Parts Handling and Self-Assembly Using Stable Limit Sets

T.D. Murphey
Electrical and Computer Engineering
University of Colorado
Boulder, CO 80304
murphey@colorado.edu
K.M. Lynch
Mechanical Engineering
Northwestern University
Evanston, IL 60208
kmlynch@northwestern.edu

***Abstract*— Vibration is commonly used in industrial parts feeding and alignment processes, and to provide energy to encourage mobility among self-assembling parts. We are studying a simple model of agitation where planar parts are repetitively thrown by a simple one-degree-of-freedom throwing surface, caught, and allowed to settle. Throwing actions result in nonlinear discrete-time maps in the parts' configuration space, exhibiting behaviors such as unique fixed points and uncertainty-reducing forward limit sets with large basins of attraction. We show how to shape these maps by choosing the arm geometry, throwing velocity, and mass parameters of the parts. In some cases, we can design a single map that is guaranteed to uniquely position and orient a part. In other cases, we can design multiple maps, corresponding to different throw velocities, such that the composition of the maps can be used to drive multiple parts to a desired assembly. Switching between the throw actions is triggered by simple sensors that recognize when the system has achieved a configuration in the basin of attraction of a subsequent map.**

***Index Terms*— parts handling, self-assembly, stable limit sets, sensorless manipulation, throwing and catching.**

I. Introduction

Traditional robot assembly relies on a monolithic robot to sequentially assemble individual parts into a growing assembly. With an eye toward future micro- and nano-scale devices consisting of large numbers of parts, *self-assembly* aims to distribute the assembly program, typically concentrated in the powerful robot manipulating the "dumb" parts, to the parts themselves. While a great deal of work in self-assembly focuses solely on designing parts so that they only bond in ways consistent with a desired static assembly, the design space of self-assembling systems can include the following dimensions:

1) *Part physical characteristics* such as shape, mass properties, friction, restitution, wetability, charge, chemical, and other properties.
2) *Local sensing, communication, computation, and actuation* by the parts, creating interaction rules beyond those that can be implemented by the part physical characteristics alone.
3) *Properties of the environment* such as templates, fixtures, shaped force fields, and energy input such as heating or agitation.
4) *Centralized feedback control* where the actuation of the environment is based on the sensed state of the assembly process.

Most work on small-scale self-assembling systems focuses on item 1. Ongoing efforts to develop "smart" parts are aimed at taking advantage of the capabilities in item 2 to achieve more flexibility in the self-assembled structures. Studies in *pattern formation* in granular flows attempt to develop explanations for the effects of characteristics in items 1 and 3 on the spontaneous formation of patterns in agitated heterogeneous granular media. *Sensorless robot manipulation* focuses on designing fixtures and open-loop motions of manipulators to sort or feed parts (item 3), while one goal of *minimalist robotics* is to design the simplest sensors necessary to create useful feedback controllers (item 4) [15]. One assumed property of centralized feedback control (item 4) is that the sensory information and control authority is limited; otherwise the environment could directly and independently actuate the parts, and we would be back to the traditional model of robot assembly. Even though the centralized control authority is limited, it can play a useful role. Consider, for example, a sensor-based annealing pattern, where the environment is actuated with high energy when the parts are far from forming an assembly, but with lower energy as the assembly forms. We consider just such an example in this paper.

Just as traditional assembly sequences can be planned automatically, ideally we would be able to automatically distribute the "assembly program" for a desired assembly into the characteristics of the parts, the environment, and the centralized feedback controller. This is far beyond our current capabilities. To begin to gain insight into the tradeoffs offered by the design dimensions described above, in this paper we study simple examples of self-assembly combining the design of part physical characteristics (item 1), properties of the environment (item 3), and centralized feedback (item 4).

Our interest in this paper is in the assembly of rigid parts, mesoscale or larger, under low-degree-of-freedom external forcing of the parts' support surface. In short, the goal is to literally throw together an assembly: to design the properties of the parts, and the geometry and motion of the parts' support

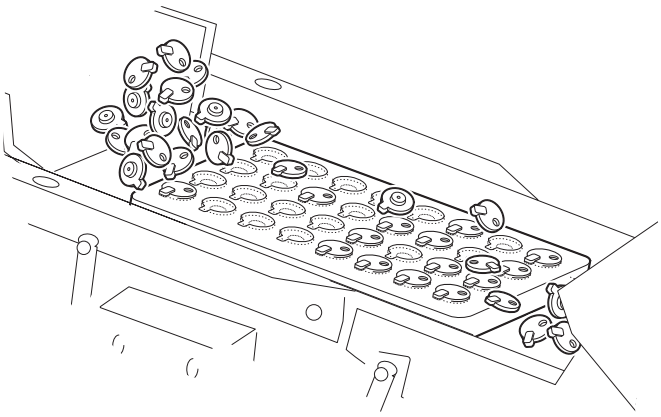


Fig. 1. The Sony APOS parts orienting system (from [34]).

surface, such that there is a single, globally attractive fixed point of the dynamical system — the desired assembly. The idea is related to one considered by Penrose and Penrose in 1957 [41], who showed that multiple copies of two specially-designed complementary planar parts on a horizontal track "assemble" into linked pairs under horizontal shaking of the track (but only if the system is first seeded with one or more assemblies).

The problem of throwing together an assembly is challenging, particularly considering that the only "binding" forces between parts are due to local potential wells induced by gravity and complementary part-part or part-environment geometries. Nonetheless, such a system is not completely without precedent. The commercial Sony APOS parts orienting system (Fig. 1) produces a tray of oriented parts by a simple agitation strategy [24]. Parts wash over the vibrating tray, and the tray has part-shaped depressions such that parts that fall into the depressions in the right orientation stay there, and those that fall in the wrong orientation pop back out. These parts continue into a return bin and then are dumped over the tray again. This process continues several times, with no sensing, and the result is a tray of oriented parts (with perhaps a small number of empty depressions). This is a kind of templated self-assembly. The design problem is to find an appropriate depression shape and vibration profile for the given part. Currently this problem is solved by experimental trial-and-error.

A. The Example System

Consider the following scenario. Two rigid planar polygonal parts in a gravitational field rest on a support surface. The support surface is a one-degree-of-freedom arm that throws the parts, lets them settle to a new configuration, then repeats. The system is a driven, dissipative, nonlinear dynamical system, with dynamics determined by repeated impacts, friction, and gravity. The goal is to bring the two parts into a desired relative configuration (i.e., an assembly).

Although this system is simple compared to many existing industrial feeding devices, the dynamics are highly complex, and it is difficult to apply the model-based rational mechanics approach, which has had significant success in kinematic, static, and quasistatic manipulation planning (see, e.g., [34]). This approach is based on precise knowledge of part geometries, mass properties, and friction and restitution coefficients, and uses Newton's laws, Coulomb friction, and various models of impact. For systems of many interacting parts, though, this level of modeling may be too fine. Just as Newton's laws are more useful than quantum mechanics for understanding rigid body dynamics, a higher level of modeling may be more appropriate in the design of self-assembling systems.

This issue was addressed by Berkowitz and Canny [5], Mirtich *et al.* [36], Gudmunddson and Goldberg [19], and Moll and Erdmann [37], who suggested methodologies for parts feeder design based on statistics accumulated during actual experiments or simulations. Because of the complexity of the dynamics, the feeders are treated as black boxes, and the behavior of the system is modeled simply by statistics. This is a high-level description of the system dynamics, with limited power to suggest changes in design to achieve desired behavior.

In this paper we look for an intermediate level of modeling, more abstract than the low-level rational mechanics model but more useful for design than input-output statistics. In the present work and in [39], [40], our approach is to use qualitative features of the induced dynamical system, such as ergodicity or the existence of limit sets or fixed points. Each throw and catch of the parts induces a discrete-time *throw map* that maps the initial resting configurations of the parts to new resting configurations. Although simulation or experiment may be required to get the details of a throw map, the existence of certain qualitative features of the map is robust, such as a small number (perhaps just one) of entropy-reducing [52] forward limit sets for all possible initial configurations of the parts. In this paper, we show that it is sometimes possible to create throw maps with unique stable fixed points, with large basins of attraction, by designing the mass properties of the parts (item 1) and the throwing velocity and geometry of the arm (item 3). When this is not possible, we can design multiple throw maps, differing only by the throwing velocity, where each map can be viewed as a "funnel" that collapses a relatively large basin of attraction into a smaller set in the configuration space. This results in a partial ordering of funnel maps, where the "spout" of a funnel is inside the "mouth" (basin of attraction) of a funnel of higher priority, and the highest priority funnels have spouts consisting of the desired assembled state. Sensors (item 4) need only recognize when the system is in the basin of attraction of a higher-priority funnel, allowing for the design of very simple sensors [15]. This view of sensor-based composition of dynamical systems is explored in detail in [11].

Figure 2 gives a conceptual example in a one-dimensional configuration space. In this example, there are three throw maps: one low-priority high-energy map (not shown) that randomizes the configuration; one medium-priority map (Figure 2 (left)) that collapses a large volume of the configuration space to a forward limit set consisting of a period-two orbit;

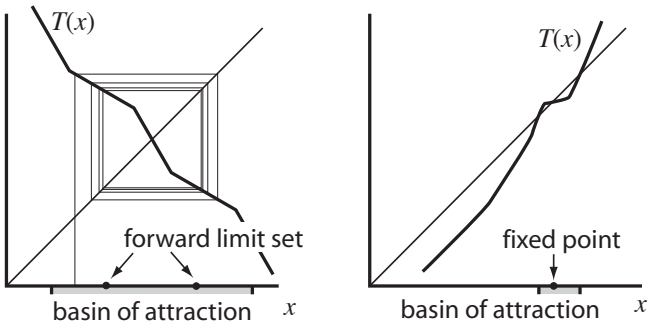


Fig. 2. Conceptual one-dimensional discrete-time throw maps. Left: A medium-priority map from the initial part state x to the state $T(x)$ after a throw and catch. This map yields a forward limit set consisting of a period two orbit with a basin of attraction as indicated. Right: A high-priority map with a stable fixed point with the basin of attraction indicated.

and one high-priority low-energy map (Figure 2 (right)) that collapses a small volume of the configuration space to a single fixed point (the desired assembly). This last funnel map is designed so that one of the points of the period-two spout of the medium-priority map is in its mouth. In this case, sensors need only recognize when the system configuration is in the basins of attraction of the medium-priority and high-priority funnel maps. Provided the randomization map is guaranteed to eventually bring the system in the basin of attraction of the medium-priority map's basin of attraction, assembly is guaranteed. (See [14] for more on the utility of randomization in robot manipulation.)

This simple example avoids two basic complications of the assembly problems we are interested in: (1) the composite configuration space for multiple parts may be high dimensional, and (2) real throw maps are effectively stochastic at the resolution with which the system can be sensed and controlled. The second issue argues for stochastic representations and estimates on the likelihood of assembly as a function of time. This does not affect the essential funneling behavior, however, even if it is expressed in terms of funneling probability mass. The more serious issue is the first. Higher-dimensional configuration spaces allow for significantly more complex dynamical behavior, including chaotic behavior, strange attractors, and a number of coexisting funnels with mouths and spouts of different dimensions within a single throw map. Our approach has been to focus on small numbers of parts (one or two) and low-energy throws, where these behaviors are rarely observed. High-energy throws are used simply for randomization. The relatively low dimension of our configuration spaces means that the mouths of some low-energy funnel maps have significant measure in the parts' resting configuration space, so that randomized actions are likely to find these basins of attraction.

B. Contributions and Overview

The work in this paper builds on [31], where it was observed that a single planar part, subjected to repeated identical throws by a one-joint arm, will enter a unique forward limit set from almost any initial configuration on the arm. The specific contributions of the current paper are:

- *Constructive controllability of a single part.* We begin by showing that a single polygonal part can be repetitively thrown to reach a desired position and orientation on the arm. This controllability result is suggestive of the feasibility of subsequent assembly tasks.
- *Design of a part and fixture to achieve a globally attractive fixed point.* By modifying the mass properties of the part (item 1) and adding a fixture to the arm (item 3), we can transform the forward limit set of a single throw map into an essentially globally-attractive fixed point.
- *Sensor-based self-assembly of two parts.* By introducing feedback with a simple one-bit sensor (item 4), we switch between a low-priority high-energy randomization throw map and a high-priority low-energy assembly throw map to achieve global "self-assembly" of two triangular parts into a rectangle.
- *An intermediate level of modeling.* We combine low-level rational mechanics modeling with an intermediate level of modeling based on robust qualitative features of the throw maps.

In Section III we describe the experimental system, establish notation, and review a previous experiment in repetitive throwing and catching of a single part that led to the experiments in this paper. In Section IV we derive constructive controllability conditions for our sample system and use these to derive feedback throwing controller to position a part on the arm using only two one-bit sensors. Section V describes the techniques used to design the forward limit set behavior of a polygonal part. By the addition of a fixture, the globally-attractive forward limit set can be turned into a globally-attractive fixed point of the configuration of the part. In Section VI, we demonstrate the assembly of two triangular parts into a rectangular part in a potential well created by a fixture. The assembly process consists of two throwing actions: a high-energy low-priority throwing motion that randomizes the configuration of the parts, and a low-energy high-priority throwing motion that induces the two parts to assemble. A simple sensor is used to detect when the parts are in the basin of attraction of the assembly under the low-energy throwing motion. Conclusions are given in Section VII.

II. Background: Sensorless Assembly and Self-Assembly

***** We should work these reference into the introduction and eliminate this section.**

Despite the many advances in recent years in understanding the mechanics of sensorless manipulation and the computational aspects of motion planning (at the macro [21], [29], [38] and micro [17] scales), we are still not very close to understanding manipulation systems that manipulate large numbers of parts, such as the Sony APOS system [24]. This is largely because the complications associated with impact, friction, dynamics, elasticity, and other effects make

simulation difficult and direct analysis nearly impossible in all but the simplest of cases. Assumptions, such as flat surfaces, minimal impact, zero friction, and others, are typically employed to make the analysis more tractable. However, this to a large extent invalidates the rational mechanics approach to understanding these systems, as there is little formal connection between the simplified systems and their more complex relatives, either in the analysis or in the simulations. Researchers have already recognized and addressed some of these obstacles [2]–[6], [12], [16], [18], [36], [37], [42], [43], [48], [58].

Self-assembly has recently become an area of intense study, particularly as it relates to micro- and nano-scale systems. In such systems, the program that encourages the system to self-assemble must in one way or another be encoded into the dynamics of the system. This includes biological and molecular self-assembly [20], [30], [56], [59], where self-assembly is seen as a minimum energy problem. These minimum energy configurations are caused by capillary forces, Van der Waals potentials, chemical bonds, electrostatic forces, etc. The ideas discovered in this context have been extended to other systems, such as magnetically induced self-assembly [25] and wetability induced self-assembly [7], [49], [50], [55], [57]. More recently, there has been progress on designing self-assembling systems using graph grammars [27], [28].

Other work has focused largely on *pattern formation*. These are largely scientific and phenomenological studies, interested in understanding the mechanisms behind the patterns that arise under different environmental forcing, rather than how to explicitly produce a particular kind of pattern. However, these are extremely valuable in that they investigate how the environment and external forcing affects the evolution of a given pattern. Good examples include investigations into pattern formation and segregation in mixing of granular materials in rotating drums of various geometries [8], [13], [22], [23], [26], [35], [44], [47], [53], [60].

The work presented here falls at the intersection of several of these areas, having characteristics of classical sensorless manipulation, self-assembly, and pattern formation. In particular, the manipulation is intrinsically dynamic (i.e., no quasi-static description can capture the behavior of the system), sensorless or with only minimal sensing, and limit sets (e.g, patterns) are the useful description of the system. These limit sets can be designed to follow a set of rules that can be used for the purpose of self-assembly. In particular, limit sets can be designed so that almost all configurations "reject" assembly and the desired configuration "accepts" assembly.

Analyzing manipulation in terms of limit sets is not new. In particular, *juggling* involves stabilizing a cyclic trajectory for an object [1], [32], [46], [51], [54]. Koditschek and colleagues studied two- and three-dimensional juggling by batting in a series of papers [9], [10], [45]. both in two and three dimensions. Rather than batting, our system "throws" and "catches" the polygons, but the philosophy is similar.

III. Background: Limit Sets in Manipulation

The Flatland testbed for assembling planar parts by throwing and catching is shown in Fig. 3. The Flatland setup is

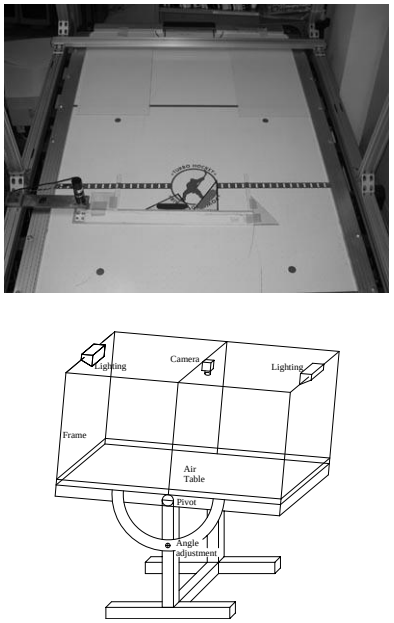


Fig. 3. The Flatland experimental system consists of an air hockey table used as an air bearing, a single-degree-of-freedom robotic arm that can throw parts, and a camera used for data acquisition.

composed of a large air hockey table that supports objects with a nearly frictionless air bearing. The angle of the table is adjustable, providing control of the effective gravity in the support plane. Parts rest on a 1DOF rotary arm, and the motion of the arm is controlled to throw the parts. The parts then settle on the stationary arm. The arm is covered in a thin layer of high-friction slow-recovery foam (Lendell Mfg, type PHS-14) and is driven by a 6W Harmonic Drive RH-8 3006 gearmotor. The air table supports an extruded 80/20 aluminum superstructure on which lights and camera are mounted. A Cognachrome vision board calculates the positions and orientations of parts on the table for purposes of data acquisition.

An initial experiment by Lynch, Northrop, and Pan [31] studied impulsive throwing and catching of a single polygonal part (Fig. 4). The "throw map" maps an initial configuration of the part to a final configuration after the part settles. The configuration space is $\mathbb{R}^+ \times \mathbb{N}\mathrm{mod}(n)$, where $\mathbb{N}\mathrm{mod}(n)$ indicates the set of n stable sides the polygon can come to rest on and $\mathbb{R}^+$ indicates the distance ("radius") along the arm from the joint. When the arm repetitively executes identical impulsive throwing motions, it was shown that, for some arm geometries and throwing impulses, the part eventually enters the same limit set of resting configurations regardless of its initial configuration. The cyclic pattern consists of "jogs" (small translational motions away from the joint while the part remains in the same orientation) and "flips" (counterclockwise rotation that additionally moves the part back toward the arm joint, seen in Fig. 4). A globally attractive forward limit set for a triangular part can be seen in Fig. 5. With the resting sides of the triangle labeled $0, 1, 2$ in the clockwise direction, the limit set consists of outward jogs on side 0 until the part hits a critical radius at which it flips onto side 1. Then it jogs

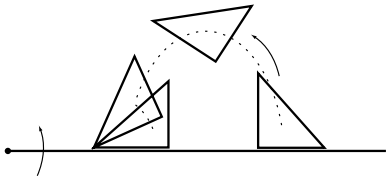


Fig. 4. A part is thrown by an impulse from the rotary arm, then settles back to rest.

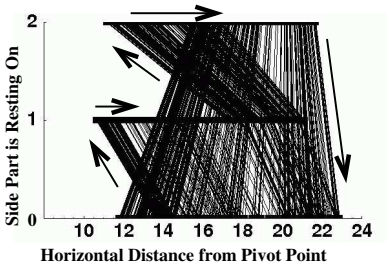


Fig. 5. A simulated forward limit set for the thrower from Fig. 4. The arrows indicate the motion of the part and its transitions from one side to the next.

until it flips to side 2. Finally, it jogs outward until it flips back to side 0, where the cycle repeats.

To understand this behavior, and to set the stage for our new experiments, we describe the equations of motion for this system.

A. Notation

We define an inertial xy coordinate frame at the axis of the one-degree-of-freedom throwing arm, where gravity acts in the $-y$ direction (Figure 6). The throwing arm geometry is described by the single parameter h, which gives the height of the throwing surface ($y = h$) when the arm is horizontal. We choose $h < 0$ to achieve desired limit set behaviors.

A part is a convex polygon, and its configuration at rest on the arm is given by the side i on which it rests and the x coordinate of its center of mass, denoted R. Thus a resting configuration of the part is denoted (R, i) or (x, i). While resting on edge i on the horizontal arm, the height of the center of mass above the support surface is $d^i > 0$ and the vectors from the center of mass to the left and right support vertices are $(u^i_L, v^i_L < 0)$ and $(u^i_R, v^i_R < 0)$, respectively. (Note $d^i = -v^i_L = -v^i_R$.) The side is *stable* in gravity if $u^i_L < 0$ and $u^i_R > 0$. The part is assumed to have n stable sides, so that its resting configuration space is $\mathbb{R}^+ \times \mathbb{N} \bmod(n)$. The mass of the part is m and its radius of gyration is ρ.

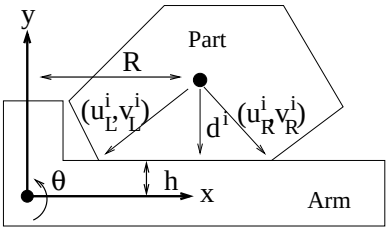


Fig. 6. Part parameters. ***** Maybe replace this figure with one that shows h having the right sign, d^i as positive, and variables in the right fonts.**

B. Flight Equations

The arm throws the part with an impulse—it releases the part at an angular velocity of $\dot\theta_r > 0$ with negligible displacement of the arm and no slipping of the part. During flight, the configuration of the part is written $[x(t), y(t), \phi(t)]$, where $t = 0$ at release and the angle ϕ is measured with respect to the initial orientation of the part. The (integrated) equations of motion of the part in flight are

$$\begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} R - (h + d^i)\dot\theta_r t \\ h + d^i + R\dot\theta_r t + \frac{1}{2} g t^2 \\ \dot\theta_r t \end{bmatrix}. \tag{1}$$

The location of the (initially bottom left) vertex can be written

$$\begin{bmatrix} V_x(t) \\ V_y(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} \cos(\phi(t)) & -\sin(\phi(t)) \\ \sin(\phi(t)) & \cos(\phi(t)) \end{bmatrix} \begin{bmatrix} u^i_L \\ v^i_L \end{bmatrix}. \tag{2}$$

Substituting Eq. (1) into Eq. (2) and knowing (by assumption) that the time of impact will be when the left bottom vertex hits the horizontal arm ($V_y(t) = h$), we solve

$$V_y(t) = h + d^i + R\dot\theta_r t + \frac{1}{2} g t^2 + \sin(\dot\theta_r t) u^i_L + \cos(\dot\theta_r t) v^i_L = h \tag{3}$$

for the smallest positive value of t to get the flight time t_{impact}. Other possible solutions to this transcendental equation are discarded.

C. Impact Equations

When the part impacts with the arm after a throw, we assume that friction is high enough and restitution is low enough that the part does not slide or bounce. This is approximately realized in experiment by the thin layer of foam covering the arm. Let the part pre-impact velocity at t_{impact} be $[\dot x^-, \dot y^-, \dot\phi^-]$, the post-impact velocity be $[\dot x^+, \dot y^+, \dot\phi^+]$, and $[r_x, r_y]$ be the vector from the center of mass to the vertex that impacts the arm. By our no-slip no-rebound impact assumption, the post-impact velocity is

$$\begin{bmatrix} \dot x^+ \\ \dot y^+ \\ \dot\phi^+ \end{bmatrix} = \frac{1}{\rho^2 + r_x^2 + r_y^2} \begin{bmatrix} r_y(\rho^2 \dot\phi^- + r_y \dot x^- - r_x \dot y^-) \\ r_x(-\rho^2 \dot\phi^- - r_y \dot x^- + r_x \dot y^-) \\ \rho^2 \dot\phi^- + r_y \dot x^- - r_x \dot y^- \end{bmatrix} \tag{4}$$

(see [31] for details).

By assumption, after impact the part comes to rest on either the initial stable side i (a jog) or on the next stable side $i + 1 \bmod(n)$ (a flip). In the space of orientations of the part, there is a potential energy peak at the critical angle where the part balances unstably on a vertex between falling toward side i or toward side $i + 1$. To determine whether the part has sufficient post-impact energy to cross this peak, we evaluate its post-impact potential energy $-m r_y$ and its post-impact kinetic energy $\frac{1}{2} m((\dot x^+)^2 + (\dot y^+)^2 + (\rho \dot\phi^+)^2)$. Depending on whether the part has enough energy to cross the peak, the part rolls without slipping to rest on its new stable side, and the new x coordinate of the part is calculated accordingly.

D. Throw Map

The composition of the flight and impact equations yields the *throw map*. Given an initial rest configuration (x_0, i_0), the new configuration after a throw with angular velocity $\dot{\theta}_r$ is written

$$(x_f, i_f) = T^{\dot{\theta}_r}(x_0, i_0)$$

or simply $T(x_0, i_0)$ if the release velocity is implicit. If $\dot{\theta}_r$ is small, the part jogs to a new position with $i_f = i_0$. If the throw energy is large, the part flips to a new side $i_f = (i+1) \mod(n)$.

The result of a sequence of k identical throws at angular velocity $\dot{\theta}_r$ is written

$$(T^{\dot{\theta}_r})^k(x_0, i_0)$$

and the result of a sequence of k different throws, $\{\dot{\theta}_r^1, \ldots, \dot{\theta}_r^k\}$, is written compactly as

$$T^{\dot{\theta}_r^k} \circ \ldots \circ T^{\dot{\theta}_r^1}(x_0, i_0) = \prod_{j=1}^{k} T^{\dot{\theta}_r^j}(x_0, i_0).$$

To discuss the result of a sequence of identical throws, we use the following definition.

Definition 3.1: Consider a map $b : Z \to Z$. The map b has a *forward limit point* $q \in Z$ of $z \in Z$ if there exists exists a sequence of positive integers $N_i, i \in \mathbb{N}$ such that $b^{N_i}(z)$ approaches q as $i \to \infty$. The *forward limit set* for z is the set of all forward limit points of z.

It was shown in [31], both theoretically and experimentally, that a part will enter a *unique* uncertainty-reducing forward limit set S under the repeated action of a throw map T for essentially all initial part configurations, given an appropriately chosen h $(0 > h > -\min_i d^i)$ and $\dot{\theta}_r > 0$. By choosing $h < 0$, jogs are guaranteed to be outward (x coordinate increasing), as the initial velocity of the left support vertex has an outward component. By choosing $h > -\min_i d^i$, the velocity of the center of mass during flight is inward, guaranteeing that flips will result in a decrease in the x coordinate. The result is that parts jog out and eventually flip inward, creating a unique forward limit set such as that illustrated in Fig. 5.

We define the unique forward limit set S associated with a throw velocity $\dot{\theta}_r$ as $S(\dot{\theta}_r)$, where the arm geometry and part properties are implicit. Where there is no possibility for ambiguity, we simply write S. The limit set S is bounded from above and below in the $\mathbb{R}^+$ component. It is usefully characterized by the bounds $\sup S$ and $\inf S$, defined by

$$\begin{aligned} \sup S &= \left\{ (r_i) \;\middle|\; \begin{array}{l} x < r_i \Rightarrow T(x,i) = (x + \delta x, i) \\ x \geq r_i \Rightarrow T(x,i) = (*, (i+1) \mod(n)) \end{array} \right\} \\ \inf S &= \{ (l_i) \,|\, x < l_i \Rightarrow (x,i) \notin S \} \end{aligned}$$

where δx is some positive number and $*$ is arbitrary. Hence, both $\sup S$ and $\inf S$ consist of n numbers for a part with n stable sides. Any throw $T(x,i)$, where $l_i < x < r_i$, results in an outward jog (the part lands on the initial edge) while any throw with $x \geq r_i$ results in a flip (the part flips to a new edge).

Beginning from this, we show how to alter the mass properties of the part to change the shape of the forward limit set, and how to add a fixture to change a globally-attractive limit set into a globally-attractive fixed point. Before doing this, however, we establish the constructive controllability of the position and orientation of a part by throws.

IV. Constructive Controllability for a Single Part

Our goal is to show that, for a fixed arm geometry h $(0 > h > -\min_i d^i)$, there exists a sequence of throws to take the part from any initial resting configuration to a desired final resting configuration (x_f, i_f) for any i_f and for x_f sufficiently large. The outline of the proof is as follows. First we show that for a constant throwing velocity $\dot{\theta}_r$, the part will eventually achieve a configuration $(x, i_f) \in S(\dot{\theta}_r)$, where $x < x_f > l_{i_f}$. Once the part has reached such a configuration, there exists a sequence of throws that jog it to the desired position (x_f, i_f). To implement a simple approximation to the control, we require two one-bit sensors: one to tell us when the part has reached a configuration $(x < x_f, i_f)$ and another to tell us when it has approximately (within ϵ) reached its destination, where ϵ can be chosen arbitrarily small.

Lemma 4.1: For a given $\dot{\theta}_r$ and corresponding limit set $S(\dot{\theta}_r)$, and any (x_0, i_0), for all i_f there exists a k such that $(T^{\dot{\theta}_r})^k(x_0, i_0) = (x, i_f) \in S(\dot{\theta}_r)$.

Proof: Note that in Eqs. (1) and (3), $x(t_{\text{impact}})$ can be determined by first solving Eq. (3) for t_{impact} (the time the part hits the arm) and then substituting into Eq. (1). We can approximately solve this by the small-angle approximation replacing $\sin(\dot{\theta}_r t_{\text{impact}}) u_L^{i_0}$ with $\dot{\theta}_r t_{\text{impact}} u_L^{i_0}$ and replacing $\cos(\dot{\theta}_r t_{\text{impact}}) v_L^{i_0}$ with $v_L^{i_0}$. Doing so, we get that

$$x(t_{\text{impact}}) = R + \frac{2}{g}(h - v_L^{i_0})(R + u_L^{i_0})\dot{\theta}_r^2.$$

This implies that $x(t_{\text{impact}})$ depends linearly on R and quadratically on $\dot{\theta}_r$. This, along with the fact that $x > 0$, implies that for any x_0 and throw k, $x_{k+1} - x_k > x_k - x_{k-1}$. Therefore, every throw increases x by some minimum amount $\delta > 0$. This implies that there are no equilibria (as was also shown in [31]) before reaching the critical r_{i_0} where it flips from side i_0 to side $i_0 + 1$, and that the part reaches r_{i_0} in finite time. This is true for all stable sides, so the part must eventually rest on all stable sides. ■

Lemma 4.2: For any initial position (x_0, i_0) and any final position (x_f, i_f) such that $i_f = i_0$ and $x_f > x_0$, there exists a sequence of throws $\{\dot{\theta}_r^1, \dot{\theta}_r^2, \ldots, \dot{\theta}_r^k\}$ such that

$$\prod_{j=1}^{k} T^{\dot{\theta}_r^j}(x_0, i_0) = (x_f, i_f).$$

Proof: ***** I changed this proof. I think it is much simpler now.** The x coordinate of the left support vertex at time t after release is

$$V_x(t) = R - (h + d^{i_0})\dot{\theta}_r t + u_L^{i_0} \cos(\dot{\theta}_r t) - v_L^{i_0} \sin(\dot{\theta}_r t).$$

For a sufficiently small release velocity $\dot{\theta}_r$, the flight time t_{impact} is also small, and the small-angle approximation

allows us to write the net displacement after time t_{impact} as

$$\Delta V_x = -h\dot{\theta}_r t_{\text{impact}},$$

where t_{impact} is monotonically increasing with the control $\dot{\theta}_r$. By our design choice of $h < 0$, we have $\Delta V_x > 0$ (an outward jog). By the Implicit Function Theorem [33], there exists a $\delta > 0$ such that for all $x^* \in (x_0, x_0+\delta)$, there exists a $\dot{\theta}_r$ such that $T^{\dot{\theta}_r}(x_0, i_0) = (x^*, i_0)$. We can connect together open $\frac{\delta}{2}$ neighborhoods from x_0 to x_f to create the sequence of throws necessary. Note that the number of throws required is bounded above by $\frac{x_f - x_0}{\delta}$, so k is finite. ■

Lemma 4.1 and 4.2 are combined in Proposition 4.3.

Proposition 4.3: Let S be a limit set for the input $\dot{\theta}_r^0$ and $l_i \in \inf S$. Then, for any initial position (x_0, i_0) and any final position (x_f, i_f) such that $x_f > l_{i_f}$, there exists a sequence of throws $\{\dot{\theta}_r^1, \dot{\theta}_r^2, \ldots, \dot{\theta}_r^k\}$ such that

$$\prod_{j=1}^{k} T^{\dot{\theta}_r^j}(x_0, i_0) = (x_f, i_f).$$

Proof: This is a direct consequence of the two previous Lemmas. Given an initial configuration (x_0, i_0) and a final configuration (x_f, i_f), we can choose any $\dot{\theta}_r$ to get the part to side i_f by Lemma 4.1, and we need only choose one such that l_{i_f}, the lower bound in x for side i_f in the limit set, is less than x_f. Then, by Lemma 4.2, we can use a sequence of low-velocity throws to jog from (x, i_f) to (x_f, i_f). ■

That is, the system is controllable from (x_0, i_0) to (x_f, i_f) provided that $x_f > l_{i_f}$, the bound on the limit set.

A. Experimental Implementation

Notice that the proof of Lemma 4.2 implies that arbitrarily small translations in x can be achieved. Therefore, if it is sufficient that the part be within ϵ of x_f, we can adopt a simple feedback controller. First we repeatedly use a relatively large release velocity to get the part to the desired side. This is sensed with a one-bit sensor. Once on the correct side, we switch to a low release velocity that slowly jogs the part to the goal x_f coordinate. If $\dot{\theta}_r$ is chosen sufficiently low, the goal x_f value can be reached with arbitrarily good precision, where the presence of the part in the goal state is sensed by another one-bit sensor. There is a tradeoff in this precision, however; the smaller the required goal radius ϵ, the lower the release velocity, meaning shorter jogs and a longer time to reach the goal region.

We implemented this control strategy on the experimental system. The part is a 30-60-90 triangle with a hypotenuse of 17 cm and a center of mass located 5 cm from the short side of the right angle and 3 cm from the long side of the right angle. The arm approximates an impulsive throw by recoiling a few degrees and then controlling its angle to a setpoint (using a PD controller) such that the maximum overshoot is when the arm is approximately horizontal. The vision system acts as the necessary one-bit sensors.

Fig. 7 (top) shows the stopped arm after the part has reached the desired position and orientation. Fig. 7 (bottom) is a plot of the configuration of the part during the run. We can see the part starts on side 2 and jogs forward until its critical point. ***** Todd: the figure doesn't match the description; it's never on side 2. Also, maybe a more important issue: everywhere else in the paper you show parts flipping to edges of increasing number, but here the flips are to decreasing numbers. What's up? Note in our previous TRA paper, we always flipped to edges of decreasing number.** At this point, the part flips counterclockwise and onto side 1 where the controller switches to low amplitude throws. The part jogs forward until it reaches the end of the arm. This experiment was repeated 20 times with random initial conditions (obtained by dropping the part onto the arm) and resulted in the part being within 1 cm of the desired position on the arm in the correct orientation 100% of those times.[1]

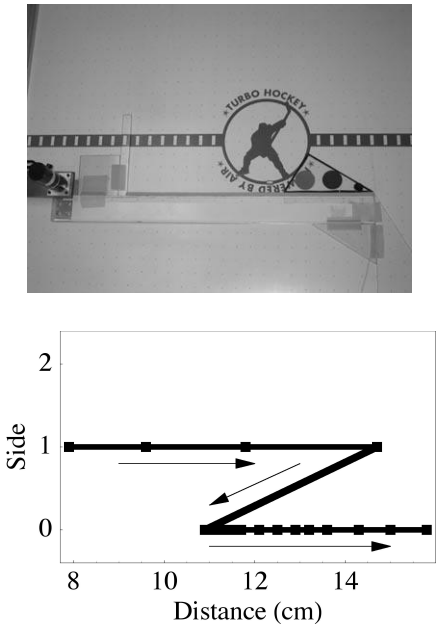


Fig. 7. (Top) A part in its final controlled configuration. (Bottom) The trace of the part's configuration.

***** experimental results would be stronger if (1) you could give the release velocity, and (2) if the results matched with simulations. especially later when you shape limit sets; do you see experimentally the kind of shaping you predicted?**

V. Shaping Forward Limit Sets

We now address the second contribution of this paper—designing part properties and the throwing velocity to shape the forward limit set of a throw map. In this case, we will design $\sup S$ to guarantee that a part can only reach a particular location on the arm if it is in the desired orientation. We first compute an approximation of the throw map mapping an initial configuration (x_0, i_0) to final configuration (x_f, i_f). We use this to compute the critical points at which the part will change from jogs to flips. Then we use a nonlinear root finding algorithm to solve for physical parameters given

[1]Movies of all experiments can be found at *http://ece.colorado.edu/∼murphey*.

desired critical points. The design of the parts handling is therefore completely in terms of the critical points associated with the limit set behavior. One can choose any number of physical parameters to vary, including density, center of mass location, throwing velocity, side length, etc. For concreteness, we will focus on the choice of only three parameters, the location of the center of mass and the throwing velocity, but the approach outlined here is valid for other choices. *** **Todd: when you change the CM location, how are you changing the inertia? are you keeping it fixed? note that the chosen inertia and CM may not be compatible (i.e., no way to get a large inertia if all the mass is near a corner of the part).** All that is required is that the mapping from the parameter space to the desired critical points be surjective.[2] In fact, we will see that this calculation comes down to an implicit function of the form $f(p,r) = 0 \in \mathbb{R}^n$, where n is the number of stable sides a part has, p are design parameters, and $r = \sup S$. We want to solve for the parameters p given a desired r in order to "design" a limit set S. The Implicit Function Theorem from classical analysis [33] provides a way of testing whether this is at least locally possible. If it is, then a nonlinear root finder can be employed to find p given a desired r.

A. Computing the Limit Set Critical Points

We need to find an approximate set of relationships between the physical parameters p we have control over and the forward limit set (such as that found in Fig. 5). For every stable side i, we want to compute $r_i \in \sup S$ such that for all $x < r_i$, $T(x,i)$ is a jog, and for all $x > r_i$, $T(x,i)$ is a flip, based on the equations in Eqs. (1-3). The part will flip when its post-impact total energy (including potential and kinetic energy) is greater than the critical potential energy U^i_{crit} needed to rotate to the new side.

Although we can certainly solve for a throw map numerically, we are looking for design rules to help us iteratively design the parameters p to achieve desired critical points r_i. We cannot solve analytically for the part post-impact energy because of the sine and cosine terms, so we adopt the small-angle approximation around $\dot\theta_r t_{\rm impact} = 0$. With this approximation, solving for the impact time requires solving

$$V_y(t) = h + d^i + R\dot\theta_r t + \frac{1}{2}gt^2 + \dot\theta_r t u^i_L + v^i_L = h$$

for t, which yields

$$t_{\rm impact} = -\frac{2\dot\theta_r(R+u^i_L)}{g}.$$

***** Todd: looks like there might be technical errors in this subsection. The kinetic and potential energies after impact seem wrong, because the units are wrong or inconsistent. Please go over or re-derive equations in this section.** Plugging this into Eq. (1) we get the impact configuration

$$[x(t_{\rm impact}), y(t_{\rm impact}), \phi(t_{\rm impact})] =$$

$$\begin{bmatrix} R + \frac{1}{g}2\dot\theta_r\left(R\dot\theta_r + \dot\theta_r u^i_L\right)(h - v^i_L) \\ -\frac{1}{g}2R\dot\theta_r\left(R\dot\theta_r + \dot\theta_r u^i_L\right) + \frac{1}{g}2\left(R\dot\theta_r + \dot\theta_r u^i_L\right)^2 - v^i_L \\ -\frac{1}{g}2\dot\theta_r\left(R\dot\theta_r + \dot\theta_r u^i_L\right) \end{bmatrix}.$$

Using Eq. (4) to solve for $\dot\phi^+$ after impact, we get

$$\dot\phi^+ = -\left(\frac{-\rho^2\dot\theta_r + u^i_L\left(R\dot\theta_r - 2\left(R\dot\theta_r + \dot\theta_r u^i_L\right)\right) - \dot\theta_r v^i_L\left(-h + v^i_L\right)}{\rho^2 + (u^i_L)^2 + (v^i_L)^2}\right)\dot\theta_r.$$

With this we can determine the kinetic energy after impact ***** check!**,

$$K^i = \frac{\rho + m\left((u^i_L)^2 + (v^i_L)^2\right)}{2\left(\rho^2 + (u^i_L)^2 + (v^i_L)^2\right)^2}\Big(-\rho^2\dot\theta_r + u^i_L\left(R\dot\theta_r - 2\left(R\dot\theta_r + \dot\theta_r u^i_L\right)\right) - \dot\theta_r v^i_L\left(-h + v^i_L\right)\Big)^2.$$

Moreover, the potential energy after impact is ***** check!**

$$U^i = -gm\left(\frac{1}{g}2R\dot\theta_r\left(R\dot\theta_r + \dot\theta_r u^i_L\right) - 2\left(R\dot\theta_r + \dot\theta_r u^i_L\right)^2 + v^i_L + \sqrt{(u^i_L)^2 + (v^i_L)^2}\right.$$

Given the parameters $p = [u^0_L, v^0_L, \dot\theta_r]$, we can calculate the function $F^i(p,R) = K^i(p,R) + U^i(p,R) - U^i_{crit}(p)$ for each of the n stable edges, $i = 0, \ldots, n-1$. The function $F^i(p,R)$ evaluates to zero at (approximately) $R = r_i$, the maximum x coordinate belonging to the limit set $S(p)$ for edge i, where the part transitions from a jog to a flip. The goal is to choose the parameters p such that $F^0(p, r^*_0) = F^1(p, r^*_1) = \ldots = F^{n-1}(p, r^*_{n-1}) = 0$, where $\{r^*_0, \ldots, r^*_{n-1}\}$ are the desired transition points encoding the desired shape of the forward limit set.

Defining $F = [F^0, \ldots, F^{n-1}]$, an iterative algorithm for designing a forward limit set can be summarized as follows.

Algorithm for Designing Limit Sets

Given a desired limit set specified by $\sup S_d$:

1) Choose a set of initial values for the design parameters p;
2) Compute the critical points $\sup S(p)$; if $\sup S(p) \approx \sup S_d$, return the successful design p, otherwise continue;
3) Approximate the throw map by linearization and determine if the Jacobian $\partial F/\partial p$ is full rank; if so, continue to step 4, otherwise abort;
4) Use a root-finding technique (such as Newton-Raphson) to take a step in the parameter space $p \leftarrow p + \delta p$ to bring $\sup S(p) - \sup S_d$ closer to zero;
5) Go to step 2.

Note that the rank of the Jacobian $\partial F/\partial p$ is limited by the number of parameters p. We have chosen three parameters (center of mass location and throwing velocity), so in our example below, we design the three critical points of a triangle's throw map.

[2] Because we use only three parameters in our example, we can independently design at most three critical points in $\sup S$.

B. Example: Designing a Triangle Limit Set

Our goal is to design the limit set for a triangle so that the part can only appear at a particular location on the arm in a particular orientation. We can then add a fixture to the arm to transform the shaped globally-attractive forward limit set into a globally-attractive fixed point: i.e., the open-loop motion of the arm is guaranteed to uniquely position and orient the part.

***** Todd: perhaps rewrite this subsection to explicitly give the dimensions of the triangle and its CM initially. (still the 17 cm hypotenuse?) Then the figure showing the design iterations should also show the designed part next to the limit set, so we can see what the design procedure is doing to the part. Is the experiment using the same part designed in the figure showing the design iterations?**

We now consider whether or not the function $F = K + U - U_{crit} = 0$ is locally solvable for $p = [u_L^0, v_L^0, \dot{\theta}_r]$ given a set of desired critical points r_0^*, r_1^*, r_2^*. For instance, consider a triangle like the one pictured in Fig. 4 with $(u_L^0, v_L^0) = (-2, -3.5)$ (with side 0 being the hypotenuse). Figure 8(a) shows the corresponding limit set with $\dot{\theta}_r = 5.6$ rad/s$(= 320$ deg/s$)$. It is clear that one cannot predict the orientation of a part purely by its location on the arm. Instead, we would like a limit set that is skewed so that some areas on the arm can only be reached in a particular orientation. That is, our design criterion is that $r_0 \gg r_i$ for $i \neq 0$. ***** This is inconsistent with the design iteration figure, that shows that r_i is largest for $i = 2$. I think you should define somewhere how you are numbering edges (increasing as you go CCW, for example), and stick to it through the paper.** The rank of the Jacobian $\partial F/\partial p$ is three near the operating point $[u_L^0, v_L^0, \dot{\theta}_r] = [-2, -3.5, 5.6]$, so we can locally deform the limit set. If we choose a desired limit set of $\{r_0^*, r_1^*, r_2^*\} = \{10, 10, 20\}$, we find (using the *Mathematica* command *FindRoot*, which uses a damped Newton's method) that a choice of $[u_L^0, v_L^0, \dot{\theta}_r] = [-2.8, -3.8, 4.6]$ gives rise to these critical points.

This is shown in Fig. 8 where Figure 8(b) shows an intermediate calculation in this root finding process and Fig. 8(c) the final limit set. Note that the linearized throw map leads to some error; Fig.8(c) indicates that the final numerically-simulated forward limit set has $\{r_0, r_1, r_2\} \approx \{14, 14, 20\}$. Nonetheless, the approximate model was sufficient for creating the behavior we were looking for: if the x coordinate is 20 cm or more, we know the part is on side 2. As we see in the experiment below, this property can be exploited by adding a fixture to create a globally-attractive fixed point of the throw map.

C. Experiment: Globally-Attractive Fixed Point

To test limit set design, we again used a 30-60-90 plexiglas triangle with a hypotenuse of 17 cm. We attached a lead weight to the triangle, allowing us to move the center of mass. Following the recipe in Section V-B, we numerically found that moving the lead weight into one of the corners helped to extremize the limit set, so that r_0 and r_1 are small and r_2 is large. ***** Can you give any numbers? Can you provide a picture of the final design? which sides are 0, 1, and 2?**

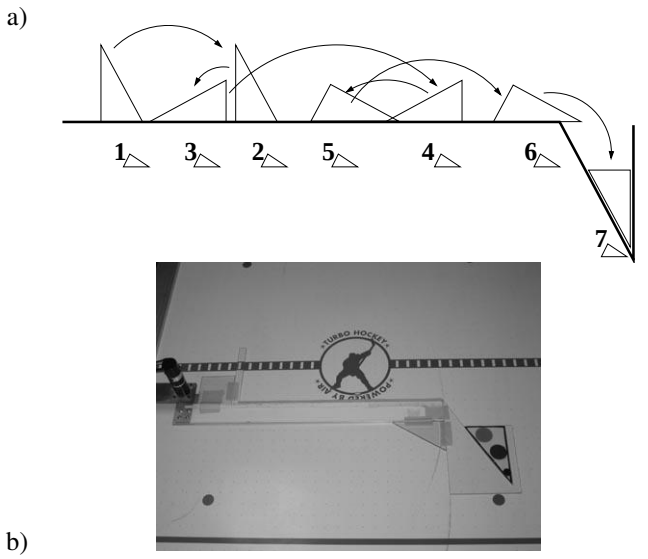


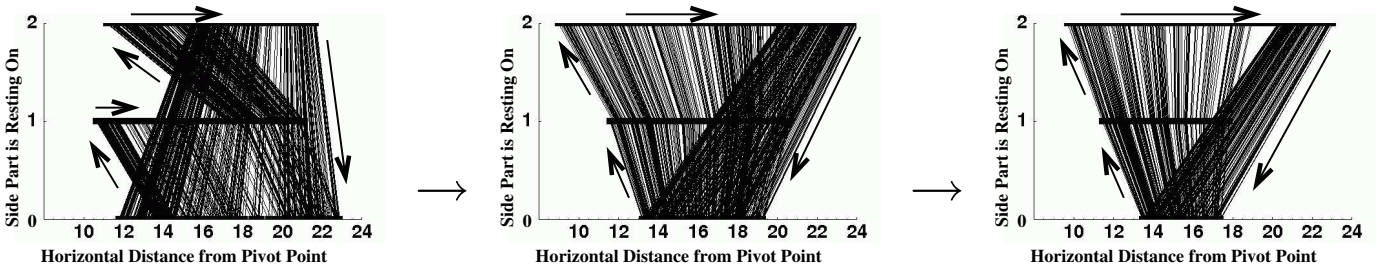
Fig. 9. By designing the limit set for one three-sided part, the part is "fed" at a unique orientation off the end of the arm in a completely sensorless fashion. The illustration shows the progress of a triangular part as it is thrown on the arm. (Note that each jog shown actually corresponds to multiple throws.)

what is the throw velocity? Did you simulate the limit set also, and does it look similar? To turn the globally-attractive forward limit set into a stable fixed point, we designed a cavity at the end of the arm (shown in Fig. 9(b)) to capture the part in a unique configuration. Because of the shaped limit set, the part only reaches the cavity on side 2, never on sides 0 or 1. The cavity is deep enough that subsequent throws do not eject the part. This experiment was repeated 20 times with random initial conditions and was successful 95% of those times (19/20). The one experiment failed because the part hit the wall of the cavity. ***** It always hits the wall, doesn't it? What does the previous sentence mean?** There is a tendency for this to occur when the center of mass of the part is very close to the end of the arm before the part falls into the cavity. This indicates the need for a better design of the cavity.

D. Experiment: Sequential Assembly

***** Todd: I don't understand this experiment. Are the parts put in sequentially? We might consider leaving this experiment out, unless it adds something new. I find the current description confusing.**

For open-loop assembly we design the stable limit sets independently such that both parts only enter the cavity in a desirable configuration. For the first part, we use the same triangle discussed above. For the second, we choose a four-sided polygon that, when placed in the cavity, will perfectly match with the sides of the cavity and the top of the triangle. This is what we consider our "assembled" state. If the limit set of the second part is designed correctly, the two parts will fall into the cavity in the arm in the assembled state (Fig. 10). We do not address the difficulty of inter-part interactions and balancing between the extremes of very low energy throws that do not provide stable limit cycle behavior and high energy throws that eject the parts from the cavity. This experiment



(a) First Iteration (b) Intermediate Iteration (c) Final Iteration

Fig. 8. A sequence of iterations in the numerical root-finding techniques used to determine the part shape. The first initial shape has the part reaching the same location on the robotic arm for each side. That is, the part flips to a new side by the time it reaches 20 cm, regardless of what side it is on. By changing the physical parameters (in this case the center of mass) one is able to compute a part that will only go to the end of the arm if it is on side 2. Hence, the part has been designed to be "self-assembled" mechanically.

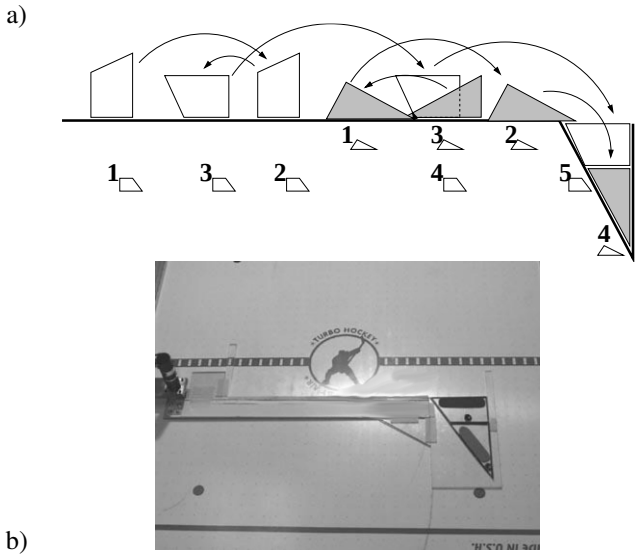


Fig. 10. By designing the limit sets for two parts (one three sided and one four sided), two parts are also "self-assembled" in a completely sensorless fashion. The illustration shows the progress of both parts as they are thrown on the arm and eventually self-assemble at the end of the arm.

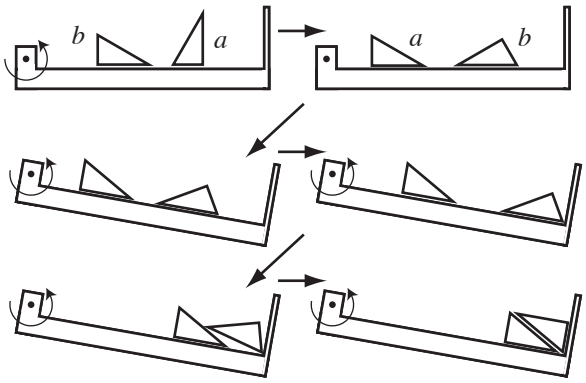


Fig. 11. Self-assembly of two parts. Top left: the parts a and b are not in the basin of attraction of the assembly throw map, so their configuration is randomized by high energy throws. Top right: the parts are in the basin of attraction. Middle left: The arm drops to a rest angle of $\alpha < 0$. Middle right: small jogs cause part b to reach the fixture. Bottom left: part a begins to assemble under part b. Bottom right: assembly is complete.

was repeated 20 times with random initial conditions and was successful only 55% of those times (11/20). Two of the most common failure modes included the second part being ejected from the assembled state on the next throw and the second part becoming stuck in the cavity in an incorrect orientation. These failures are due to two main factors. First, we did not analyze inter-part interaction, which needs further study. This could be mitigated by having the parts handled separately (either serially or in parallel by two different manipulators) in the experimental implementation. Secondly, we did not analytically design the cavity for capturing the parts, which meant that sometimes the parts were ejected before they were successfully assembled. Indeed, once the parts were in their assembled state, the parts only remained assembled after the next throw 85% (17/20) of the time.

VI. Self-Assembly of Two Parts

In this last experiment, two 30-60-90 triangles are assembled into a rectangle. To do this, we use two throw maps—one high-energy low-priority map to randomize the parts configuration, and one low-energy high-priority map to perform the assembly. A one-bit sensor recognizes when the parts are in the basin of attraction of the assembled state in the high-priority map.

To accomplish the assembly, we modify the arm to have a fixture consisting of a vertical wall at the end of the arm. It remains to design the high-priority throwing motion so that the two parts assemble against the wall.

The two parts are uniform-mass 30-60-90 right triangles, as shown in Figure 11. Each triangle is a reflection of the other. The basin of attraction of the high-priority assembly throwing motion has part a resting on its intermediate-length edge wholly to the left of part b, which is resting on its hypotenuse. This two-dimensional set of resting states has nonzero measure in the space of all possible resting configurations, and we have verified experimentally that randomizing high-velocity throws eventually take the parts to this set.

When the sensor recognizes the system in the basin of attraction of the assembled state of the high-priority map, the arm changes its rest angle from horizontal to an angle $\alpha < 0$, so that the vertical wall at the end of the arm serves to create a potential well (Figure 11). The arm then undergoes a series

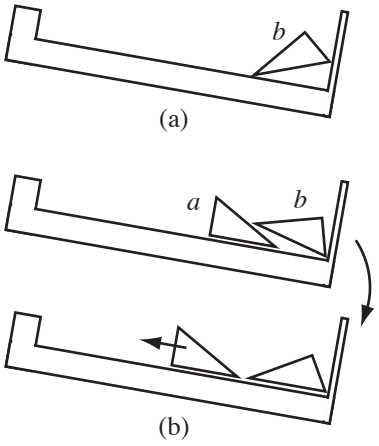


Fig. 12. To ensure proper assembly, (a) part b must not lean against the fixture, and (b) part a should not slip out from under part b when the assembly is only partial.

of low-amplitude throws that have the effect of causing part b to jog outward until it reaches the wall, at which point its discrete-time throw map becomes the identity map. Part a moves outward more slowly, due to the fact that the jog distance depends approximately linearly on the part's distance from the arm's rotation joint. Part a eventually slides under part b, as can be verified by a kinematic analysis of the initial upward velocities of part a's right support vertex and part b's left support vertex.

To ensure correct assembly, we have the freedom to choose the arm angle $\alpha < 0$ and the release velocity $\dot{\theta}_r > 0$ for the high-priority assembly map. We must ensure that part b does not lean against the wall (Figure 12(a)), and that intermediate assembly states, when part a is only partially under part b, are stable (Figure 12(b)), so that assembly can proceed.

Figure 13 analyzes the possibility of leaning. Triangle b is shown at top leaning against the wall at an angle β with the gravitational force acting at an angle $-\alpha$ with respect to vertical in the figure. Assuming a friction coefficient of μ between the part and the arm and frictionless contact between the part and the wall, Figure 13 graphically indicates the possible contact forces on the part as *moment-labeling regions* [34]. The resultant forces that can be provided by the contacts must pass counterclockwise around the region labeled $+$ and clockwise around the region labeled $-$. Therefore, the contacts cannot generate a force to resist the gravitational force, which passes through the $-$ labeled region. This indicates that the part must fall. If the friction coefficient μ were significantly larger, the part could remain stuck in the leaning position.

After a little manipulation, the condition that the gravity vector pass through the $-$ labeled region, and therefore part b does not get stuck in a leaning position, can be written

$$-\cos\alpha((0.14+\mu)\sin\beta - 0.58\cos\beta) + \sin\alpha(0.42\sin\beta - 0.14\cos\beta) \quad > \quad 0.$$

This condition is plotted in Figure 13 for $\mu = 1, 2$. Even for the extremely high friction coefficient $\mu = 2$, the part will

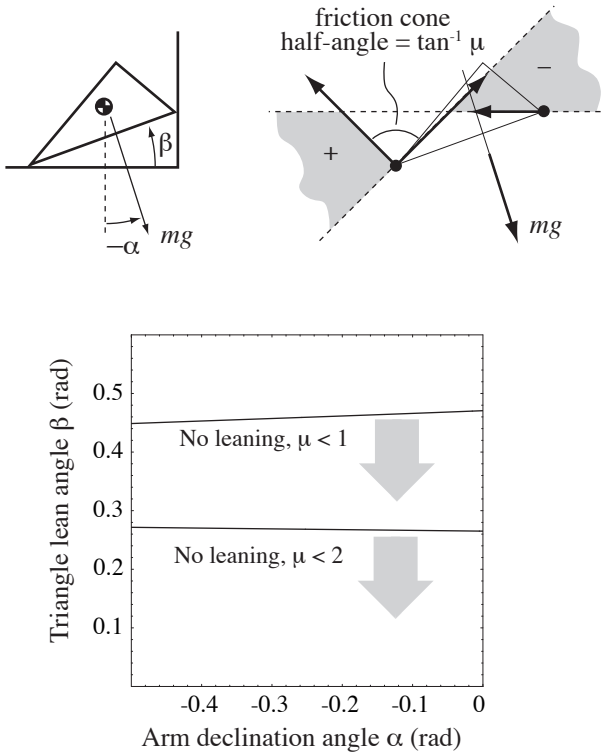


Fig. 13. Top left: a triangle leaning against the wall with the arm declined at an angle $\alpha < 0$. Top right: the possible wall and arm contact forces, represented as moment-labeling regions, and the gravitational force acting on the part. The contacts cannot resist the gravitational force, and the part must fall against the arm. Bottom: Leaning is impossible for the (α, β) regions shown for two different friction coefficients.

not remain leaning for $\beta < 0.25$ rad for any reasonable tilt angle α of the arm. This condition is enforced by limiting the release velocity $\dot{\theta}_r$, so the part never receives enough energy to achieve a lean angle greater than 0.25 rad.

Figure 14 illustrates a partially assembled state. Part b leans against part a at an angle $\gamma \in (-30°, 0°)$, where $\gamma = -30°$ is the fully assembled state. Assuming frictionless contact between the parts and static stability of the partial assembly, the contact force f_c applied by part a to part b is completely specified, as shown graphically in the free-body diagram for part a. Stability of the partial assembly is guaranteed if the force $f_{\rm tot}$ acting on part a, consisting of the gravitational force and the force $-f_c$, can be resisted by contact forces with the arm. Because the friction coefficient with the arm is large (measured $\mu > 1$ for our system), static stability is assured for all reasonable choices of α and all $\gamma \in (-30°, 0°)$.

For our experiments, we chose $\alpha = -10°$ and $\dot{\theta}_r =$, satisfying the "no lean" and partial assembly stability conditions above. ***** Todd, do we have these numbers?** The hypotenuse of each triangle was 17 cm. Three snapshots of the assembly process are shown in Fig. 15. Data for one experiment are shown in Fig. 16. The arm throws parts with high amplitude throws until the 33rd throw, when the vision system recognizes that they are in the basin of attraction of the desired assembled state under the low-amplitude throwing

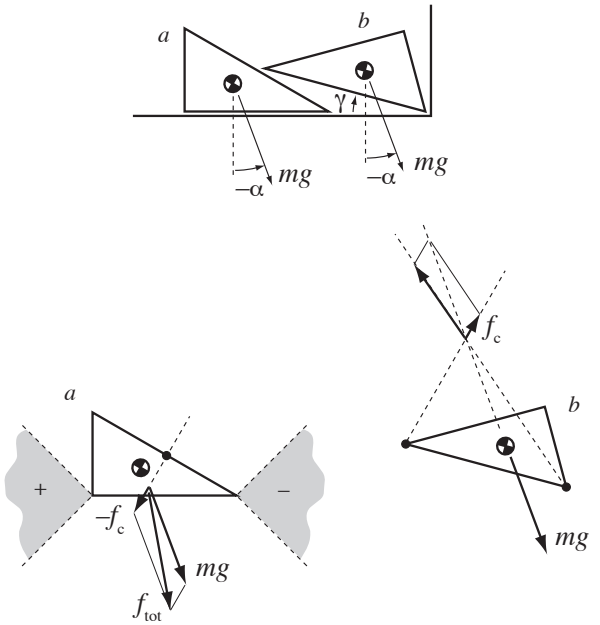


Fig. 14. Top: a partially assembled state with part b rotated $\gamma < 0$ relative to the arm, which is declined at an angle $\alpha < 0$. Bottom right: The uniquely determined contact force f_c part a applies to part b to keep part b stationary. Bottom left: f_{tot} is the sum of the gravitational force and the force $-f_c$ applied by part b to part a. This force can be resisted by the feasible contact forces with the arm (shown as moment-labeling regions), so the partial assembly is stable.

motions. The controller switches to low-amplitude throws until the two parts converge into their assembled state.

This experiment was repeated 20 times with the same initial conditions ***** Same initial conditions? Not randomized? what was the initial condition?** and was successful 70% of those times (14/20). The two reasons for failure were faulty detection of the controller switching condition and disruption of the orientations of the parts at the switch due to lowering the arm to angle α too rapidly. ***** Hmmm... these problems seem easily fixed; will the reviewer say "why didn't you fix them?" or were there more fundamental problems?**

VII. Conclusions

In this paper we proposed a technique for part manipulation and assembly that relies on the limit set behavior that parts experience during repeated throws. In order to successfully analyze an example system, we made many assumptions in the physical modeling and in the numerical techniques we employed, but we found that these assumptions successfully predicted our ability to design the limit set behavior in the part. This suggests that the limit set behavior is quite robust with respect to the specifics of materials used and other experimental particulars. Hence, we hope that these techniques may be extended beyond their current preliminary stage to develop self-assembly design techniques for a broader class of systems.

We develop a design technique that only relies on the gross-level features of the dynamics, which we can expect to be characteristic of many different part types, materials, contact

Fig. 15. Three snapshots of the two parts being assembled. Top: parts are in the basin of attraction of the assembled state under the high-priority low-energy throwing map. Middle: a partial assembly. Bottom: the final assembly.

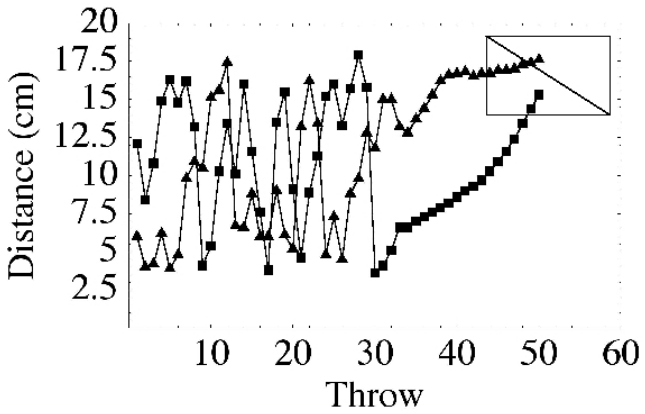


Fig. 16. A plot of distance on arm versus number of throws. Triangles denote the location of one triangle while squares denote the location of the other triangle. Assembly was achieved in this particular experiment after fifty throws.

dynamics, etcetera. This differs from traditional approaches in that it does not require accurate representations of the dynamics and moreover does not rely on any "quasi-static" assumption.

In this paper, we use the existence of limit sets in a particular, very simple experiment to show that parts will eventually land in configurations suitable for assembly. We then validate that this approach can produce stable assemblies for this experiment. However, it is certainly true that this approach is not generic, and it is additionally true that there are many more high-level dynamical structures (e.g., bifurcations, weakly invariant sets, etcetera) that could be exploited. Nevertheless, our hope is that the technique presented here may be further explored as a means for developing generic methods for design of mechanical systems that will "self-assemble" under external forcing.

For parts sorting and feeding, the proposed approach seems to work very well, and experiments matched analytical predictions quite well. For open-loop assembly, more work must be done. The fact that the open-loop assembly worked at all experimentally when only using rigid body properties of the parts and arm indicates the strength of these techniques. However, the fact that it only works a little more than half of the time indicates the need to have better models of inter-part impacts and to design the trapping mechanism. Moreover, the ultimate goal of this research is to be able to successfully self-assemble hundreds of parts, again indicating that inter-part interaction must be a fundamental aspect of further study. This will mean that the explicit integration of the equations of motion (such as that done in Section V-A) will be impossible, leading to the need for implicit techniques that only depend on analyzing the nonsmooth equations of motion for the parts. This will be the object of future study.

ACKNOWLEDGMENT

This work was supported in part by National Science Foundation grants IIS-0082957 and IIS-9811571.

REFERENCES

[1] E. W. Aboaf, S. M. Drucker, and C. G. Atkeson. Task-level robot learning: Juggling a tennis ball more accurately. In *Proceedings 1989 IEEE Int Conf Robotics and Automation*, pages 1290–1295, Scottsdale, AZ, 1989.

[2] S. Akella, W. Huang, K. M. Lynch, and M. T. Mason. Parts feeding on a conveyor with a one joint robot. *Algorithmica*, 26(3):313–344, 2000.

[3] S. Akella and M. T. Mason. Parts orienting with partial sensor information. In *IEEE International Conference on Robotics and Automation*, pages 557–564, 1998.

[4] S. Akella and M. T. Mason. Parts orienting with shape uncertainty. In *IEEE International Conference on Robotics and Automation*, pages 565-572, 1998.

[5] D. Berkowitz and J. Canny. Designing parts feeders using dynamic simulation. In *IEEE International Conference on Robotics and Automation*, pages 1127–1132, 1996.

[6] R. P. Berretty, K. Goldberg, L. Cheung, G. Smith, and A. F. van der Stappen. Trap design for vibratory bowl feeders. In *In IEEE International Conference on Robotics and Automation*, 1999.

[7] N. Bowden, A. Terfort, J. Carbeck, and G. M. Whitesides. Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, 276(5310):233–235, April 1997.

[8] V. Buchholtz, T. Poschel, and H. J. Tillemans. Simulation of rotating drum experiments using non-circular particles. *Physica A*, 216:199–204, 1995.

[9] M. Buehler, D. E. Koditschek, and P. J. Kindlmann. Planning and control of robotic juggling and catching tasks. *International Journal of Robotics Research*, 13(2):101–118, Apr 1994.

[10] M. Buhler and D. E. Koditschek. From stable to chaotic juggling: Theory, simulation, and experiments. In *Proceedings 1990 IEEE Int Conf Robotics and Automation*, pages 1976–1981, Cincinnati, OH, 1990.

[11] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–55, June 1999.

[12] B. R. Donald. On information invariants in robotics. *Artificial Intelligence*, (72):217–304, Jan. 1995.

[13] E. E. Ehrics, H. M. Jaeger, G. S. Karczmar, V. Y. Kuperman, and S. R. Nagel. Granular convection observed by magnetic resonance imaging. *Science*, 267(5204):1632–1634, Mar. 1995.

[14] M. A. Erdmann. Randomization for robot tasks: Using dynamic programming in the space of knowledge states. *Algorithmica*, 10:248–291, 1993.

[15] M. A. Erdmann. Understanding action and sensing by designing action-based sensors. *International Journal of Robotics Research*, 14(5):483–509, October 1995.

[16] M. A. Erdmann and M. T. Mason. An exploration of sensorless manipulation. *IEEE Transactions on Robotics and Automation*, 4(4):369–379, 1988.

[17] J. T. Feddema, P. Xavier, and R. Brown. Assembly planning at micro scale. In *Proc. IEEE Workshop on Precision Manipulation at Micro and Nano Scales*, Belgium, 1998.

[18] K. Y. Goldberg. Orienting polygonal parts without sensing. *Algorithmica*, 143(2/3/4):201–225, 1993.

[19] D. Gudmundsson and K. Goldberg. Tuning robotic part feeder parameters to maximize throughput. *Assembly Automation*, 19(3):216–221, 1999.

[20] J. D. Hartgerink, E. Beniash, and S. I. Stupp. Self-assembly and mineralization of peptide amphiphile nanofibers. *Science*, 294:1684–1688, 2001.

[21] T. Hasegawa, K. Murakami, and T. Matsuoka. Grasp planning for precision manipulation by multingered robotic hand. In *International Conference on Systems Management and Cybernetics*, 1999.

[22] K. M. Hill, J. F. Gilchrist, J. M. Ottino, D. V. Khakhar, and J. J. McCarthy. Mixing of granular materials: a test-bed dynamical system for pattern formation. *International Journal on Bifurcation and Chaos*, 9(8):1467–1484, 1999.

[23] K. M. Hill, D. V. Khakhar, J. F. Gilchrist, J. J. McCarthy, and J. M. Ottino. Segregation-driven organization in chaotic granular flows. *Proceedings of the National Academy of Sciences*, 96(21):11701–11706, October 1999.

[24] H. Hitakawa. Advanced part orientation system has wide application. *Assembly Automation*, 8(3):147–150, 1988.

[25] K. Hosokawa, I. Shimoyama, and H. Miura. Dynamics of self-assembling systems: Analogy with chemical kinetics. *Artif. Life*, 1(4):413–427, 1994.

[26] D. V. Khakhar, J. J. McCarthy, J. F. Gilchrist, and J. M. Ottino. Chaotic mixing of granular materials in two-dimensional tumbling mixers. *Chaos*, 9(1), March 1999.

[27] E. Klavins. *Control Problems in Robotics*, chapter Toward the Control of Self-Assembling Systems, pages 153–167. Springer Tracts in Advanced Robotics. Springer-Verlag, 2003. Volume edited by A. Bicchi, H.I. Christensen, and D. Prattichizzo.

[28] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962, June 2006.

[29] A. M. Ladd and L. E. Kavraki. Measure theoretic analysis of probabilistic path planning. *IEEE Tran. on Robotics and Automation*, 20(2):229–242, April 2004.

[30] E. J. Leon, N. Verma, S. Zhang, D. A. Lauffenburger, and R.D. Kamm. Mechanical properties of a self-assembling oligopeptide matrix. *J. Biomater. Sci. Polym. Ed.*, 9:297–312, 1998.

[31] K. M. Lynch, M. Northrop, and P. Pan. Stable limit sets in a dynamic parts feeder. *IEEE Transactions on Robotics and Automation*, 18(4):608–615, 2002.

[32] Kevin M. Lynch and Craig K. Black. Recurrence, controllability, and stabilization of juggling. *IEEE Transactions on Robotics and Automation*, 17(2):113–124, April 2001.

[33] J. E. Marsden and M. J. Hoffman. *Elementary Classical Analysis*. Freeman, second edition, 1993.

[34] M. T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[35] G. Metcalfe, T. Shinbrot, J. J. McCarthy, and J. M. Ottino. Avalanche mixing of granular materials. *Nature*, 374:39–41, 1995.

[36] B. Mirtich, Y. Zhuang, K. Goldberg, J. Craig, B. Carlisle, and J. Canny. Part pose statistics: Estimators and experiments. *IEEE Transactions on Robotics and Automation*, 15(5):849–857, Oct. 1999.

[37] M. Moll and M. A. Erdmann. *Algorithmic and Computational Robotics: New Directions*, chapter Manipulation of pose distributions. A. K. Peters, 2000. In B. R. Donald, K. M. Lynch, and D. Rus, editors.

[38] P. Moncevicz, M. Jakiela, , and K. Ulrich. Orientation and insertion of randomly presented parts using vibratory agitation. In *Proceedings of the ASME 3rd Conference on Flexible Assembly Systems*, pages 41–47, 1991.

[39] T. D. Murphey, J. Bernheisel, D. Choi, and K. M. Lynch. An example of parts handling and self-assembly using stable limit sets. In *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, 2005.

[40] T. D. Murphey, D. Choi, J. Bernheisel, and K. M. Lynch. Experiments in the use of stable limit sets for parts handling. In *International Conference on MEMS, NANO, and Smart Systems (ICMENS)*, pages 218–224, Banff, Canada, 2004.

[41] L. Penrose and R. Penrose. A self-reproducing analogue. *Nature*, 179:1183, June 1957.

[42] A. Rao, D. Kriegman, and K. Y. Goldberg. Complete algorithms for feeding polyhedral parts using pivot grasps. *IEEE Transactions on Robotics and Automation*, 12(6), Apr. 1996.

[43] A. S. Rao and K. Y. Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *International Journal of Robotics Research*, 13(1):16–37, Feb. 1994.

[44] S. J. Rao, S. K. Bhatia, and D. V. Khakhar. Axial transport of granular materials in a rotating cylinder. 2 experiments in a non-flow system. *Powder Technol.*, 67:153–160, 1992.

[45] A. A. Rizzi and D. E. Koditschek. Progress in spatial robot juggling. In *Proceedings 1992 IEEE Int Conf Robotics and Automation*, pages 775–780, Nice, France, 1992.

[46] R. Ronsse, P. Lefevre, and R. Sepulchre. Sensorless stabilization of bounce juggling. *IEEE Transactions on Robotics*, 22(1):147–159, February 2006.

[47] A. Rosato, K. J. Strandburg, and F. Prinz. Why the brazil nuts are on top: Size segregation of particulate matter by shaking. *Physical Review Letters*, (58), Mar. 1987.

[48] S. Rusaw, K. Gupta, and S. Payandeh. Determining polygon orientation using model based force interpretation. In *IEEE International Conference on Robotics and Automation*, pages 544-549, 1998.

[49] K. Saitou. Conformational switching in self-assembling mechanical systems. *IEEE Tran. on Robotics and Automation*, 15(3):510–520, June 1999.

[50] K. Saitou and M. J. Jakiela. Subassembly generation via mechanical conformational switches. *Artif. Life*, 2(4):377–416, 1995.

[51] T. Sakaguchi, Y. Masutani, and F. Miyazaki. A study on juggling task. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1418–1423, Osaka, Japan, 1991.

[52] A. Sanderson. Parts entropy methods for robotic assembly system design. In *IEEE International Conference on Robotics and Automation*, pages 600–608, 1984.

[53] S. B. Savage and C. K. K. Lun. Particle size segregation in inclined chute flow of dry cohesionless granular materials. *J. Fluid Mech.*, 189:311–335, 1988.

[54] S. Schaal and C. G. Atkeson. Open loop stable control strategies for robot juggling. In *IEEE International Conference on Robotics and Automation*, pages 3:913–918, Atlanta, GA, 1993.

[55] A. Terfort and G. M. Whitesides. Self-assembly of an operating electrical circuit based on shape complementarity and the hydrophibic effect. *Adv. Mater.*, 10(6):470–473, 1998.

[56] G. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[57] H. J. Yeh and J. S. Smith. Fluidic self-assembly of gaas microstructures on si substrates. *Sensors and Materials*, 6(6):319–332, 1994.

[58] R. Zhang and K. Gupta. Automatic orienting of polyhedra through step devices. In *In IEEE International Conference on Robotics and Automation*, pages 550-556, 1998.

[59] S. Zhang, T. Holmes, C. Lockshin, and A. Rich. Spontaneous assembly of a self-complementary oligopeptide to form a stable macroscopic membrane. *Proc. Natl. Acad. Sci*, 90:3334–3338, 1993.

[60] O. Zik, D. Levine, S. G. Lipson, S. Shtrikman, and J. Stavans. Rotationally induced segregation of granular materials. *Phys. Rev. Lett*, 73:644–647, 1994.