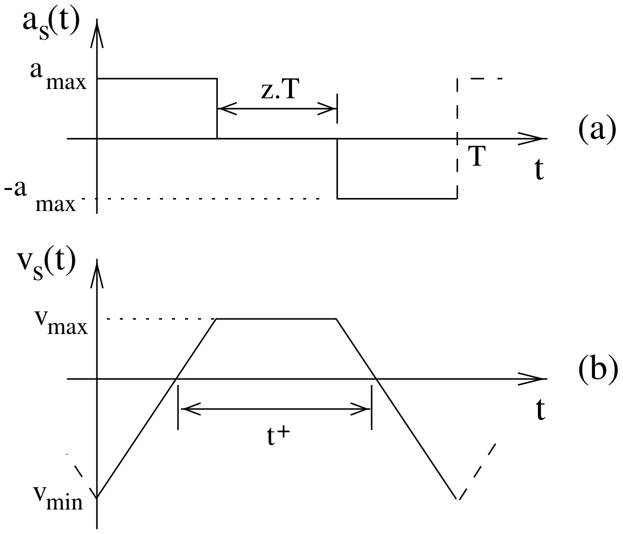


Figure 3.4: (a) A "bang-bang" acceleration profile with $z = 1/3$. (b) The corresponding velocity profile $\nu_s(t)$ shown with zero steady state DC value (due to springiness in the suspension). Shown also is the t^+, the length its positive semi-cycle.

The following expressions yield $\nu_s(t)$'s maximum and minimum values:

$$\begin{aligned} \nu_{min} &= a_{max}T\frac{(z^2-1)}{4} < 0 \\ \nu_{max} &= a_{max}T\frac{(z-1)^2}{4} > 0 \end{aligned} \tag{3.8}$$

For a fixed part velocity ν_p, we derive an expression for t^+ in $\nu_s(t) - \nu_p$:

$$t^+ = \frac{T}{2}(z^2 + 1) - \frac{2\nu_p}{a_{max}} \tag{3.9}$$

From Equation 3.4 obtain the average force applied to the part per cycle:

$$\bar{f}_{1d} = \mu m g \left(z^2 - \frac{\nu_p}{a_{max} T} \right) \tag{3.10}$$

The equilibrium velocity ν_{eq} is obtained by solving $\bar{f}_{1d} = 0$, above, for ν_p:

$$\nu_{eq} = a_{max} T \frac{z^2}{4} \tag{3.11}$$

The above equation is only valid for $z < 1/2$. Beyond $1/2$, ν_{eq} is “clamped” by ν_{max} (see [40] for details), and decreases to zero as z approaches 1, as shown in Figure 3.5.

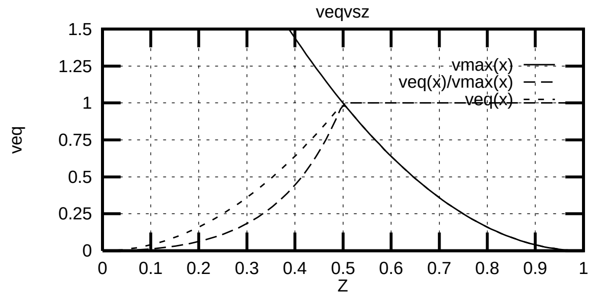


Figure 3.5: The feed rate ν_{eq} plotted against the z parameter, for $a_{max} = 16$, and and $T = 1$. Shown also the graphs of ν_{max} and the ratio $r = \nu_{eq}/\nu_{max}$. Notice that if $z > 1/2$, ν_{eq} tracks ν_{max}.

A maximum feed rate of $a_{max}T/16$ is achieved at $z = 1/2$ so $\lambda_1 = .0625$. Plugging $z = 1/2$ into Equation 3.8, obtain $\lambda_2 = 1/3$ for this type of control waveform.

Viscous force model

Taking the derivative of $\bar{f}_{1d}$ in Equation 3.10 with respect to ν_p and substituting in the value of ν_{eq} from Equation 3.11, yields an interesting expression for the average frictional force as a function of part speed:

$$\begin{aligned} \bar{f}_{1d} &= K(\nu_{eq} - \nu_p) \qquad (3.12) \\ \text{with } K &= \frac{4\mu m g}{a_{max} T} \end{aligned}$$

So in the average sense, the part perceives the plate (under bang-bang control) as a *viscous fluid* flowing forward at the feed rate, i.e., given enough time, the part's speed will approach that of the fluid.

3.2.2 Sinusoidal acceleration

Trivially, a cosine-like $a_s(t)$ integrates to a symmetric velocity profile. The lowest bandwidth $a_s(t)$ which integrates to a time-asymmetric waveform is a cosine plus a scaled, phase-shifted, double-frequency cosine:

$$a_s(t) = \cos(t) + 2b\cos(2t + \phi) \qquad (3.13)$$

$$\nu_s(t) = \sin(t) + b\sin(2t + \phi) \qquad (3.14)$$

where b and ϕ are arbitrary[2]. Equation 3.4 links $\bar{f}_{1d}$ to the roots of $\nu_s(t) - \nu_p$. It can be shown that Equation 3.14 will have either 2 or 4 real roots. Let us label these in ascending

[2] Interestingly, $a_s(t) = \cos(t) + b\cos(3t - \phi)$ integrates to a velocity profile with a symmetry of the type $\nu_s(t) = -\nu_s(-\pi + t)$, yielding zero feeding force for any b, ϕ.

order in the $[-\pi, \pi)$ interval as r_i, $i = 1 \ldots 4$. Without loss of generality, assume that $a_s(r_1)$ is positive. So the length of $\nu_s(t)$'s positive portion will be simply the sum of the distances between consecutive root pairs, i.e., $t^+ = (r_2 - r_1) + (r_4 - r_3)$. By expressing each root as a function of b and ϕ we can derive an expression for $\bar{f}_{1d}$ using Equation 3.4. In particular, with $\phi = \pi/2$, i.e., $\nu_s(t) = \sin(t) + b\cos(2t)$ we derive:

$$\begin{aligned}
\bar{f}_{1d} &= -\mu m g \frac{2}{\pi} c_1 + \begin{cases} -1 + \frac{2}{\pi} c_2 & b > 1 \\ 0 & |b| < 1 \\ +1 - \frac{2}{\pi} c_2 & b < 1 \end{cases} \\
c_1 &= \sin^{-1}\left(\frac{1 - \sqrt{1 + 8b^2}}{4b}\right) \\
c_2 &= \sin^{-1}\left(\frac{1 + \sqrt{1 + 8b^2}}{4b}\right)
\end{aligned} \tag{3.15}$$

A good approximation for the $|b| < 1$ range is $f_{1d} \cong b/3$. To derive an expression for ν_{eq}, we need to consider the roots of $\nu_s(t) - \nu_{eq}$, in particular we want $t^+ = T/2$. Careful analysis yields a surprisingly simple result:

$$\nu_{eq} = b \sin\phi \,,\ |b| < 1/2 \tag{3.16}$$

At $|b| > 1/2$, the number of roots of $\nu_s(t) - \nu_{eq}$ jumps from 2 to 4 causing ν_{eq} to decay, similar to saturation at $z > 1/2$ in the bang-bang case. In particular, for $\phi = \pi/2$, we obtain:

$$\nu_{eq} = \begin{cases} b & |b| < 1/2 \\ \frac{1}{4b} & |b| \geq 1/2 \end{cases} \tag{3.17}$$

At $b = 1/2$ and $\phi = \pi/2$, i.e., $\nu_s(t) = \sin(t) + \cos(2t)/2$, the feed rate is maximal[3]. $\bar{f}_{1d}$ and ν_{eq} versus b, with $\phi = \pi/2$, is plotted in Figure 3.6.

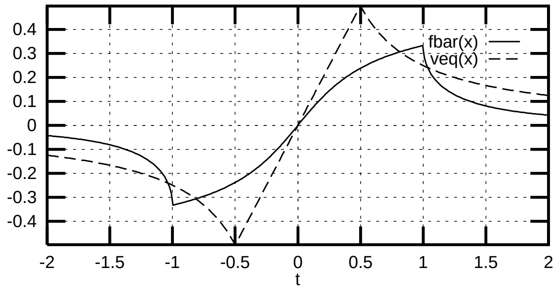


Figure 3.6: $\bar{f}_{1d}$ and ν_{eq} plotted against parameter b, for $\nu_s(t) = \sin(t) + b\cos(2t)$. Notice that these quantities are maximal at $b = 1$, and $b = 1/2$, respectively.

Consider the more general expressions for the ν_{eq}-optimal acceleration and velocity profiles:

$$a_{s,opt}(t) = \frac{a_{max}}{\gamma}\left[\cos(\omega t) - 2b\sin(2\omega t)\right] \tag{3.18}$$

$$\nu_s(t) = \frac{a_{max}}{\gamma}\left[\sin\omega t + b\cos(2\omega t)\right] \tag{3.19}$$

With $\omega = 2\pi/T$. The constant γ is the maximum absolute value of $\cos(t) + 2b\sin(2t)$. It normalizes $a_{s,opt}(t)$ so a_{max} becomes the actual maximum acceleration. Its value is given by:

$$\gamma = \frac{(3 + c_1)\sqrt{c_1 + 64b^2 - 1}}{32\sqrt{2}}$$

$$c_1 = \sqrt{1 + 128b^2}$$

At $b = 1/2$, $\gamma \cong 1.76$. The derived maximum, minimum, and equilibrium levels for this function are as follows:

[3]Equation 3.17 curiously implies that the "cousin" function $\cos(t) + b\sin(2t)$ ($\phi = -\pi$) yields zero feed rate, for any choice of b.

$$
\begin{aligned}
\nu_{min} &= -a_{max}T\frac{3}{4\pi\gamma} \\
\nu_{max} &= a_{max}T\frac{3}{8\pi\gamma} \\
\nu_{eq} &= a_{max}T\frac{1}{4\pi\gamma}
\end{aligned} \tag{3.20}
$$

Yielding $\lambda_1 = .0452$ and $\lambda_2 = 1/3$. Interestingly, the contraction in harmonic content afforded by the sinusoidal drive results in no penalty for λ_2 and a small reduction in λ_1, relative to the bang-bang method.

Dynamic simulation

Dynamic simulation was used to visualize part motion under sinusoidal acceleration (similar results for the bang-bang case can be found in [40]). Part motion is obtained through numerical integration of the instantaneous frictional forces corresponding to a $\nu_s(t) = \sin(t) + b\sin(2t + \phi)$ velocity profile. In Figure 3.7, the part speed vs. time is shown for various combinations of b and ϕ. The thickness of each curve is related to the variability in part speed during each cycle. As apparent, part speed approaches ν_{eq} as time advances. Figures 3.8(a,b) illustrate this phenomenon at a much smaller (per-cycle) scale. Namely, part speed becomes, at equilibrium, a triangular waveform of positive (resp. negative) derivative when $\nu_p < \nu_s(t)$ (resp. $\nu_p < \nu_s(t)$).

3.3 Hardware Experiments

A prototype of the parts feeder built out of inexpensive parts is shown in Figure 3.9. A thin slab of lightweight, smooth, kitchen-top material is used as the plate. Four motors are attached to the plate (one per side) by springy brass shafts, forming a combined suspension/actuation mechanism.

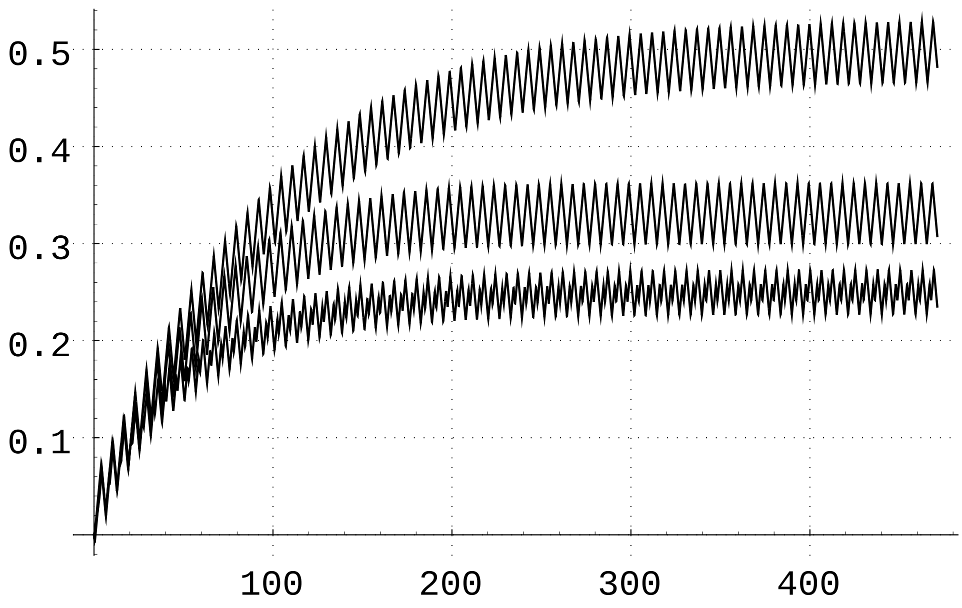


Figure 3.7: Dynamic simulation of part motion: under $\nu_s(t) = \sin(t) + b\sin(2t + \phi)$, the part's velocity is plotted against time for $(b, \phi) = \{(\frac{1}{2}, \frac{\pi}{2}), (1, \frac{\pi}{2}), (\frac{1}{2}, \frac{\pi}{4})\}$, corresponding to the top, middle, and lower curves. Each curve shows the part accelerating from zero velocity to the predicted feed rates $(\frac{1}{2}, \frac{1}{3}, \frac{1}{4})$. The curves' "thickness" reveal the change in part speed per cycle.

Each motor is a voice coil removed from old hard disks. The motors can command horizontal plate vibrations along its 3 dof's – this feature is used in our work on parallel part manipulation [38]. To achieve 1d vibrations, two opposing motors are turned off, while the other two opposing ones are operated in tandem, at 180 degrees phase shift. Figure 3.10 shows snapshots of a part feeding experiment – the bang-bang acceleration profile is used with $z = 1/3$ and $1/T = 30$ Hz. Parts propel forward at about 1 cm/s.

3.4 Summary

With respect to existing hop-based vibratory feeders, our design is mechanically simpler, reduces part wear caused by repeated collisions, and increases part motion controllability. Guide-

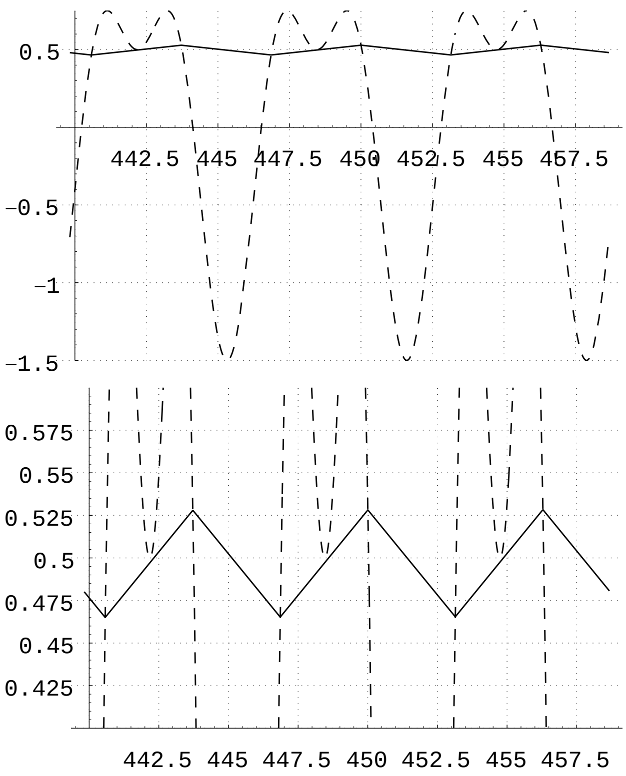


Figure 3.8: The part's speed (solid line) shown superimposed on the velocity profile (dashed line), for $(b, \phi) = (\frac{1}{2}, \frac{\pi}{2})$, at the equilibrium level $\nu_{eq} = \frac{1}{2}$. Top: part speed variability is small compared to ν_{max}. Bottom: part acceleration/de-acceleration visible at a smaller scale.

lines based on simplicity, bandwidth, and feed rate performance have been defined in designing a "good" control waveform. The sinusoidal method, despite its minimal bandwidth, results in a small performance loss with respect to the bang-bang control method. A sinusoidal waveform is more likely to be used in practice since it allows us to compensate for both the amplifier's and the motor's frequency response. Quantitative experiments with the feeder's prototype are the next step on this work. In particular, feed rates have to be measured and matched with the analytical results as waveform parameters (e.g., phase and amplitude relationship in sinusoidal case) are varied.

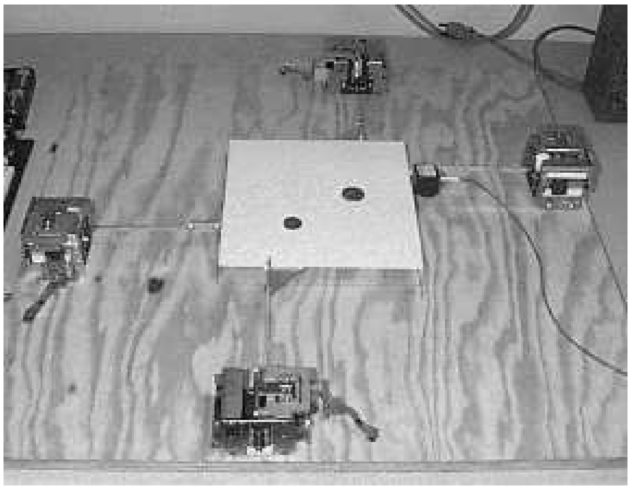


Figure 3.9: Our feeder's prototype: the plate is a thin square slab of kitchen-top material. Four disk drive motors are attached to the plate through thin brass shafts. Two coins are shown on the plate.

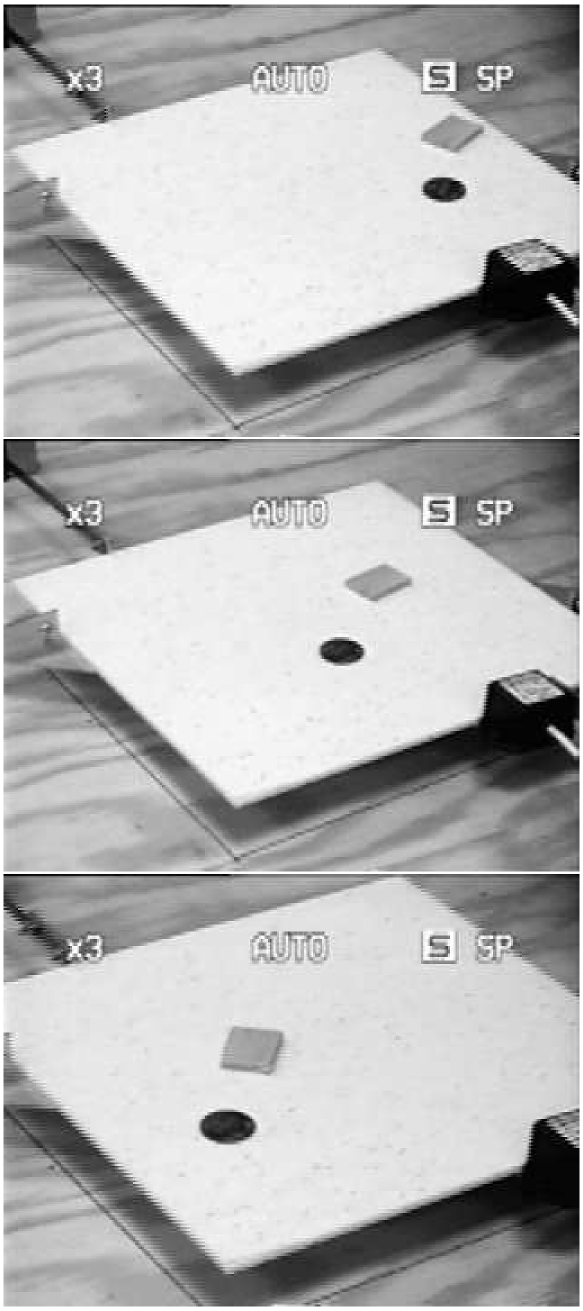


Figure 3.10: Three snapshots showing a penny and thin wood slab feeding along our feeder's prototype. The bang-bang acceleration method was used.

Chapter 4

A Flat Rigid Plate is a Universal Planar Manipulator

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 [9, 20, 43, 24, 31, 23].

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

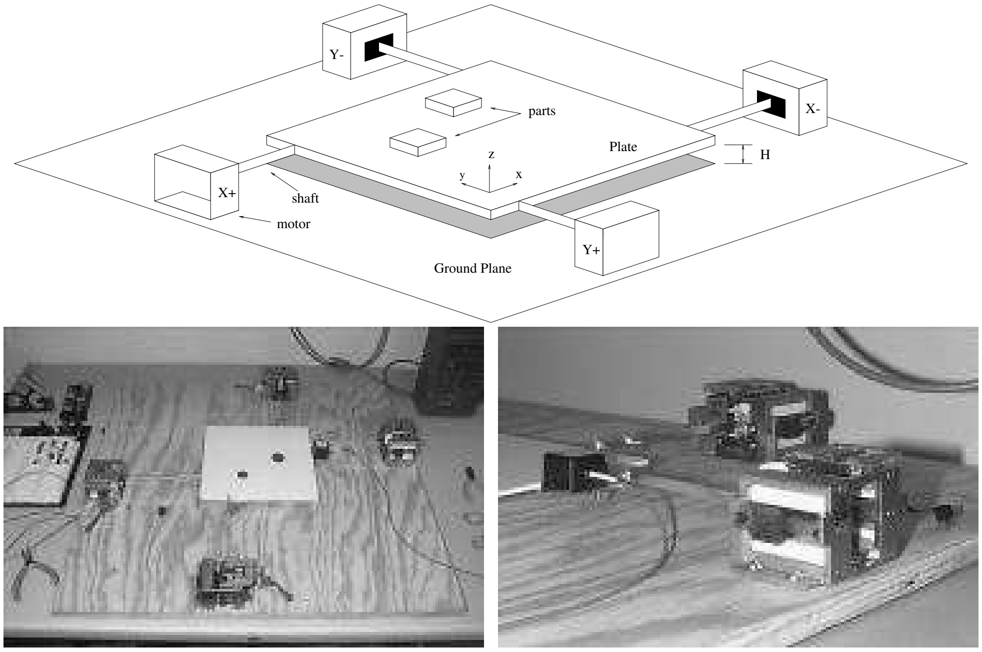


Figure 4.1: Top: Model of the proposed device – a flat horizontal plate, actuated by motors, is constrained to move horizontally. Parts are shown resting on the plate. Bottom left: a prototype of the device we've built with inexpensive parts: the plate is a thin square of smooth kitchen-top material; the motors are hard disk voice coils. Bottom right: a closeup of two motors, showing a motion sensor attached to the plate.

squared-cm [9]. The large number of tiny moving actuators can pose both fabrication and maintenance difficulties. In the spirit of minimalism [12], 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.

A model of our proposed device is shown in Figure 4.1. 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 [19]. 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 4.1. 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.

The remainder of this Chapter is organized as follows: In Section 4.1 we describe device kinematics, the friction model, and the non-additivity of frictional force fields. In Section 4.2 we describe the plate motion computation method based on a sequence of rigid rotations. In Section 4.3 we present two dynamic simulation results for two parallel manipulation examples. A Summary along with directions for future work is presented in Section 4.4.

4.1 Part Manipulation Principles

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 4.2(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:

$$\nu_s(P,t) \quad = \quad \dot{O}_s(t) + \dot{\theta}_s(t) \times [P - O_s(t)] \tag{4.1}$$

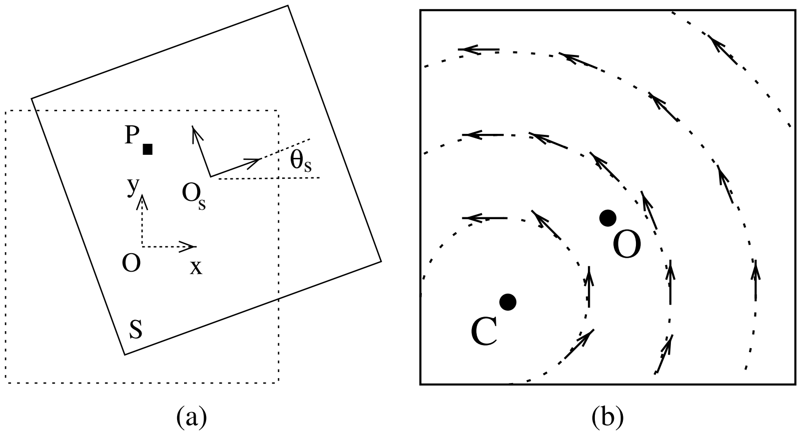


Figure 4.2: (a) The dotted square represents S at its home position, with its center at the origin O; The solid square represents S translated by O_s and rotated by θ_s. The part is at point P. (b) A rigid motion of S about C yields a frictional force field of tangential vectors of constant magnitude.

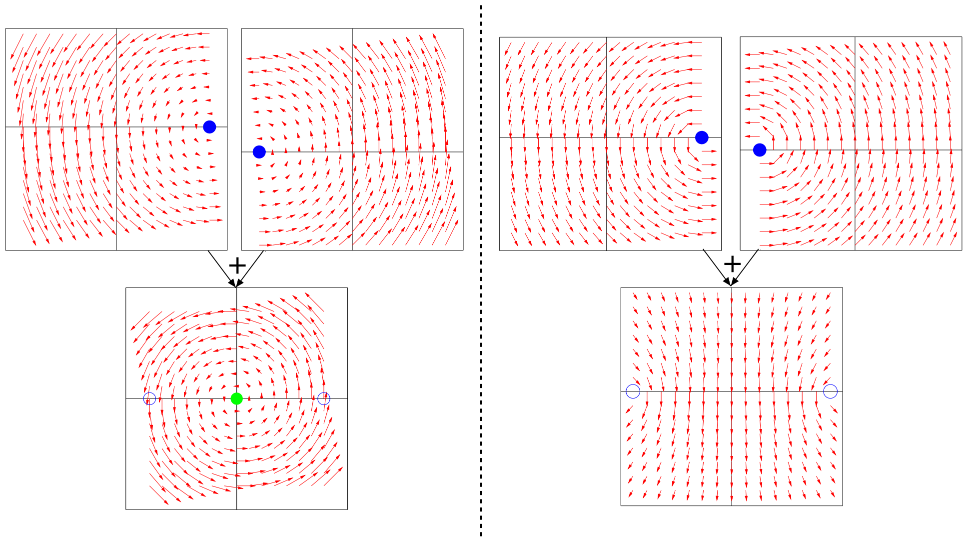


Figure 4.3: Left: the sum of two velocity fields is a velocity field. Right: the sum of two frictional force fields is not in the same family.

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 4.1 reduces to:

$$\begin{aligned} \nu_s(P,t) &= \dot{O}_s(t) + \dot{\theta}_s(t) \times P \\ &= [u_1(t) - y u_3(t), u_2(t) + x u_3(t)]^t \end{aligned} \qquad (4.2)$$

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 [26], 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 [26], i.e.:

$$f_{2d} = \mu m g \, \frac{(\nu_s - \nu_p)}{||\nu_s - \nu_p||} \qquad (4.3)$$

If the part's velocity $\nu_p(t)$ is negligible compared to $\nu_s(P,t)$ for most $t \in [0, T)$, Equation 4.3 reduces to:

$$f_{2d}(P,t) = \mu m g \frac{\nu_s(P,t)}{||\nu_s(P,t)||} \qquad (4.4)$$

[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.

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:

$$\bar{f}_{2d}(P) = \frac{1}{T} \int_0^T f_{2d}(P, t)\, dt \tag{4.5}$$

4.1.1 Instantaneous force fields

Chasles' theorem [14] 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:

$$[c_x, c_y, w] = [-u_2/u_3, u_1/u_3, u_3] \tag{4.6}$$

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:

$$\nu_s(P) = w \begin{bmatrix} c_y - y \\ x - c_x \end{bmatrix} \tag{4.7}$$

With each point on the plate moving tangentially, a stationary part at point P will experience an instantaneous frictional force of the type $f_{2d}(P) = \mu mg \hat{\nu_s}(P)$, 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 4.2(b): force at every point is perpendicular to the radial line, and of fixed magnitude μmg. 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.

4.1.2 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:

$$\begin{aligned} \psi_{k,C}(P) &= k\ \mu mg\ R_{90}(P - C) \\ \phi_{k,C}(P) &= k\ \mu mg\ R_{90}\frac{P - C}{||P - C||} \\ R_{90} &= \text{rotate-by-}90^{o}\text{-matrix} \end{aligned} \tag{4.8}$$

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

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

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 4.3.

4.1.3 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 4.4.

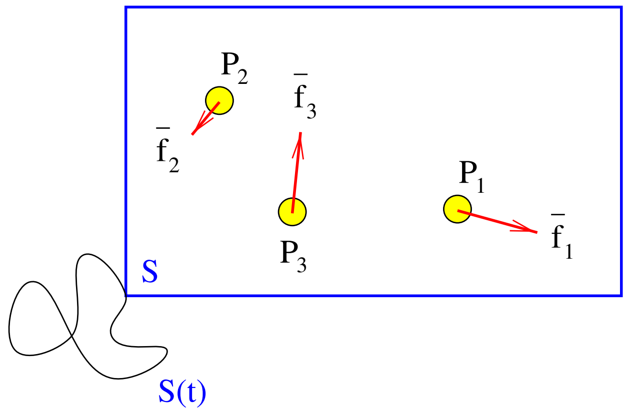


Figure 4.4: The parallel manipulation task consists in computing a closed motion $S(t)$ for the surface S which can generate desired frictional forces $\bar{f}_i$ (averaged over the motion) at each of the parts' locations P_i.

The setup envisioned is as shown in Figure 4.5. 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.

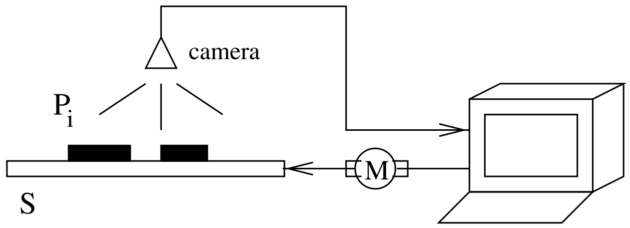


Figure 4.5: Closed-loop manipulation: a camera sends images to a computer, which sends commands to a motor which actuates the surface which moves the parts.

4.2 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 [37] (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:*

$$\bar{f_i} = \frac{1}{T}\sum_{j=1}^{N} \phi_j(P_i), i = 1 \ldots M \tag{4.9}$$

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 4.8 and 4.9 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:

$$\begin{aligned} A_{ij} &= \begin{cases} -a(i,j) & , \quad i = 1, \cdots M \\ b(i-M,j) & , \quad i = M+1, \cdots 2M \end{cases} \\ & \quad j = 1 \cdots N \\ b_i &= \frac{N}{\mu m g} \begin{cases} \bar{f}_i^x & , \quad i = 1, \cdots M \\ \bar{f}_{i-M}^y & , \quad i = M+1, \cdots 2M \end{cases} \end{aligned} \tag{4.10}$$

and:

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

4.2.1 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 4.6 with each such motion "scaled", in duration, by the computed k_j. The components of the 4×4 linear system become:

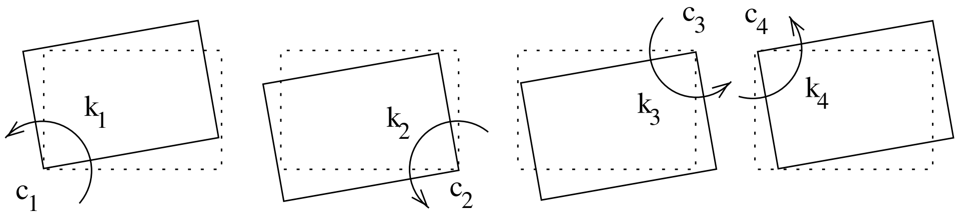


Figure 4.6: The surface executes 4 rotations about each of its corners, say, in CCW order. Each rotation is scaled (e.g., in duration) by a computed k_j.

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

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

By inverting A, compute $k = A^{-1}.b$ and obtain $K = (.58, -.39, .48, -.095)^t$. As shown in Figure 4.7, the resulting force field is as desired at the part's locations.

4.2.2 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 [42]. A simple expression yields the curl of a rotation-induced force field $\phi_{k,C}$:

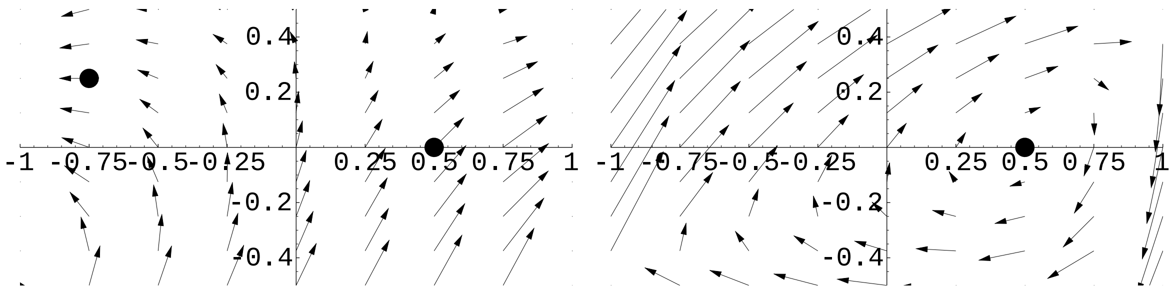


Figure 4.7: Left: average force field after 4 rotations about S's corners. The field points in the desired directions at the location of each part, shown as black disks. Right: average force field obtained after 3 rotations about C_1, C_2, and C_3 in sequence. Zero force and negative curl are desired at the part's location $(0.5, 0)$. The result is a CW whirlpool centered on the part.

$$\nabla\times\phi_{k,C}(P) \quad = \quad k\frac{\mu m g}{||P-C||} \tag{4.11}$$

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 4.9) can be written as:

$$\nabla\times f_i(P_i) \quad = \quad \frac{\mu m g}{T}\sum_{j=1}^{N}\frac{k_j}{||P_i-C_j||} \tag{4.12}$$

The above amounts to a single linear constraint, i.e., it becomes a row of the A matrix used in Equation 4.10. 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.

4.2.3 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:

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

Solving the above yields $K = (-.59, -.27, -.53)$. The resulting field, illustrated in Figure 4.7, 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.

4.2.4 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 4.2.1, $(\rho_1 \cdots \rho_4)^t = (.16, .031, .43, .34)^t$, 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 [32].

4.2.5 Ill-conditioning

The inverse of A (Equation 4.10) 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 [33]. Conveniently, this algorithm allows us to place maximum bounds on the entries of A^{-1} [33].

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.

4.2.6 Sensorless manipulation

Divergence relates to the shrinkage or expansion of a unit area element flowing through the field [42]. 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 [19] 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 4.5.

4.3 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.

4.3.1 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 4.8.

4.3.2 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 4.9.

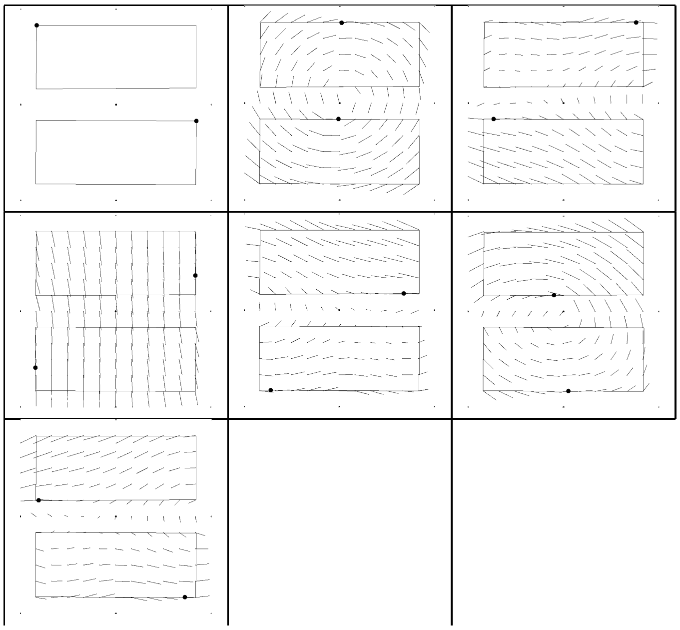


Figure 4.8: The trajectory following simulation. Read from left to right, top to bottom. At the top left snapshot, two parts are shown (dark disks) at the beginning of their planned rectangular trajectories. Net force field vectors (assuming superposition) are shown at each snapshot.

4.4 Summary

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.

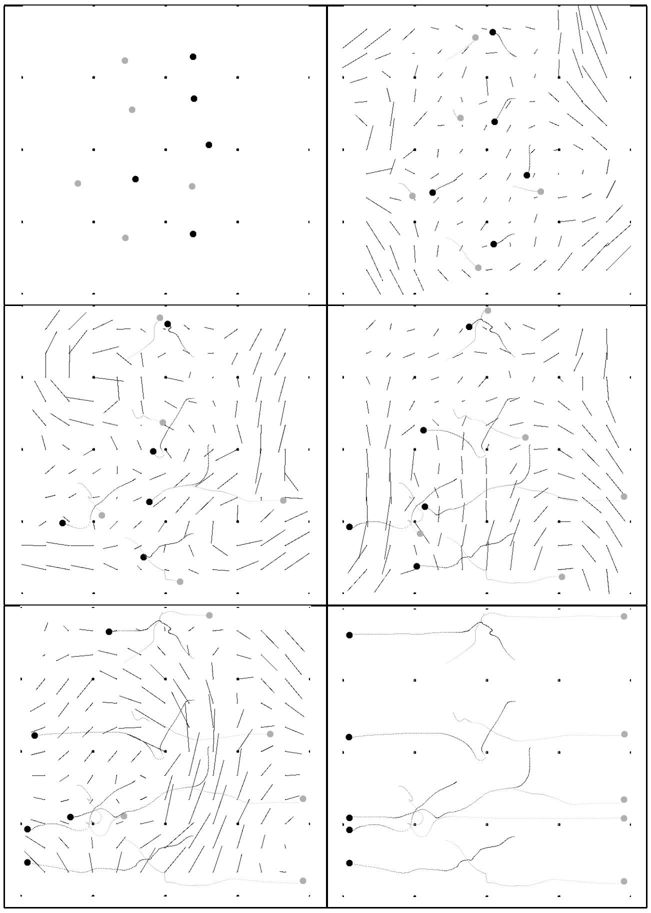


Figure 4.9: The part sorting simulation. The sequence should be read left to right, top to bottom. 5 dark and 5 grey disks appear in the first snapshot at their original locations. The goal is manipulate them in parallel so at the end all black go to the left and all grey go to the right. Force field 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.

Chapter 5

Building a Universal Planar Manipulator

Distributed manipulation devices make use of a large number of actuators, organized in array fashion, to manipulate a small number of parts [25, 19, 10]. Inspired by minimalism in robotics [12], 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 [11] 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 [37, 38, 39] 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 [37] 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 [37]). 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 [39]. 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 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.

This Chapter is organized as follows: in Section 5.1 we concisely review the theory behind our manipulation algorithm. In Section 5.2 we address practical implementation issues in building a prototype of our device. In Section 5.3 we present preliminary experimental results. A Summary is presented in Section 5.5.

5.1 Review

5.1.1 The Manipulation Algorithm

The manipulation problem being addressed is illustrated in Figure 5.1(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 5.1(b). This primitive causes parts to *flow* along a vector field $\phi_C = (\phi_C^x, \phi_C^y)^T$ defined as:

$$\phi_C = \frac{(P - C)^{\perp}}{||P - C||} \tag{5.1}$$

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

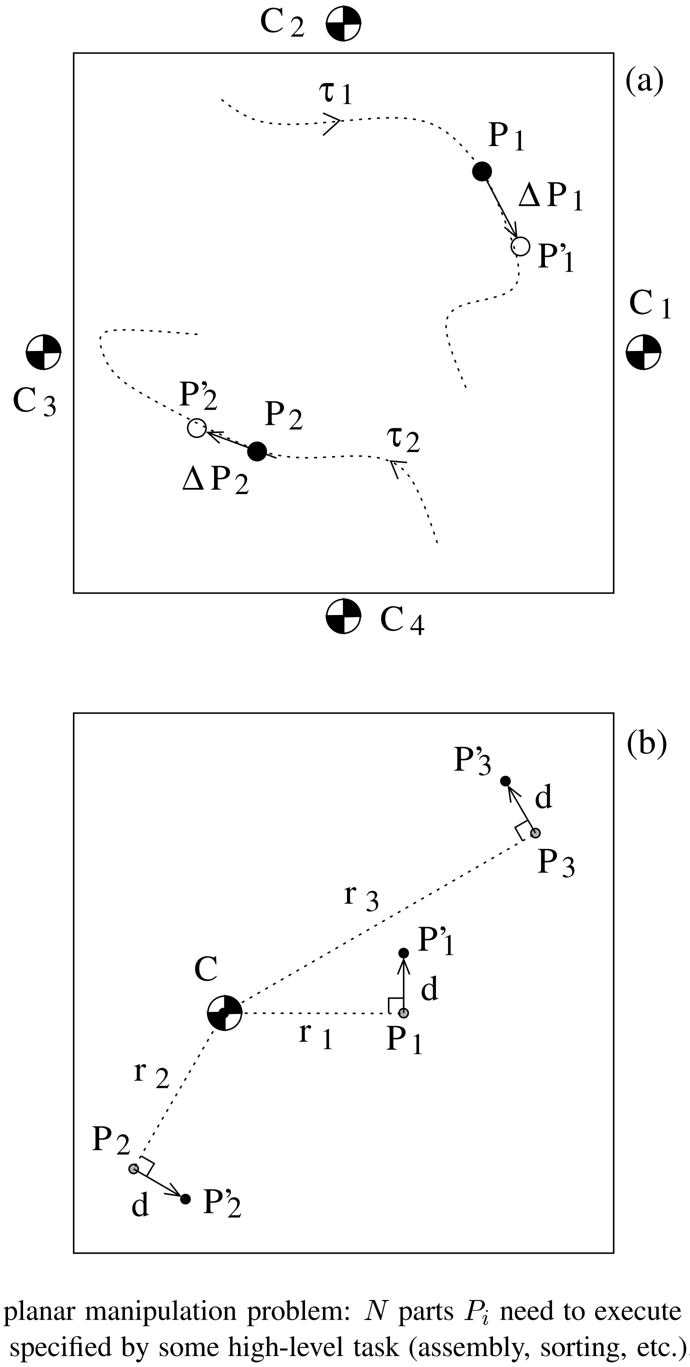


Figure 5.1: (a) The planar manipulation problem: N parts P_i need to execute a motion ΔP_i, e.g., along trajectories τ_i specified by some high-level task (assembly, sorting, etc.). A set of M points C_j is pre-specified about which the parts can execute a special type of rotation (see below). In the 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.

dimension 3. [38].

Define a set of scalars $d_j, j = 1...M$. Define $\Delta P'_i$ as part i's net displacement after it has flowed a distance d_j along ϕ_{C_j}, sequentially, for $j = 1...M$. 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 $\Delta P'_i$:

$$\Delta P'_i = \sum_{j=1}^{M} d_j \phi_{ij},\; i = 1...N \tag{5.2}$$

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

$$\Delta P = \Phi.d \tag{5.3}$$

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 5.3 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 5.2 and 5.3.

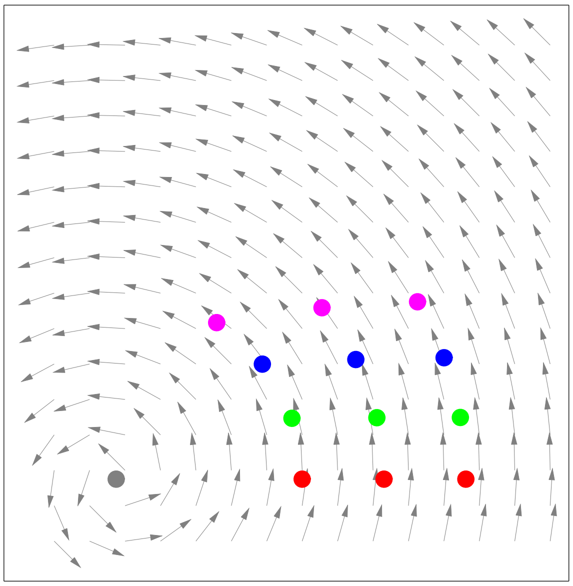


Figure 5.2: Race Track Experiment: three parts are allowed to "race" simultaneously (i.e., flow) along a non-linear rotation field. Their initial positions are all along a line directly to the right of the center of rotation, which is located at the lower left corner of the field. Four consecutive snapshots of the motion are shown. As expected, the inner parts advance more rapidly than the outer ones.

5.1.2 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

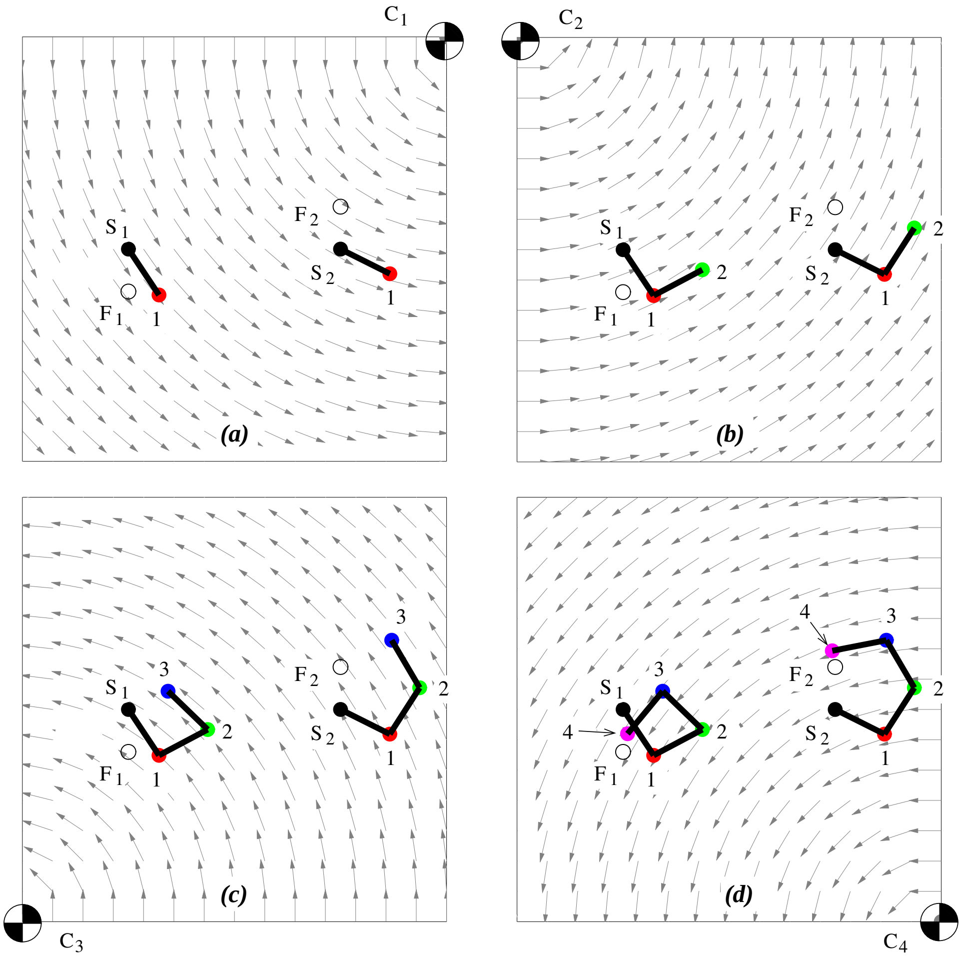


Figure 5.3: Four snapshots of a 2-part parallel manipulation problem: in (a) two parts are shown lying at starting locations S_1 and S_2; the goal is to move them to final locations F_1 and F_2. Four centers of rotation $C_j,\ j = 1 \ldots 4$ are specified, each at the corners of a square workspace. 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.

on S, with velocity ν_p. Assume S's 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:

$$\bar{f}_{1d} = \frac{\mu m g}{T} \int_0^T sgn[\nu_s(t) - \nu_p]\, dt \tag{5.4}$$

Define t^+ as the duration of positive $\nu_s(t) - \nu_p$ within one cycle. It can be shown [37] that:

$$\bar{f}_{1d} = \mu m g(\frac{2t^+}{T} - 1) \tag{5.5}$$

With $t^+ > T/2$, $\bar{f}_{1d}$ is positive, and the part will feed. In [37], we considered $\nu_s(t)$ of the form:

$$\nu_s(t) = \cos(wt) - \frac{1}{2}\cos(2wt) \tag{5.6}$$

With $w = 2\pi/T$.

Modulo a phase change, the above is equivalent to functions:

$$\begin{array}{rcl} \pm\sin(wt) & + & \frac{1}{2}\cos(2wt) \\ -\cos(wt) & - & \frac{1}{2}\cos(2wt) \end{array}$$

This particular velocity waveform was picked because it contains only two harmonics and delivers a large $\bar{f}_{1d}$ relative to its peak acceleration [37]. 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 5.6. 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 5.1.1.

5.2 Practical Challenges & Solutions

5.2.1 Actuation Kinematics

One way to accomplish the oscillatory surface motion prescribed in Section 5.1.2 is to have the surface's three dof's (x, y, and θ) move in phase with velocities as in Equation 5.6. Note that the instantaneous velocity of a rigid body in the plane is related to its instantaneous center of rotation by the following map [14]:

$$\begin{bmatrix} c_x \\ c_y \\ w \end{bmatrix} = \begin{bmatrix} -\dot{y}/\dot{\theta} \\ \dot{x}/\dot{\theta} \\ \dot{\theta} \end{bmatrix} \tag{5.7}$$

The actuation kinematics illustrated in Figure 5.4 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 5.4(a). At the operating frequencies, overall table displacements will be small, so we can decouple cross-talk between dof's.

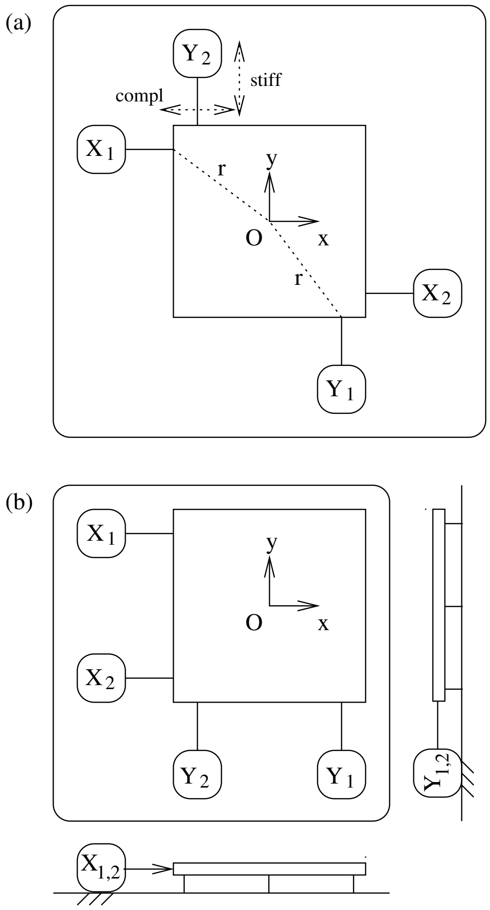


Figure 5.4: The actuation kinematics: (a) Four linear actuators, labeled $X_i, Y_i, i = 1, 2$, apply force to an individual side of the table through a shaft, attached at a distance r from the center of the table. Shafts are stiff along the driving direction and compliant perpendicularly. In the figure, 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-).

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:

$$\begin{aligned} f_x &= X_1 - X_2 \\ f_y &= Y_1 - Y_2 \\ \tau_\theta &= r[(Y_1 + Y_2) - (X_1 + X_2)] \end{aligned} \tag{5.8}$$

where r denotes the table's center distance to the actuation point on each side. In Figure 5.4(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:

$$f(t) = \cos(wt) - \sin(2wt) \tag{5.9}$$

scaled by chosen constants X_1, X_2, Y_1, Y_2. Then, because the map in (5.8) 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:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} (t) = \begin{bmatrix} \frac{1}{M} f_x \\ \frac{1}{M} f_y \\ \frac{r}{I} \tau_\theta \end{bmatrix} \frac{1}{w} [\sin(wt) + \frac{1}{2}\cos(2wt)] \tag{5.10}$$

Using Equations 5.7 and 5.8, 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.

5.2.2 Signal Generation and COR Visualization

We use *voice coils* [3] 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 5.9; a block diagram of the signal-generation hardware is shown in Figure 5.5. Two microcontrollers [27] 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 5.9, 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 [1] 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 Section 5.4.1 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:

$$|w| = \frac{||v_2 - v_1||}{2||r_1||} \tag{5.11}$$

$$C = \frac{(v_1 + v_2)^{\perp}}{2w} \tag{5.12}$$

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 5.5 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 5.7(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 [17]), yielding coefficients c_1, s_1, c_2, s_2, i.e., we fit the following "model" acceleration $a(t)$ to our data:

$$\begin{aligned} a(t) &= c_1 \cos(wt) + s_1 \sin(wt) \\ &+ c_2 \cos(2wt) + s_2 \sin(2wt) \end{aligned} \tag{5.13}$$

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 5.7(a) is shown in Figure 5.7(b). To visualize the least squares-fit velocity waveform, we simply integrate Equation 5.13, obtaining an identical waveform expressed as:

$$v(t) = \frac{1}{w}[-s1, c_1, -\frac{s_2}{2}, \frac{c_2}{2}] \tag{5.14}$$

This is used to generate the velocity waveform shown in Figure 5.13. 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 $\phi \in (0, \pi)$ [37]. In Section 5.4.2, we shown that this corresponds to the following expression in terms of the four free parameters in Equation 5.13:

$$sgn[\text{force}] = sgn[2 s_1 c_1 c_2 + s_2 (s_1^2 - c_1^2)] \tag{5.15}$$

5.2.3 Synthesizing Scaled Displacement Fields

Given an r_{min} and other dynamic parameters (such as motor power and coefficient of friction), $w(t)$'s amplitude and frequency can be chosen so that the part's velocity ν_p can be considered constant within one oscillation cycle. In that Section it also shown that if the part's velocity is negligible compared to the peak tangential velocity at P_i, call it ν_{max}, the average force $\bar{f}_{1d} = \bar{f}_0$ is constant and independent of ν_p. In particular, as ν_p grows, the force applied to it by the oscillating plate decreases, in linear viscous fashion. At $\nu_p = \frac{2}{3}\nu_{max}$, $\bar{f}_{1d} = 0$.

Consider a surface rotating periodically about point C with angular velocity $w(t)$ as in Equation 5.6. 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

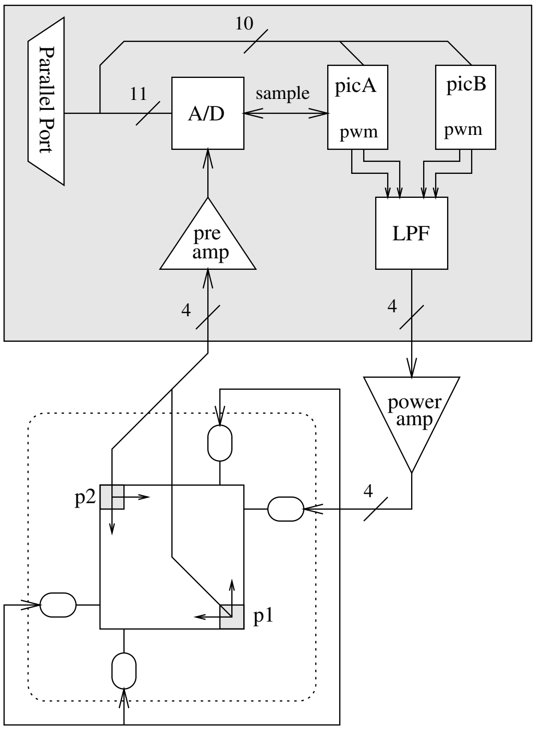


Figure 5.5: Block diagram of the signal generation and acceleration acquisition hardware: two microcontrollers (picA and picB) generate four independent PWM signals. These are low-pass filtered and power-amplified, and then applied to each motor. Two 2-axis accelerometers are glued under opposite corners 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.

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

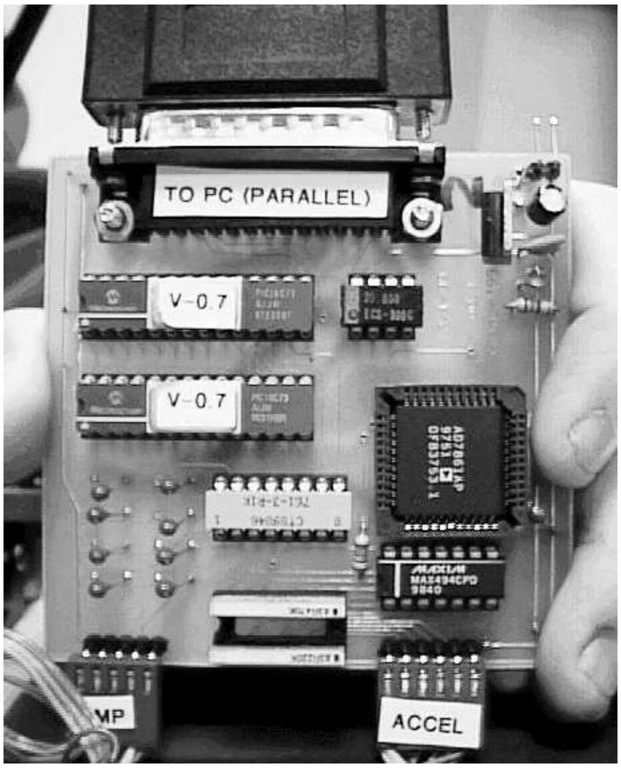
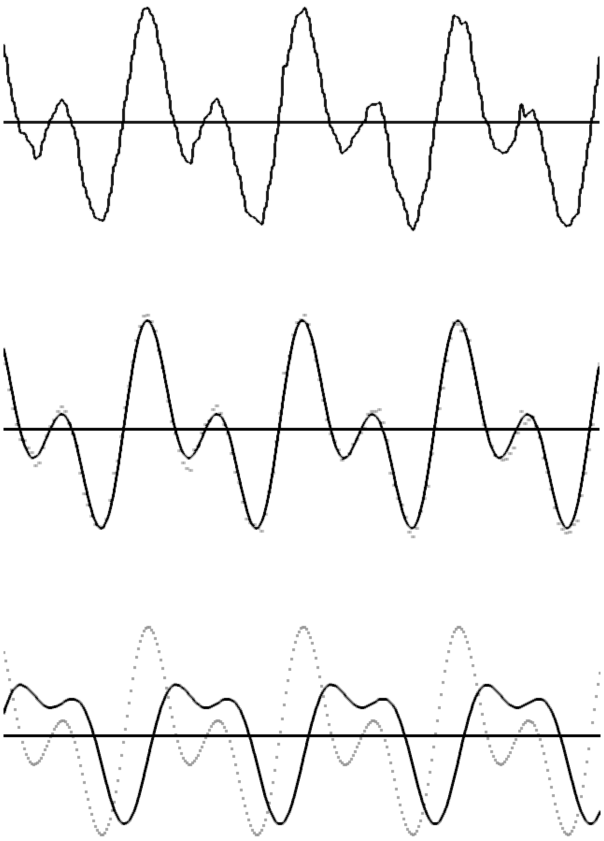


Figure 5.6: Photograph of the signal-generation and real-time acceleration acquisition hardware.

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 5.9. 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 5.8.



(a)

(b)

(c)

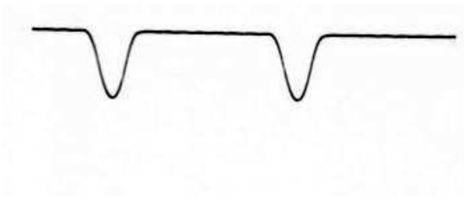
Figure 5.7: The waveform fitting process: (a) Samples coming from one sensed axis; (b) Least-squares fit (shown solid) and original samples (shown dotted); (c) Least-squares fit (shown dotted) and closed-form integral, i.e., the fitted velocity signal (shown solid).

5.2.4 Tracking Parts

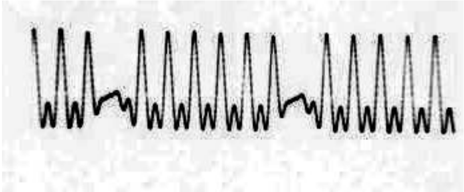
A camera is placed a few feet above the table pointing downward at the latter's center. The ground is black, the table is white and the parts are pennies painted black.

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 [41]. Each edge in the image

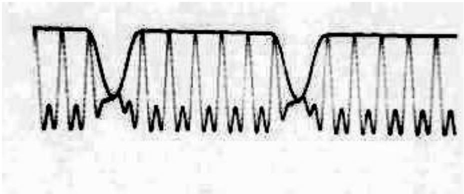
(a)



(b)



(c)



(d)

Figure 5.8: 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).

is then hashed by its distance to the image's center and angle onto a 2d Hough-vote array [41]. Edges making up the table's four sides will cluster at four locations on the Hough-array. Each of the Hough peaks gives rise 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

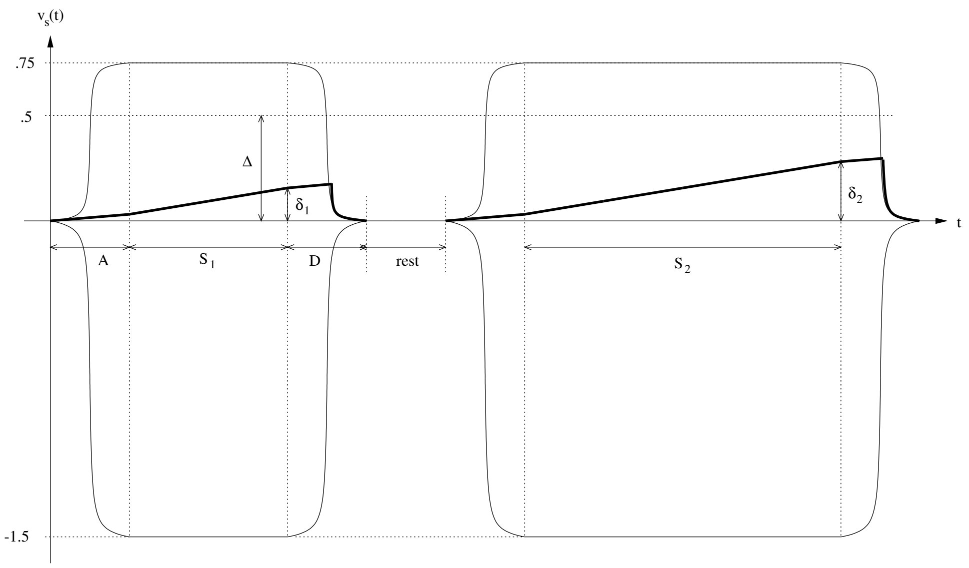


Figure 5.9: Emulating rotation intensity through pulse duration: two shaped pulses are shown. The pulse is represented by the *outline* of a normalized tangential velocity, covering $[-1.5, 0.75]$ along y (i.e., the range of $\cos(t) - \cos(2t)/2$). Each pulse contains 10s-100s of cycles of the basic 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.

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.

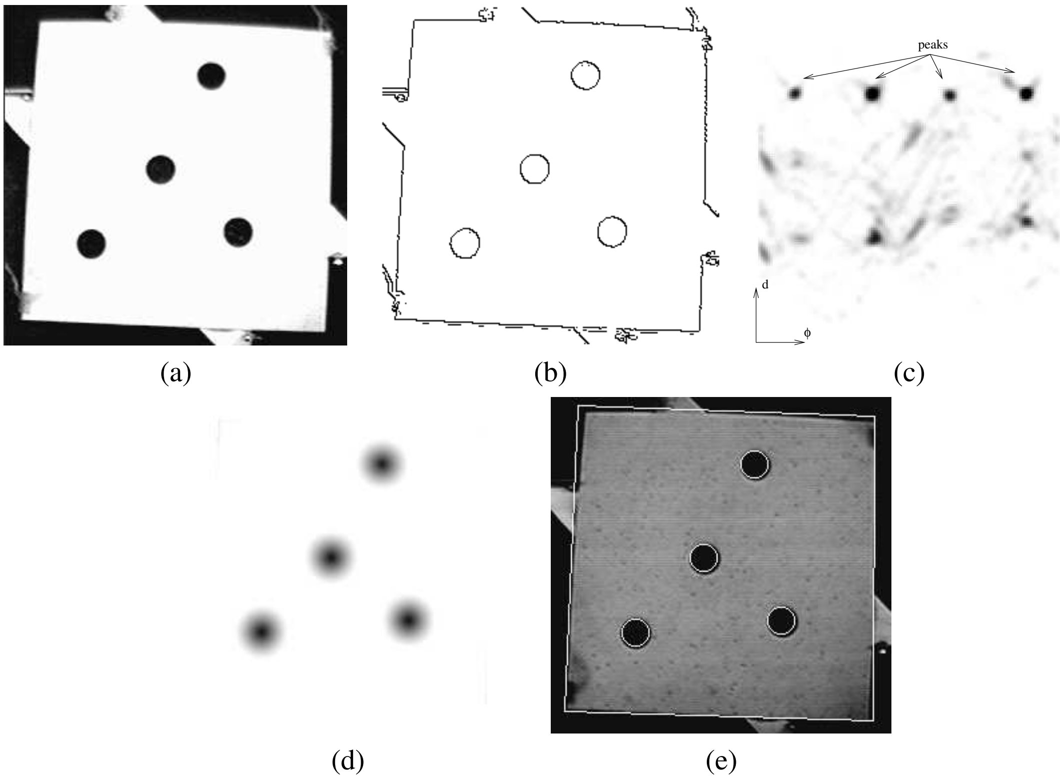


Figure 5.10: Image processing stages: (a) the original image, as seen from the camera overlooking the table. Four pennies have been placed on the table; (b) after edge detection; (c) Hough-vote space, showing the four peaks corresponding to the table's 4 long sides; (d) image convolved with a penny-sized disk, showing peaks at coins' centers; (e) final result showing detected features (table and coin outlines).

5.2.5 The Control Loop

The sequence of steps suggested in Section 5.1.1 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

5.3 Experiments

5.3.1 COR Steering and Calibration

Given the actuation kinematics in Figure 5.4(a), Equations 5.10 and 5.7 give rise to the following proportionality laws:

$$\begin{aligned} w &\propto (Y_1 + Y_2) - (X_1 + X_2) \\ c_x &\propto (Y_2 - Y_1)/w \\ c_y &\propto (X_1 - X_2)/w \end{aligned} \tag{5.16}$$

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 5.1. As 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 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 5.11 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 5.1.

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

	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 5.1: 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 5.16.

and recording the required amplitudes gives rise to a "COR library" which can then be used by our parallel manipulation algorithm.

5.3.2 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 5.12.

(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 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 5.13.

5.4 Useful Calculations

5.4.1 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 5.14. 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:

$$v_1 = w(r_1 - c)^{\perp} \tag{5.17}$$

$$v_2 = w(r_2 - c)^{\perp} \tag{5.18}$$

Taking the difference (5.17)-(5.18) eliminates c, i.e.:

$$v_2 - v1 = w(r_2 - r_1)^{\perp} = -2wr_1 \tag{5.19}$$

Which implies:

$$\begin{aligned} |w| &= \frac{||v_2 - v_1||}{2||r_1||} \\ sgn(w) &= sgn[(v_2 - v_1) \times r_1] \end{aligned} \tag{5.20}$$

Taking the sum (5.17)+(5.18) yields:

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

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

$$\begin{aligned} (v_1 + v_2)^{\perp} &= 2wc \\ c &= \frac{(v_1 + v_2)^{\perp}}{2w} \end{aligned} \tag{5.21}$$

With w computed as in (5.20). Equations 5.20 and 5.21 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 5.14). This method is inconvenient since the intersection is ill-defined with nearly parallel v_1 and v_2.

5.4.2 Positive Force Test

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

$$a_p(t) = \cos(t) + 2b\cos(2t + \phi) \tag{5.22}$$

In [37] we show that under the above plate motion, the part's equilibrium velocity ν_{eq} is $b\sin(\phi)$, with constant $|b| < 1/2$. 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, this implies that the *sign* of the average force is given by $sgn[b\sin(\phi)]$. An alternative representation for Equation 5.22 is:

$$\begin{aligned} a_p(t) &= c_1\cos(t) + s_1\sin(t) + \\ &\quad c_2\cos(2t) + s_2\sin(2t) \qquad (5.23) \\ &= m_1\cos(t-\alpha_1) + m_2\cos(2t-\alpha_2) \qquad (5.24) \\ (m_i,\alpha_i) &= (\sqrt{c_i^2+s_i^2}\,,\ \tan^{-1}\frac{s_i}{c_i}),\ i=1,2 \end{aligned}$$

Let $t' = t - \alpha_1$, then Equation 5.24 can be rewritten as:

$$a_p(t) = m_1[\cos(t') + \frac{m_2}{m_1}\cos(2t' + 2\alpha_1 - \alpha_2)] \qquad (5.25)$$

Modulo the m_1 scaling factor, Equation 5.25 is in the form of Equation 5.22, 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., $2\alpha_1 - \alpha_2 \in (0, \pi)$. 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 the previous condition is equivalent to stating $z_1^2 \times z_2 > 0$, or equivalently:

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

5.4.3 Table Dynamics

Let each actuation degree-of-freedom of the table be, under small displacements, modeled as a forced mass-damped-spring system, obeying the following linear differential equation [14]:

$$f_{ext} = m\ddot{q} + b\dot{q} + kq \qquad (5.26)$$

Where "q" represents the axis under question (x, y, or θ), and m, b, k are the mass (moment of inertia for θ), damping, and coefficient of friction along the appropriate axis. The resonance frequency w_0 of such a system is given by [14]:

$$w_0 \cong \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2} \tag{5.27}$$

In general, each axis will have distinct w_0's, i.e., $w_{0,x} \neq w_{0,y} \neq w_{0,\theta}$, so define:

$$w_{0,max} = \max\{w_{0,x}; w_{0,y}; w_{0,\theta}\}$$

For $f_{ext} = \sin(wt)$, the solution to Equation 5.26 is $q = \alpha(w)\sin(wt + \phi)$ [14], which implies w and $-w^2$ coefficients in the $\dot{q}$ and $\ddot{q}$ derivatives, respectively. So sufficiently above the system's highest resonance, i.e., $w >> w_{0,max}$, the inertial term in Equation 5.26 dominates over the damped-spring forces, and $f_{ext} \cong m\ddot{q}$, i.e., the velocity along that d.o.f. is proportional to the integral of the external force:

$$w \gg w_{0,max} \quad \Rightarrow \quad \dot{q}(t) = \frac{1}{m}\int f_{ext}(t)\, dt$$

5.4.4 Concatenation of Infinitesimal Flows

Let $\vec{U}$ and $\vec{V}$ denote two independent *vector fields* over $\mathbf{R}^n$ and q be a discrete point in that space, $q \in \mathbf{R}^n$. Starting at q, consider an ϵ-long flow along $\vec{U}$ followed by an equidistant flow along $\vec{V}$; call the resulting point q'. Proceed from the latter in ϵ steps along $-\vec{U}$, and $-\vec{V}$; call the resulting point q''. This process in illustrated in Figure 5.15. Equivalently:

$$
\begin{aligned}
q' &= \vec{V}_\epsilon \circ \vec{U}_\epsilon \circ q \\
q'' &= \vec{V}_{-\epsilon} \circ \vec{U}_{-\epsilon} \circ q'
\end{aligned}
$$

A straightforward Taylor expansion of the above [32] yields:

$$
\begin{aligned}
q' &= q + \epsilon(U+V) + \frac{\epsilon^2}{2}[[U,V]] + O(\epsilon^3) && (5.28) \\
q'' &= q + \epsilon^2[U,V] + O(\epsilon^3) && (5.29)
\end{aligned}
$$

Where:

$$
\begin{aligned}
[[U,V]] &= V'.(U+V) + (U+V)'.U && (5.30) \\
[U,V] &= V'.U - U'.V
\end{aligned}
$$

And U' is the *matrix* $[\partial U_i/\partial q_j]$ (similarly for V'). The reader will recognize $[U,V]$ as the *Lie-Bracket* [32] of the field pair, which arises when the flow is loop-like. In the half-loop case, the non-linear term $[[U,V]]$ has no symmetries (compare: $[U,V] = -[V,U]$), and, as evidenced by Equation 5.30, can grow out of bounds near high-derivative regions of U and V. For fields as in Equation 5.1, this corresponds to being close to the center of rotation. In summary, the second-order term in Equation 5.28 will be negligible if (i) ϵ is small and (ii) the entries in V' and $(U+V)'$ are evaluated not too close to the centers of rotation of U and V.

5.5 Summary

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.

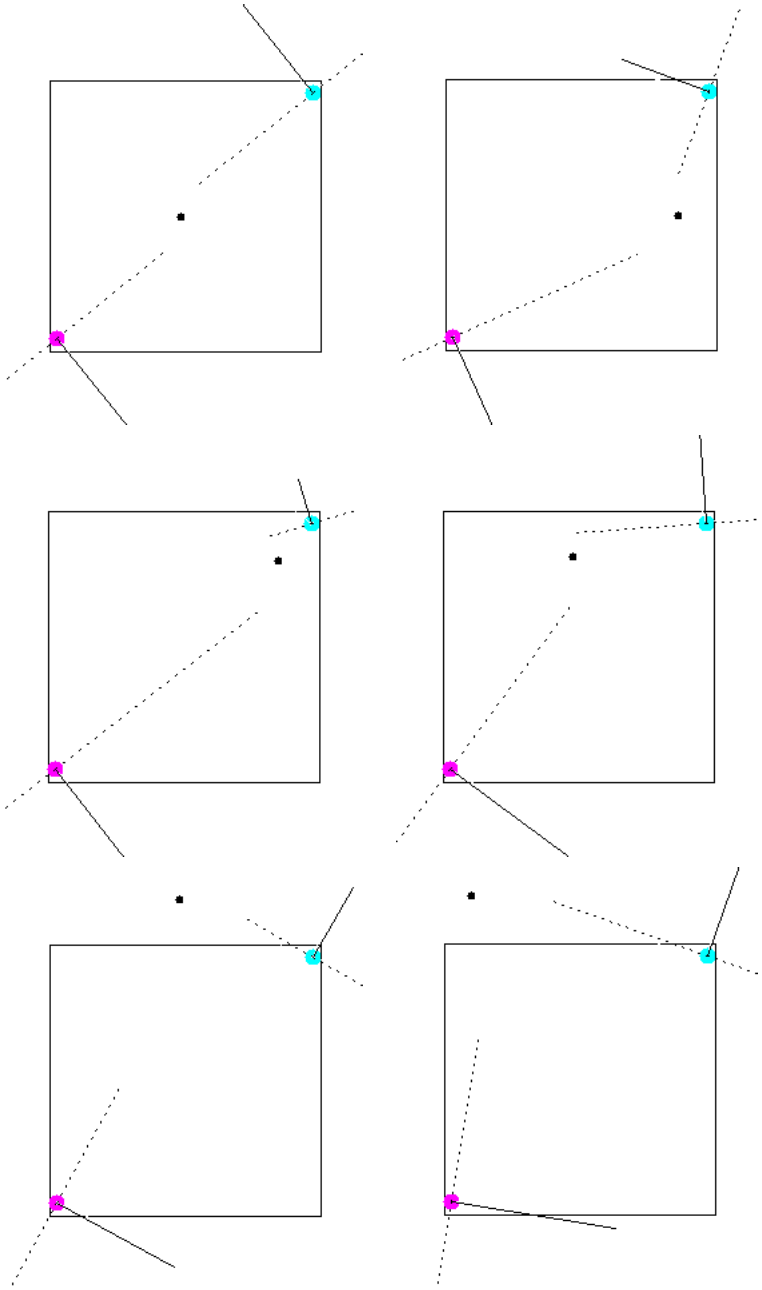


Figure 5.11: Steering the COR with the 6 amplitude combinations shown in Table 5.1. The table is drawn as an outline; the accelerometers are drawn centered at their actual locations near the lower-left and upper-right corners of the table. The actual magnitude of acceleration measured by each two-axis 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.

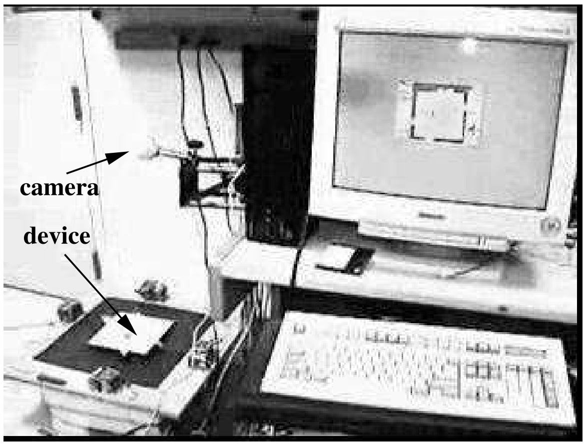


Figure 5.12: Experimental Setup for the 1-coin experiment: the computer, shaker table, and camera are visible.

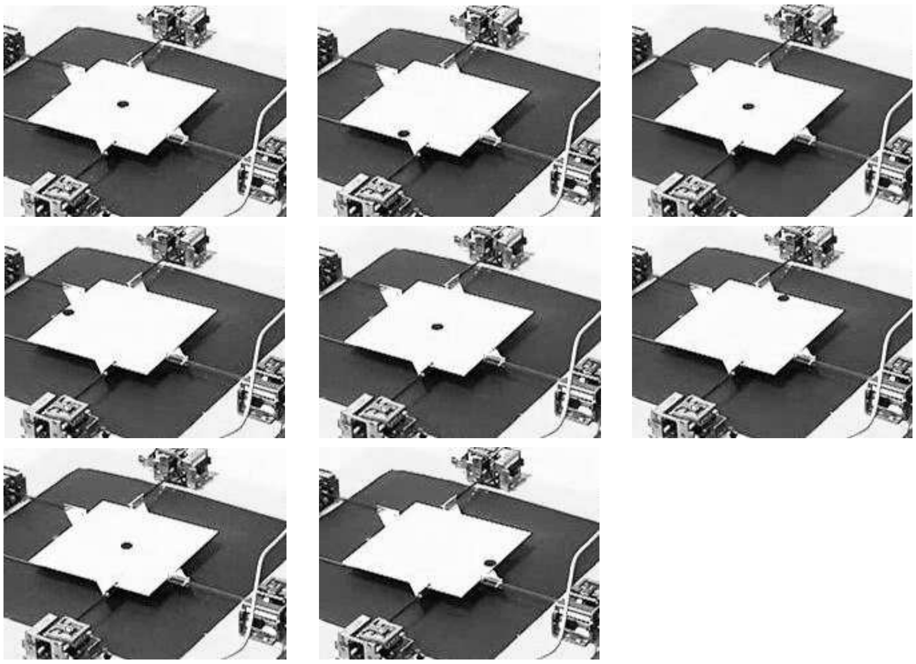


Figure 5.13: Eight consecutive snapshots (to be read left-to-right, top-to-bottom) of a simple visually-servoed trajectory-following task involving a single part (black penny). The plate is vibrated along x and y to steer the coin along the branches of an imaginary "plus" sign centered on the board. 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.

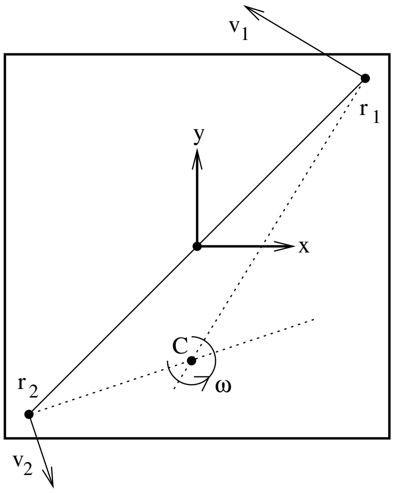


Figure 5.14: The shaker table is shown with two accelerometers placed at r_1 and r_2, with $r_2 = -r_1$. The instantaneous velocities at these points are v_1 and v_2, respectively. Lines L_1 and L_2 pass through r_1 and r_2, and are perpendicular to v_1 and v_2, respectively. The instantaneous center of rotation C and angular velocity w are also shown. Notice that C will lie at $L_1 \cap L_2$.

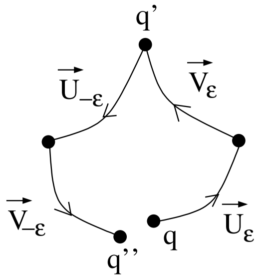


Figure 5.15: Concatenation of infinitesimal flows: The initial state q flows ϵ units along $\vec{U}$, and $\vec{V}$, leading to q'. The latter then flows ϵ units along $-\vec{U}$, and $\vec{V}$, yielding q''. When ϵ is small, the deviation $q'' - q$ is proportional to $[\vec{U}, \vec{V}]\epsilon^2$.

Chapter 6

Jets as a Local Motion Primitive

In this chapter we present a culmination of our research on vibrations-based planar part manipulation, centered on the Universal Planar Manipulator (UPM). The UPM is shown with some generic objects on its surface on Figure 6.1.

Figure 6.1: The Universal Planar Manipulator (UPM): generic objects placed on a flat plate can be manipulated independently via vibrations/friction.

The most interesting property of the UPM is its minimalism: despite its simple construction, the UPM can manipulate a large number of parts in parallel. Namely, given known locations of N parts, and N independent displacements desired for each part, a closed, rigid motion of the plate can be computed which, once executed, displaces all parts as desired. This result owes to the fact that, under Coulomb friction, distinct rigid rotations of the plate produce part displacement fields which are linearly-independent bases of the space of all possible part displacements [38, 7]. Namely, given desired displacements, a sum of (scaled) rigid plate rotations exists which produce them.

One problem is that a rigid body cannot execute a motion in the space of summed rigid rotations. Fortunately, the sum space can be approximated by a sequence of $M \geq 2N$ rotations about known centers $\mathbf{C}_j$, $j = 1, 2, \cdots, M$, provided each rotation displaces parts by a small amount. This was exploited in our original method [38]: given part positions and desired displacements, the system solved for rotation durations k_j, a process which required inverting a matrix. Though linearly independent, rotations are not orthogonal: each rotation displaces all parts, resulting in a cross-talk matrix with much off-diagonal energy. The end result is that matrix inversion is ill-conditioned, so computed durations are long, and part displacements are slow and noisy. This was partly addressed by adding redundancy to the linear system, i.e., more rotations per parallel update, $M \gg 2N$. This slows down execution and does not rule out ill-conditioning.

The main contribution on this paper is to present a new motion primitive – called the *jet* – which diagonalizes the inversion process, so that the solving step is robust and execution is significantly sped up. A jet is a force field "focused" on a single part, which can be told do move while keeping all others still. Parallel manipulation then reduces to applying a jet to each part in sequence. Because the jet's action is local to a part, execution time is proportional to the number of parts being moved, irrespective of how many currently sit on the plate. In the previous method, N parts required at least $2N$ rotations, even if just a single part was being displaced.

The jet idea is supported by a complete characterization of the feeding forces produced when plate motion is a sum of two sinusoids. We derive specific conditions under which two sinusoids produce maximal or no feeding forces, based purely on their relative phase and frequency. With this theory, we are able to pick the components of plate motion which generate optimally-focused (as-local-as-possible) jets.

This Chapter is organized as follows: In Section 6.1 we describe the jet, our new local force field. In Section 6.2, the main parts of the UPM prototype are explained. Experiments using jet fields are presented in Section 6.3. In Section 6.4 we present a characterization of feeding forces with two harmonics. A Summary is presented in Section 6.5.

6.1 The Jet as a Local Force Field

A new closed motion of S is described which integrates to a *local* force field called a jet. Local in the sense that it is only non-zero in the vicinity of a single part, and oriented along the part's desired motion.

Let the xy plane coincide with the horizontal plate S. S executes a *closed* motion lasting T seconds. "Closed" in the sense that at the end of the motion (time T) the surface returns to its initial position. Let $\mathbf{v}(\mathbf{P}, t)$ denote the instantaneous velocity at a point $\mathbf{P}$ in S. Consider a part of mass m lying at point $\mathbf{P}$ on S. Assume the part's speed is negligible with respect to $\mathbf{v}(\mathbf{P}, t)$. Assume plate motion is such that friction is always of the *sliding* type [35]. The part will perceive an instantaneous frictional force $\mathbf{f}(\mathbf{P}, t)$ of fixed value $\mu m g$ in the direction of $\mathbf{v}(\mathbf{P}, t)$, where μ, g are the constant of sliding friction and the acceleration of gravity, respectively:

$$\mathbf{f}(\mathbf{P}, t) = \mu m g \frac{\mathbf{v}(\mathbf{P}, t)}{||\mathbf{v}(\mathbf{P}, t)||} \tag{6.1}$$

Note: if $\mathbf{v} = \nu_x(t)$ is along a single direction, e.g., x, the above reduces to $\mu m g \, \mathrm{sgn}[\nu_x(t)]$. From

Equation (6.1) obtain the frictional force $\bar{\mathbf{f}}$ applied to the part *averaged* over the entire motion:

$$\bar{\mathbf{f}}(\mathbf{P}) = \frac{\mu m g}{T} \int_0^T \frac{\mathbf{v}(\mathbf{P}, t)}{||\mathbf{v}(\mathbf{P}, t)||} dt \tag{6.2}$$

Consider a 1d translational vibration $\mathbf{v}(t)$ of S along $\mathbf{d} = (d_x, d_y)$, the *feeding direction*. $\bar{\mathbf{f}}$ will be non-zero if $\mathbf{v}(t)$ is *time-asymmetric*, i.e., its positive and negative (along $\mathbf{d}$) portions have different durations [37]. A low-bandwidth $\mathbf{v}(t)$ is desirable since it avoids resonances in the mechanical system. Non-zero $\bar{\mathbf{f}}$ requires at least two sinusoids, since a single sine is time-symmetric. Corollary 1 in Section 6.4 states that the velocity profile:

$$\mathbf{v}(t) = \mathbf{d}[\cos(t) - \cos(2t)] \tag{6.3}$$

delivers the maximum possible force $\mu m g/3$ in the feeding direction over all choices of frequencies and phases for two sinusoids. To the above 1d motion, considering superimposing a sinusoidal rotation about a point $\mathbf{C}$, namely:

$$\mathbf{v}(\mathbf{P}, t) = \mathbf{d}[\cos(t) - \cos(2t)] + \frac{2|\mathbf{d}|}{\rho} \sin\left(\frac{2}{3}t\right) (\mathbf{P} - \mathbf{C})^{\perp} \tag{6.4}$$

Where ρ is a scaling constant for the rotation component. This motion is closed with period $T = 6\pi$. Near $\mathbf{C}$ the rotation component vanishes, and Equation (6.4) reduces to Equation (6.3). At large radii from $\mathbf{C}$, the rotation component (i.e., tangential velocities) dominates. Since this signal is time-symmetric, it will produce zero feeding forces. At distance ρ from $\mathbf{C}$, the rotational and translational waveforms have equal peak values. So parameter ρ can be set to control the rate of decay from maximum feeding force at $\mathbf{C}$ to zero at infinity. The larger the ρ, the more the field's active zone is "focused" (i.e., concentrated) on $\mathbf{C}$.

We chose $\sin(2t/3)$ for the rotational component so as to produce zero feeding forces with either component of Equation (6.3). This helps in creating "destructive interference" (in the feeding force sense) anywhere but in the vicinity of $\mathbf{C}$. We now refer to the results derived in Section 6.4. First, $\sin(2t/3)$ is non-feeding with $\cos(2t)$ because their frequencies are at a 1:3 ratio. Because both these numbers are odd, $\bar{\mathbf{f}} = 0$, Lemma 1. Second, $\sin(2t/3)$ is non-feeding with $\cos(t)$ because though their frequencies are at a 2:3 ratio (this is potentially feeding, Lemma 3), their phases are such that both waveforms have at least one coinciding root (at $t = 3\pi/2$), so $\bar{\mathbf{f}} = 0$, Corollary 2.

Other possible choices for the rotation component are $\sin(t) + \cos(t)$, $\sin(2t)$, and so on to higher frequencies. We opted for going below the feeding fundamental since for a given motor power, higher peak rotational velocities ($\propto \rho$) are feasible, yielding better-focused fields.

By plugging Equation (6.4) into the average force field integral, Equation (6.2), we obtain an average force field which is "local" to $\mathbf{C}$, as shown in Figure 6.2. We call this primitive a "jet", since it resembles a field of the same name in fluid mechanics. Under a jet field, only a part at $\mathbf{C}$ will experience any feeding force at all.

A straightforward step is to compute controls (forces and torques) which will position and orient the jet at will. Parallel manipulation then reduces to applying jets to individual parts, in round-robin fashion: for each part (i) track coin positions $\mathbf{P}_i, i = 1, 2, \cdots, N$ using vision; (ii) get desired displacements $\mathbf{d}_i$ from task; (iii) apply a jet field focused on $\mathbf{P}_i$ and oriented along $\mathbf{d}_i$.

Because local fields are nearly orthogonal (little cross talk), the solving step is direct (no matrix inversion required). While parts away from the jet's center do flow a bit (the field is small but non-zero there), this can be easily corrected with vision feedback. Jet-based manipulation is also much more scalable: if a subgroup of N_0 parts needs to be manipulated within a group of N parts only $N_0 \leq N$ jets are needed. In the old method, this required at least $2N$ rotations [38].

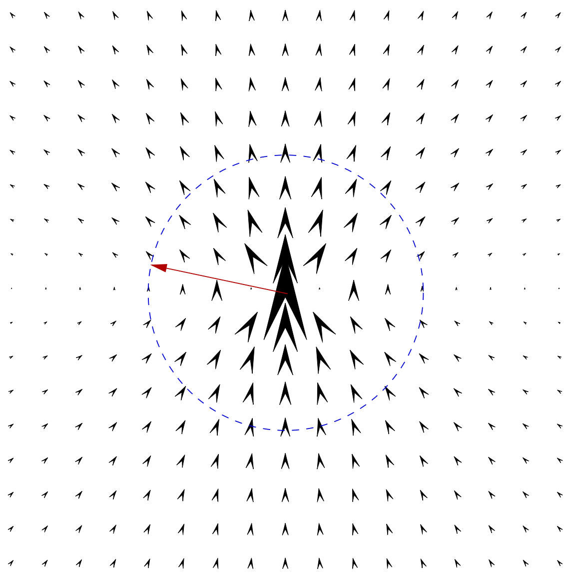


Figure 6.2: The jet force field: a feeding velocity $\hat{y}[\cos(t) - cos(2t)]$ superimposed to a rotation $\sin(2t/3)/\rho$ about the origin. Frictional forces are represented by scaled arrows. The dotted circle – a measure of the jet's focus – has radius 2ρ.

6.2 UPM Details

A block diagram of our prototype appears in Figure 6.3. Its various parts are explained next.

The plate itself is a 16"x16" tile of honeycomb material. Honeycomb is both cheap and has a very large stiffness-by-density ratio [16], i.e., vertical oscillations of the plate are kept to a minimum. The plate is constrained to move in its own plane by four vertical nylon rods supporting each of the plate's corners. This bearingless, flexure system is ideal since plate oscillations are of just a few millimeters.

Four voice coils (delivering up to 50 lbf each) actuate the plate in two differential pairs, along x and y. Each can apply either a force or a torque to the plate, depending on whether input signals are in- or out-of-phase, respectively.

For calibration purposes, two 2-axis accelerometers are installed at opposite corners of the plate. Prior to an experiment, the plate is run through a battery of standard motions; accelerometer data is used to compensate for distortions.

A camera is mounted over the plate and feeds color NTSC back to the controlling PC. This is used to locate the table's edges and track moving parts ("visual" feedback).

An interface board containing two micro-controller chips manages both the sampling of accelerometer signals and the generation of four phase-precise analog signals to the motors. A consumer-grade audio amplifier boosts the four analog signals to power levels required by each motor. To synthesize a jet, the PC downloads appropriate waveform parameters to the interface board. The PC commands the board to issue a pulse (a few cycles) of the four analog waveforms, causing the plate to vibrate and the parts to displace. Parts' positions are re-tracked, and the process repeats.

The jet waveform, Equation (6.4), inject three frequencies into the plate, call them f, $2f$, and $2f/3$. We found that $f = 35$Hz avoided any natural resonances of the system, while allowing for large peak velocities with the existing motors (large peak velocities promote both strong feeding forces and good jet focusing).

6.3 Experiments

Parallel manipulation under the new method is demonstrated in two experiments. First, we attempt to move three pennies along the same bowtie curve, Figure 6.4. In this 6-dof system, pennies have to reorganize themselves in a clearly non-rigid way as they traverse the curve. The controller ensures pennies remain equidistant from each other as they move along the curve. For

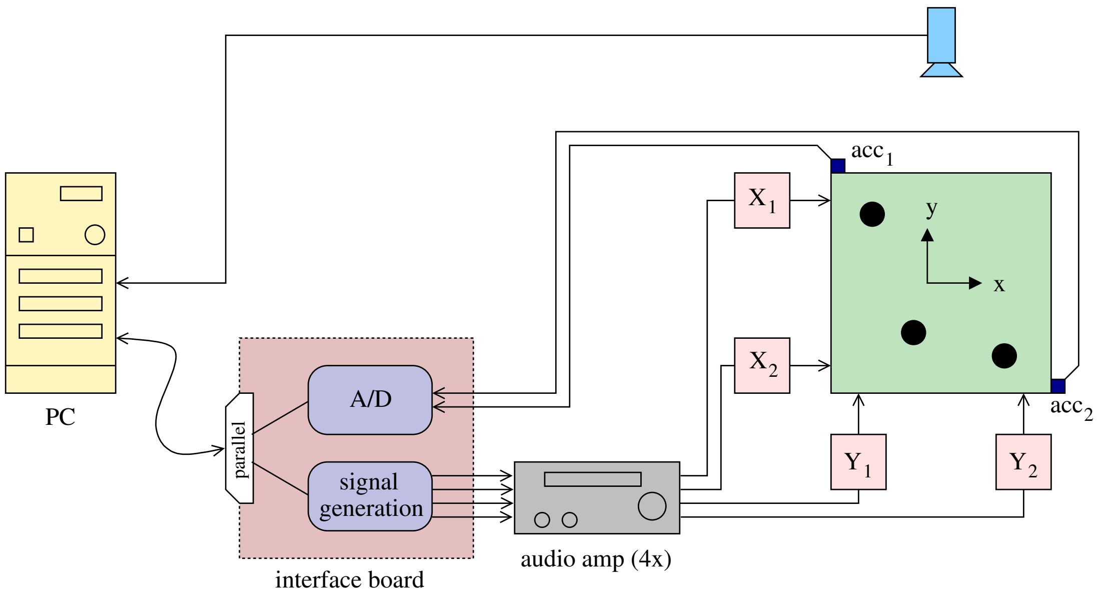


Figure 6.3: Block diagram of the UPM. Three parts (dark disks) are shown on the surface of the plate. The plate is actuated by four external voice coils (linear motors), organized in pairs X_1X_2 and Y_1Y_2. Two 2-axis accelerometers acc_1 and acc_2 are installed at opposite corners of the plate. A PC is connected to an interface board via a parallel port. The PC passes to the board motion parameters (phases, amplitudes, frequencies) for four independent analog signals. The board generates four signals with phase-precision. The signals are amplified by an audio amplifier and then fed to the four motors. The accelerometer signals (four in total) are sampled at the interface board and passed back to the PC for calibration purposes.

each update of the three coins, the system applies three jets, each centered at a specific coin. To speed up execution, a jet is executed in parallel with part tracking and motion computation for the next part (jet execution is the bottleneck). Snapshots of the experiment are shown in Figure 6.5

A second experiment involves the sorting of 8 plastic poker chips (a 16-dof system) based on color, Figure 6.6. Light and dark chips need to go to opposite sides of the UPM. An automatic labeling of part color is done by the vision system. The control loops involves applying jets to each individual part in the appropriate direction, round-robin. A better approach would include motion planning (e.g., using potential fields) but here the controller simply pushes chips to the appropriate side. Videos of these experiments can be found on the web at: `www.cs.berkeley.edu/~dreznik/UPM2000`

Manipulation with the UPM is not restricted to disk-shaped objects. As shown in Figure 6.1, this device can manipulate generic objects such as tools, bottles, etc. Because force scales with weight, objects with different weights will move at similar speeds, provided they have similar friction coefficients with the plate.

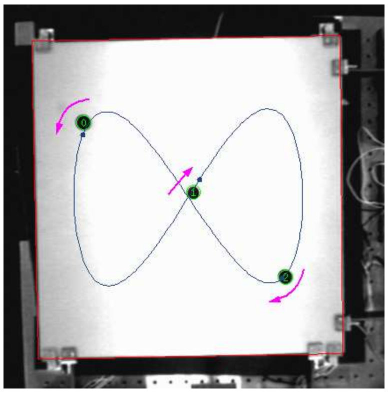


Figure 6.4: Bowtie experiment as viewed from the overhead camera. Three pennies and the intended bowtie curve are shown.

6.4 Part Feeding with two Sinusoids

In this Section, we characterize the frictional forces produced by plate motion made up to two sinusoids. The less mathematically inclined should skip the details and simply review the lemmas, theorem, and corollaries.

Consider a flat horizontal surface S; let the xy-plane lie on its surface, with the z-axis pointing upwards, opposite to gravity. Consider a rigid vibration of S along x, with velocity $\nu(t)$ of the form:

$$\nu(t) = \sin(f_1 t) + b \sin[f_2(t - \varphi)] \tag{6.5}$$

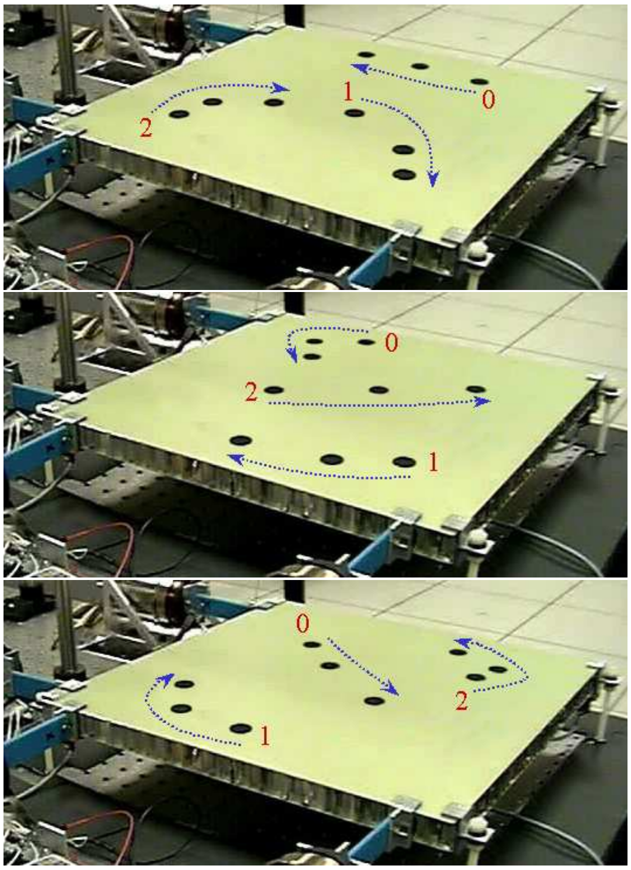


Figure 6.5: Snapshots of motion along the bowtie curve. Each snapshot combines 3 consecutive frames in "stop motion". Arrows label part motion across frames. Parts complete a loop around the bowtie in about a minute.

Parameters b and φ are relative amplitude and phase between the components. t and φ are given in radians. We consider the case of f_2/f_1 rational, $f_2 \geq f_1$. Let n_2/n_1 be the reduced fractional representation of f_2/f_1 in terms of two relatively prime integers [28] n_2, n_1, with $n_2 \geq n_1$, $\gcd(n_1, n_2) = 1$. Without loss of generality, we normalize $\nu(t)$'s period to 2π by writing:

$$\nu(t) = \sin(n_1 t) + b \sin[n_2(t - \varphi)] \tag{6.6}$$

Consider a part P lying on S with negligible velocity $\nu_p(t) \cong 0$, i.e., S's velocity relative to P is simply $\nu(t)$. We use Coulomb friction in sliding mode [35] as our model: the force S applied to

P is (i) in the direction of $\nu(t)$, and (ii) of constant magnitude μmg, where μ, m, g symbolize the frictional constant, the part's mass, and the acceleration of gravity, respectively. Then the average frictional force $\bar{f}$ applied to P per cycle is:

$$\bar{f} = \frac{\mu mg}{2\pi} \int_0^{2\pi} \mathrm{sgn}[\nu(t)]dt \tag{6.7}$$

where $\mathrm{sgn}[u]$ is the sign function, defined as $1, u \geq 0$, and -1 otherwise.

Lemma 1. *If both n_1 and n_2 are odd then $\bar{f} = 0$, for any choice of b, φ.*

Proof. A sinewave has the symmetry $\sin(nt) = -\sin[n(t-\pi)],\ \forall t$, provided n is odd. When both n_1 and n_2 are odd, each of Equation (6.6)'s harmonics will display this type of symmetry, and thus $\nu(t) = -\nu(t-\pi)$, i.e., $\mathrm{sgn}[\nu(t)]$ integrates to zero in $[0, 2\pi)$. □

Lemma 2. *For any choice of n_1, n_2, b, a phase φ exists which causes $\bar{f} = 0$, for any b.*

Proof. Choose $\varphi = 0$: $\nu(t)$ is a sum of two pure sines, i.e., it is an odd function with symmetry $\nu(t) = -\nu(t)$. So Equation (6.7) integrates to zero. □

Lemma 3. *If one of n_1, n_2 is even*[1] *then $\bar{f} \neq 0$ for some b, φ.*

Proof. First we choose a phase which gives feeding, namely we adjust φ so that the positive peak of the n_1 sinusoid aligns with the negative peak of n_2, allowing $\nu(t)$ to be written from Equation (6.6) as:

$$\nu(t) = \cos(n_1 t) - b\cos(n_2 t) \tag{6.8}$$

Now rewrite Equation (6.8) as the following product (using standard trig identities):

$$\nu(t) = 2\sin\left(\frac{\alpha t}{2}\right)\sin\left(\frac{\beta t}{2}\right) \tag{6.9}$$

[1] They can't both be even since they're relatively prime.

where $\alpha = n_2 - n_1$ and $\beta = n_2 + n_1$. Because exactly one of n_1, n_2 is even, both α, β are odd. It is also true that α, β are relatively prime since $\gcd(\alpha, \beta) = \gcd(\alpha - \beta, \beta) = \gcd(2n_2, \beta) = \gcd(\alpha, 2n_1)$. Therefore $\gcd(\alpha, \beta)$ is also a divisor of $2n_2$ and $2n_1$ and therefore of $2\gcd(n_1, n_2) = 2$. But it can't be two since α and β are odd. Thus $\gcd(\alpha, \beta) = 1$.

Since α and β are relatively prime, their least common multiple – the *lcm* – is given by their product:

$$\operatorname{lcm}(\alpha, \beta) = \alpha\beta = n_2^2 - n_1^2 \tag{6.10}$$

$\nu(t)$ has a root whenever either of Equation 6.9's factors is zero, i.e., t is a multiple of either $2\pi/\alpha$ or $2\pi/\beta$. Equivalently, the roots of $\nu(t)$ may only occur at "grid points" $t_k = 2k\pi/(\alpha\beta)$, $k = 0, 1, \cdots, \alpha\beta$. Define open intervals $\sigma_k = (t_{k-1}, t_k)$, $k = 1, 2, \cdots, \alpha\beta$, each of equal length $2\pi/(\alpha\beta)$. Within each σ_k, $\nu(t)$ has no roots, i.e., it is of constant sign. So the integral in Equation (6.7) becomes the following discrete sum:

$$\bar{f} = \frac{\mu m g}{\alpha\beta} \sum_{k=1}^{\alpha\beta} \operatorname{sgn}[\nu(\sigma_k)] \tag{6.11}$$

We note the quantity $\alpha\beta$ is odd, since it is the product of two odd numbers, i.e., Equation (6.11) is a sum of an odd number of ± 1's. An imbalance must exist in this sum, and therefore $\bar{f} \neq 0$. $\square$

Because $\bar{f}$ is a continuous function of b, φ, property $\bar{f} \neq 0$ holds true within an open neighborhood of $b = 1$ and the φ chosen to render $\nu(t)$ of the form of Equation (6.8).

The previous result is illustrated in Figure 6.7 for the case where $n_1 = 2$ and $n_2 = 5$. The sign imbalance in this example is exactly one σ_k, which turns out to be the maximum possible imbalance, explained next.

Theorem 1. *Consider a surface velocity of the form:*

$$\nu(t) = \cos(n_1 t) - \cos[n_2(t - \psi)] \tag{6.12}$$

where $n_2 > n_1$ are relatively prime, and exactly one is even. $\psi = 0$ yields the maximum possible average force $\bar{f} = \mu m g/(n_2^2 - n_1^2)$. In general, $\bar{f}$ is a triangular waveform on ψ, taking the following form:

$$\bar{f} = \frac{\mu m g}{n_2^2 - n_1^2} \triangle(n_1 n_2 \psi) \tag{6.13}$$

where $\triangle(t)$ is a unit-amplitude, triangular waveform of period 2π:

$$\triangle(t) = \begin{cases} 1 - 2|t|/\pi, & |t| < \pi \\ \triangle(|t| - 2\pi), & \textit{otherwise} \end{cases}$$

Proof. First, using trigonometric product formulae, rewrite Equation (6.12) as the product:

$$\begin{aligned} \nu(t) &= \sin\left[\frac{\alpha}{2}(t - \psi_1)\right] \sin\left[\frac{\beta}{2}(t - \psi_2)\right] & (6.14) \\ \psi_1 &= \frac{n_2}{n_2 - n_1}\psi & (6.15) \\ \psi_2 &= \frac{n_2}{n_2 + n_1}\psi & (6.16) \end{aligned}$$

where α, β are defined as in Equation (6.9). With the substitution $t \to t - \psi_1$ obtain:

$$\nu_\Psi(t) = 2h_\alpha(t)h_\beta(t - \Psi) \tag{6.17}$$

where $h_n(t)$ stands for $\sin(nt/2)$, and $\Psi = \psi_2 - \psi_1$ is obtained from Equations (6.15) and (6.16):

$$\Psi = \frac{-2n_1 n_2}{n_2^2 - n_1^2}\psi \tag{6.18}$$

Equation (6.17) has roots when either factor is zero, i.e., t is a multiple of either $2\pi/\alpha$ or $2\pi/\beta$. Let a time t for which both factors vanish be called a *common root*.

Case 1. $\Psi = 0$

By inspection, $t = 0$ is a common root, and no others may exist in $[0, 2\pi)$, as this would imply two non-negative integers $k_1 < \alpha$ and $k_2 < \beta$ exist such that $2\pi k_1/\alpha = 2\pi k_2/\beta$, i.e., $k_1\beta = k_2\alpha$. Because $\gcd(\alpha, \beta) = 1$, this is impossible. Notice that $\Psi = 0$ implies $\psi = 0$, Equation (6.18). Lemma 3 tells us that for $\psi = 0$, $\bar{f} \neq 0$. Call this non-zero average force $\bar{f}_0$. In general, we will use $\bar{f}_\Psi$ to denote the average force for the phase Ψ defined above.

Case 2. $0 < \Psi < \frac{2\pi}{\alpha\beta}$

In this range there are no common roots since none of $h_\beta(t - \Psi)$'s roots fall on integral multiples of $2\pi/(\alpha\beta)$. Let $r_k = 2k\pi/\beta$ denote the kth root of $h_\beta(t)$. Define β intervals η_k:

$$\eta_k = (r_k, r_k + \Psi) \; k = 0, 1, \cdots \beta - 1$$

The η_k represent the "sweep" of the zeros of h_β as Ψ varies. Notice that for t in one of the η_k, the signs of the shifted waveform $\nu_\Psi(t)$ and the unshifted $\nu(t)$ will be different, while they will be the same for t outside those intervals.

By a slight abuse of notation, let $\mathrm{sgn}(\eta_k)$ denote the sign of $\nu_\Psi(t)$ in the interval η_k. Since $\nu(t)$ and $\nu_\Psi(t)$ differ exactly in the η_k, the net feeding force will be changed by exactly their contributions. In other words, from equation (6.7), we get that:

Remark 1. *The change in feeding force is the sum of changes due to the intervals η_k, or:*

$$\bar{f}_\Psi = \bar{f}_0 + \frac{\mu m g}{2\pi} 2\Psi \sum_{k=0}^{\beta-1} \mathrm{sgn}(\eta_k)$$

where Ψ is included because it is the length of every interval η_k, and thus equal to the magnitude of the integral of the sign function over the η_k; and 2 is included because each interval of length Ψ contributes by its own sign, but also by reducing the integral of regions of the opposite sign.

We complete the proof by showing that only η_0 contributes to the sum. The rest cancel each other in symmetric pairs.

Now $h_\alpha(t)$ is symmetric in the interval $[0, 2\pi]$, i.e., $h_\alpha(t) = h_\alpha(2\pi - t)$, and in particular, $h_\alpha(r_k) = h_\alpha(r_{\beta-k})$. Furthermore, because $h_\alpha(t)$ has no roots within the η_k's, we can infer:

$$\operatorname{sgn}[h_\alpha(\eta_k)] = \operatorname{sgn}[h_\alpha(\eta_{\beta-k})],\ \forall k > 0 \tag{6.19}$$

$h_\beta(t)$ is also symmetric in $[0, 2\pi]$, but more importantly, its derivative is anti-symmetric. That is, $h'_\beta(t) = -h'_\beta(2\pi - t)$. This implies that $h_\beta(t)$'s zero-crossings at r_k and $r_{\beta-k}$ are in opposite directions, and thus:

$$\operatorname{sgn}[h_\beta(\eta_k)] = -\operatorname{sgn}[h_\beta(\eta_{\beta-k})], \forall k > 0 \tag{6.20}$$

Since $\operatorname{sgn}(\eta_k) = \operatorname{sgn}[h_\alpha(\eta_k)] \operatorname{sgn}[h_\beta(\eta_k)]$, the last two equations tell us that

$$\operatorname{sgn}[\eta_k] = -\operatorname{sgn}[\eta_{\beta-k}], \forall k > 0 \tag{6.21}$$

For β ranging from $0, \ldots, \beta - 1$, this means that only η_0 is missing a partner of opposite sign. All the other terms cancel and we have shown that only η_0 contributes to the sum in remark 1. Since $\operatorname{sgn}[\eta_0] = -1$, remark 1 simplifies to:

Remark 2. *The change in feeding force with Ψ is due entirely to η_0, and is equal to:*

$$\bar{f}_\Psi = \bar{f}_0 - \frac{\mu m g}{\pi}\Psi \tag{6.22}$$

These concepts are illustrated in figure 6.8.

Case 3. $\Psi = 2\pi/(\alpha\beta)$

Remark 3. *At $\Psi = 2\pi/(\alpha\beta)$ a new common root $t_k = 2k\pi/\alpha\beta$ is generated at some k, which is unique in the range $t \in [0, 2\pi]$.*

To show this let $t' = t\alpha\beta/(2\pi)$. So $\nu(t')$ has period $\alpha\beta$ and the roots of $h_\alpha(t')$ [resp. $h_\beta(t')$] are on integers $t = k_1\beta,\ k_1 = 0, 1, \cdots, \alpha$ [resp. $t = k_2\alpha,\ k_2 = 0, 1, \cdots, \beta$]. Conveniently, $h_\beta(t - 2\pi/\alpha\beta)$ becomes $h_\beta(t' - 1)$, so we need to show that $h_\alpha(t')$ has a root exactly one unit above a root of $h_\beta(t')$, i.e., a unique pair k_1, k_2 exists such that $k_1\beta = k_2\alpha + 1$.

Because α and β are co-prime, this is exactly Bézout's relation [28], which guarantees that a unique solution pair (k_1, k_2) exists modulo $\alpha\beta$. For example, if $\alpha = 3$, $\beta = 5$, $k_1 = 2$ and $k_2 = 3$ is the unique solution, i.e., the common root corresponding to a unit shift occurs at $t' = k_1\beta = 10$, Figure 6.8.

Remark 4. *At $\Psi = 2\pi/(\alpha\beta)$, $\nu_\Psi(t)$ is identical to $\nu(t)$ up to a sign flip and a shift by the common root t_k, i.e.:*

$$\nu_{(2\pi/\alpha\beta)}(t) = \pm\nu(t - t_k)$$

This is true because the factors $h_\alpha(t)$ and $h_\beta(t - 2\pi/\alpha\beta)$ are sinusoids. When shifted by a multiple of their root separation, sinusoids are identical up to a sign change. When a product of two sinusoids is shifted by a common root, the resulting function is also identical up to a sign change.

Equation (6.22) tells us that at $\Psi = 2\pi/\alpha\beta$, $\bar{f}_\Psi = \bar{f}_0 - 2\mu mg/\alpha\beta$, a non-zero change. Remark 4 tells us that $\nu_\Psi(t) = \pm\nu(t - t_k)$ and therefore $\bar{f}_\Psi = \pm\bar{f}_0$. But since there is a finite change, we cannot have $\bar{f}_\Psi = \bar{f}_0$, so we must have $\bar{f}_\Psi = -\bar{f}_0$. Their difference is $2\mu mg/\alpha\beta$, and so

$$\bar{f}_0 = \frac{\mu mg}{\alpha\beta} = \frac{\mu mg}{n_2^2 - n_1^2} \tag{6.23}$$

We can re-apply this argument to the next interval of $\Psi \in [2\pi/(\alpha\beta), 4\pi/(\alpha\beta)]$ and we would see a change in $\bar{f}$ in the opposite direction back to the original $\bar{f}_0$. Within each interval, $\bar{f}_\Psi$ varies linearly with Ψ, and so it generates a triangular waveform.

From Equation (6.18) we see that $\Psi = 2\pi/(\alpha\beta) = 2\pi/(n_2^2 - n_1^2)$ corresponds to $\varphi = -\pi/(n_1 n_2)$. Therefore, for $\varphi \in (0, 2\pi)$, $\bar{f}(\varphi)$ will hit $\pm\bar{f}_0$ $2n_1n_2$ times. Because the change in $\bar{f}$ is linear on Ψ, φ, Equation (6.22), and because the peaks alternate, $\bar{f}(\varphi)$ must also be a triangular waveform whose period is 2π divided by half the number of peaks, i.e., its period is $2\pi/(n_1n_2)$, and of amplitude given by the peak value, Equation (6.23). □

The shape of $\bar{f}$ as a function of φ for various n_1, n_2 combinations is illustrated in Figure 6.9.

Corollary 1. *The choice $n_1 = 1$, $n_2 = 2$, and $\psi = 0$ yields the highest possible $\bar{f} = \mu mg/3$.*

Proof. At $\psi = 0$, the triangular waveform is at a peak. We know $n_2 \geq n_1 + 1$. Therefore the denominator of $\bar{f}$ is:

$$n_2^2 - n_1^2 \geq (n_1 + 1)^2 - n_1^2 = 2n_1 + 1$$

And this bound is attained (i.e., $\bar{f}$ is maximized) when $n_2 = n_1 + 1$; the global minimum of this expression (and the global maximum of $\bar{f}$) occurs when $n_1 = 1$ and $n_2 = 2$, yielding $\bar{f} = \mu mg/3$. □

In practice, we typically use $\nu(t) = \cos(t) - \cos(2t)/2$ ($b = 1/2$); though this yields $\bar{f} \cong 0.24\mu mg$ (lower than the $b = 1$ case) it also yields a higher equilibrium velocity.[2]

Corollary 2. *A sum of harmonics n_1, n_2 (exactly one of which is odd) produces zero (resp. maximum) feeding force if they have at least one root (resp. peak) aligned.*

[2]The part's equilibrium velocity ν_p is such that $\mathrm{sgn}[\nu(t) - \nu_p]$ integrates to zero in 2π.

Proof. When at least one root (resp. peak) is aligned, the sum waveform is equivalent (modulo a phase shift and a sign flip) to a sum of pure sines (resp. cosines), which yields zero (resp. maximum) feeding force, Lemma 2 (resp. Theorem 1). ☐

6.5 Summary

A new local motion primitive called the "jet" has been described which makes parallel manipulation with the UPM fast and robust, overcoming many of the shortcomings present in a previous method. Optimum jet components were chosen based on a complete characterization of feeding forces when plate motion is a sum of two sinusoids. Due to its mechanical simplicity, the UPM is an attractive technology for existing industrial applications such as part feeding, sorting, singulation, etc. The "open face" nature of its workspace suggests interesting applications in novel areas such as product display, interactive toys, and active desks.

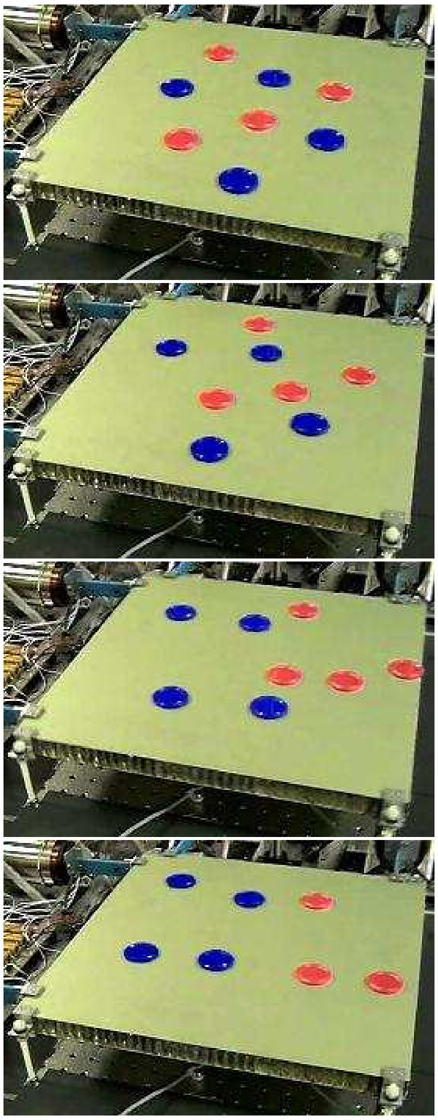


Figure 6.6: The sorting experiment: eight plastic poker chips (4 dark/blue, and 4 light/red) initially scattered randomly over the device (top) are sorted by color to opposite sides of the plate. In the last frame, one of the light chips has fallen the right edge of the table. All chips get sorted in about 30 seconds.

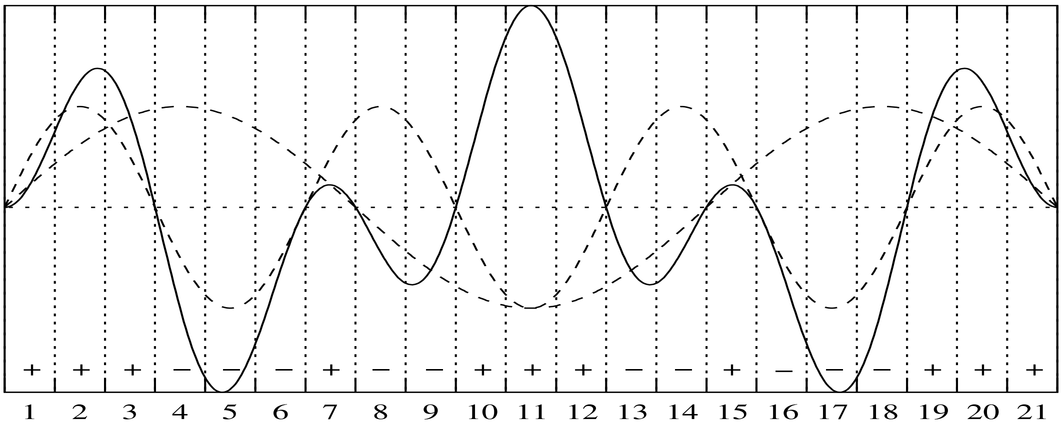


Figure 6.7: Sign imbalance for the case $n_1 = 2$ and $n_2 = 5$, i.e., $\alpha = 3$, $\beta = 7$. $\nu(t) = cos(2t) - cos(5t)$ is plotted as a solid curve over the $[0, 2\pi)$ interval. Its two factors: $sin(\frac{3}{2}t)$, and $sin(\frac{7}{2}t)$, are plotted as dashed curves. $\mathrm{lcm}(\alpha, \beta) = \alpha\beta = 21$ yields the number of equal-length sub-intervals of $[0, 2\pi]$ which preserve the sign of $\nu(t)$; in the above, intervals are numbered and identified with a "+" or "-", according to ν's sign in that interval. Because $\alpha\beta$ is odd, the number of positive- and negative-sign intervals must differ: above one counts 11 positive vs. 10 negative intervals. This implies $\bar{f} \neq 0$.

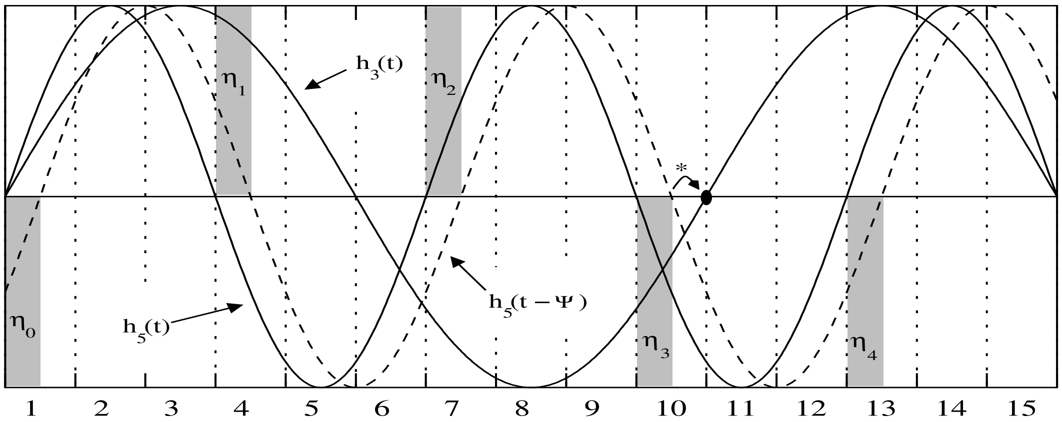


Figure 6.8: Force cancellation for $\alpha = 3$, and $\beta = 5$ (15 constant-sign intervals). The two solid curves are $h_3(t) = \sin(3t/2)$ and $h_5(t) = \sin(5t/2)$. The dashed curve is $h_5(t - \Psi)$, where Ψ is $\pi/(\alpha\beta)$, i.e., half a basic interval. The η_k intervals appear shaded above (resp. below) the x-axis depending on the sign of $h_\alpha(t)h_\beta(t - \Psi)$. Each η_k cancels with $\eta_{\beta-k}$, with the exception of η_0; so the canceling pairs are (η_1, η_4), (η_2, η_3). As Ψ approaches a full interval's length, a new common root is generated at the end of the 10th interval, marked with a "*".

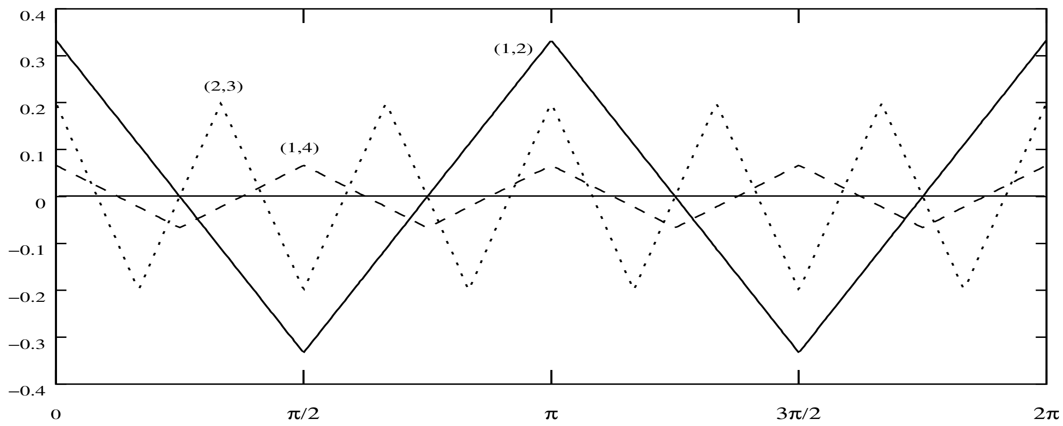


Figure 6.9: Plot of $\bar{f}/(\mu m g)$ versus ψ for $(n_1, n_2) = \{(1,2),(2,3),(1,4)\}$. The points plotted were obtained through numeric integration. As predicted, the function is a triangular waveform of period $2\pi/(n_1 n_2)$ and of amplitude $1/(n_2^2 - n_1^2)$.

Chapter 7

Conclusion

A novel technology for planar part manipulation has been presented based on a grip-perless, mechanically-simple device, called the Universal Planar Manipulator (UPM). Our main contribution has been to show that the UPM, despite its minimalism (3 degrees of freedom) can manipulate several objects in parallel. This is due to the fact that Coulomb friction is non-linear on sliding velocity, which implies that the space of rotation friction fields is not closed under addition. Two control algorithms have been presented: one based on a sequence of rotations and another based on a local primitive called the jet. We expect this technology, due to its simplicity to be appealing to existing industries (automation, food handling, etc.) and to inspire novel devices such as toys and interactive desks.

Future work with the UPM will include (a) the manipulation of generic objects such as tools, bottles, and books, which will require more sophisticated vision software and the ability to rotate parts, e.g., by applying localized force couples; (b) manipulation in other non-linear force-generation environments such as magnetic fields and fluids; (c) redesign/integration of actuators to reduce the UPM's total cost and weight: voice coils could be replaced by either piezo or polymeric (artificial muscle) actuators.

Bibliography

[1] Analog Devices Catalog. Winter 1999 short form designer's guide, 1999.

[2] E. Avallone and T. Baumeister III. *Marks' Standard Handbook fo Mechanical Engineers*. McGraw-Hill, New York, NY, 10th edition, 1996.

[3] BEI Kimco Magnetic Systems. Voice coil actuators, an applications guide, 1999. Information booklet.

[4] D. Berkowitz and J. Canny. Designing parts feeders using dynamic simulation. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

[5] D. Berkowitz and J. Canny. A comparison of real and simulated designs for vibratory parts feeding. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[6] 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.

[7] K. Böhringer and H. Choset. *Distributed Manipulation*. Kluwer Academic Publishers, Norwell, MA, 2000.

[8] K. Böhringer, B. Donald, and N. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. In *IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, San Diego, CA, February 1996.

[9] 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.

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

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

[12] 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.

[13] M. Cohn, C. Kim, and A. Pisano. Self-assembling electrical networks: an application of micromachining technology. In *IEEE Micro-electromechanical Systems*, 1991.

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

[15] P. Frei and M. Wiesendanger. 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, May 1999.

[16] L. Gibson and M. Ashby. *Cellular Solids: Structure and Properties*. Cambridge University Press, Cambridge, MA, 1999.

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

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

[19] 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.

[20] 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.

[21] J. Krim. Friction at the atomic scale. *Scientific American*, pages 74–80, October 1996.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] Microchip Catalog. Pic16/17 microcontroller data book, 1996.

[28] M. Mignotte. *Mathematics for Computer Algebra*. Springer-Verlag, New York, NY, 1992.

[29] B. Mirtich and J. Canny. Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, New York, NY, 1995. ACM Press.

[30] B. Mirtich, Y. Zhuang, K. Goldberg, J. Craig, R. Zanutta, B. Carlisle, and J. Canny. Estimating pose statistics for robotic part feeders. In *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, April 1996.

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

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

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

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

[35] R. Resnick and D. Halliday. *Physics, part I*. John Wiley & Sons, Inc., New York, NY, 11th edition, 1987.

[36] D. Reznik, S. Brown, and J. Canny. Dynamic simulation as a design tool for a microactuator array. In *IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.

[37] 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.

[38] 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.

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

[40] 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.

[41] S. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach*. Prentice Hall, New Jersey, NJ, 1995.

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

[43] 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.

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

[45] W. Zesch, R. Büchi, and R. Siegwart. Inertial mechanisms for positioning microobjects: two novel mechanisms. In *SPIE conference on microrobots and micromechanical systems*, pages 80–88, Philadelphia, PA, October 1995.