Experiments in the Use of Stable Limits Sets for Parts Handling

Todd D. Murphey, David Choi, Jay Bernheisel, and Kevin M. Lynch
Mechanical Engineering Department
Northwestern University
Evanston, IL 62008

Abstract

Throwing and catching parts promises to be a powerful manipulation technique, but is analytically complicated by equations of motion involving friction and impacts. Traditional control techniques would suggest that one should directly consider the equations of motion and design inputs that produce a unique fixed point. Such analysis can be anywhere from tedious to intractable. However, one can show that some simple part manipulators exhibit limit set behavior, where the parts enter a set that is invariant under the mapping that corresponds to the throwing action. We show here that analyzing limit sets rather than the equations of motion provides a more user friendly method of analysis, yielding tractable methods for understanding and designing part manipulators. These methods are additionally related to the study of self-assembly, and we are able to "nearly" self-assemble parts. In particular, we use a experimental setup to investigate methods to throw parts into a stable assembly using binary feedback.

1. Introduction

Traditional robotic assembly involves a monolithic robot sequentially placing a number of parts into an assembly. Nearly all of the "assembly power" of the system is concentrated in the robot. As the number of parts in the assembly becomes large, or as the individual parts become small, however, the monolithic approach becomes intractable. In this case, we might distribute the "assembly power" among the parts themselves (e.g., by appropriate design of their shape, density, wetability, friction, charge, chemical, or other properties or local control laws determining the inter-part coupling forces) and the environment they move in (e.g., global fixtures, templates, shaped force fields, and energy input in the form of heating or agitation). When centralized environmental control of the individual parts is limited, so that the properties of the parts are crucial in inducing the assembly, the process is called "self-assembly."

Self-assembling systems come in many different flavors [12, 11, 1] and are receiving increased attention in the robotics and manufacturing communities [8, 9]. 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 motions of the parts' support surface, perhaps based on simple state feedback, such that there is a single, globally attractive fixed point of the dynamical system — the desired assembly. In these systems, the dominant forces are contact and restitution forces, friction, and gravity. These systems are driven, dissipative, nonlinear dynamical systems, and the detailed dynamics are complex. Our approach is to exploit qualitative features of the dynamical system, such as ergodicity or the existence of limit sets, to find controllers that induce the assembly using a small number of motion primitives and simple sensing.

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 [3]. 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). 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.

In this paper we study a very simple system to begin to explore the trade-offs in environment design (fixtures, like the depressions in the APOS tray), environment control variables (agitation), sensing requirements, and part in-

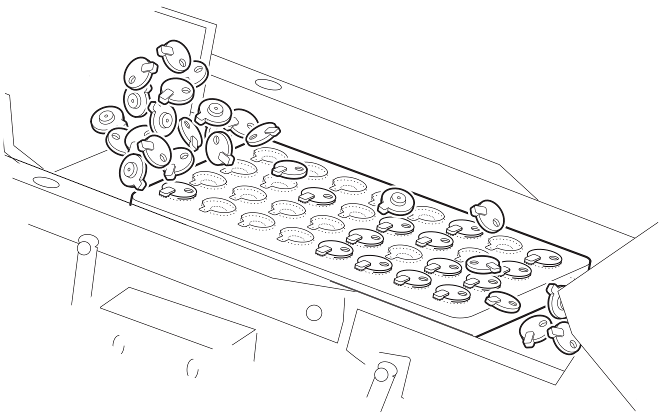

Figure 1. The Sony APOS parts orienting system (taken from [6]).

teraction forces (e.g., due to part geometry). 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 in a new configuration, then throws them again. The goal is to design the throws, based on very simple state feedback (perhaps just one or two bits), so that the parts eventually settle in the desired "assembly" (stacked in a corner in this case). This assembly is at a local minimum in the potential field. The solution we arrive at essentially uses high-energy throws to randomize the part configurations and switches to a low-energy throwing strategy once the state is recognized to be in the basin of attraction of the assembled fixed point.

Although this is a simple toy system, already the dynamics are sufficiently complex, and the control authority sufficiently limited, that the system resists traditional approaches to feedback control design. Instead, we rely on the existence of qualitative features such as ergodicity and limit sets under certain throwing actions. These features can be used to select actions and design simple sensors [2]. Our goal is to understand the minimal set of design, sensing, and control resources, expressed in terms of the environment and the individual parts, that induces the assembly.

In Section 2 we review a previous experiment in repetitive throwing and catching of a single part that led to the experiments in this paper. Section 3 describes the controllability of the configuration of a single part by throwing and catching it. Finally, Section 4 describes two experiments in feedback-controlled assembly: controlling a single part to a desired configuration, and controlling two parts to a desired assembly.

2. Previous work

The Flatland testbed for assembling planar parts by throwing and catching is shown in Fig. 2. The Flatland setup is composed of a large air hockey table that can support objects with a nearly frictionless air bearing. The angle of the table is adjustable, providing control on the effect of gravity in the support plane. Parts rest on a 1DOF rotary arm, and we control the motion of the arm to throw the parts. The parts then settle on the stationary arm. The arm is driven by a 1.1 amp Harmonic Drive RH-8 3006 motor. The air table supports an extruded 80/20 aluminum superstructure on which lights and camera are mounted. A Cognachrome vision board calculates the positions of parts on the table, and this data is used to simulate simple one-bit state sensors.

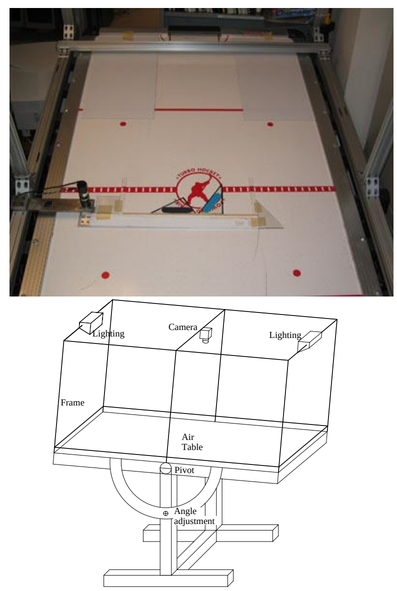

Figure 2. Flatland Experimental System

An initial experiment by Lynch, Northrop, and Pan [4] studied impulsive throwing and catching of a single polygonal part (Fig. 3). The "throw map" maps an initial configuration of the part to a final configuration after the part settles. The configuration space is $\mathbb{R}^+ \times \mathbb{N} \mod n$, where $\mathbb{N} \mod n$ 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 would eventually enter 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). A globally attractive forward limit set for a triangular part can be seen in Fig. 4. Label the sides of the triangle $0, 1, 2$. The limit set consists of outward jogs on side 1 until the part hits a critical radius at which it flips onto side 0. Then it flips to side 2, jogs outward to a critical point, and flips back to side 1, where the cycle repeats.

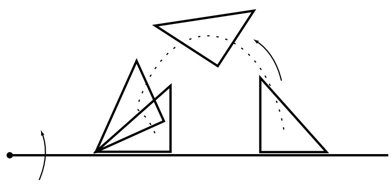

Figure 3. A part is thrown by an impulse from the rotary arm, then settles back to rest

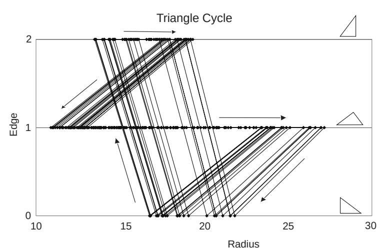

Figure 4. A limit set for a triangular part

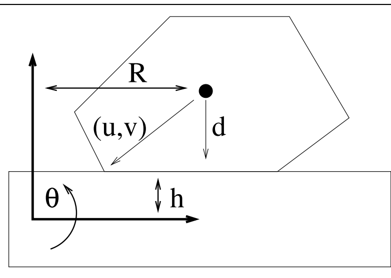

Figure 5. Triangle parameters

To understand this behavior, and to set the stage for our new experiment, we describe the equations of motion for this system. Assume that the part is always thrown such that the left bottom vertex (seen in Fig. 5) always hits the arm first, so that the resulting configuration after a throw will always either be on the initial stable side or on the stable side to the left. Using the parameters from Fig. 5, the (integrated) equations of motion of the part in flight after being thrown at time $t = 0$ are

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

where R is the location of the center of mass of the part on the arm, $\dot{\theta}_r$ is the impulsive angular velocity of the arm, h is the distance from the center of the arm to the edge of the part, and d^i is the distance from the center of mass to the arm when the part is on side i. The x and y coordinates are the location of the center of mass with respect to an inertial frame at the arm's joint and ϕ is the angle of the part with respect to the arm. (Note that when the arm is horizontal we have $R = x$.) The location of the (initially bottom left) vertex can be written

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

where u, v are the coordinates of the vertex in a body frame at the part's center of mass and aligned with the inertial frame at $t = 0$. Substituting Eq. (2) into Eq. (1) 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 see that we have to solve

$$h + d^i + R\dot{\theta}_r t + \frac{1}{2}gt^2 + \sin(\dot{\theta}_r t)u + \cos(\dot{\theta}_r t)v = h \tag{3}$$

for the smallest positive value of t, as there may be multiple solutions because of the trigonometric terms. Moreover, we assume that the impulses are sufficiently small and friction is sufficiently high that the left bottom vertex will always be the one that hits the arm and that it will stick after hitting. That is, we do not allow the object to rotate more than one side at a time and we do not allow for chattering after impact. Even with these simplifications, the throw map dynamics cannot be solved analytically.

It was shown in [4] that in this system a part will enter a unique forward limit set for a wide range of initial conditions, given an appropriately chosen $h > 0$ and $\dot{\theta}_r > 0$. That is, if we denote the resting location on the arm by x, the stable side the part is resting on by i, and the throw map from initial condition (x_0, i_0) to final position (x_f, i_f) for a given input $\dot{\theta}_r$ by $T(x_0, i_0)$, the part will enter a set of configurations $S \subseteq \mathbb{R}^+ \times \mathbb{N} \mod n$ such that $T(x, i) \in S$ for all $(x, i) \in S$. For an approximation of this system it is true that the trajectories will repeat themselves over time [4], because only a finite number of throws are required before a part will flip onto another side and there are only a finite number of sides. These facts will be key in understanding the relationship between the limit set properties and the controllability properties.

Beginning from this simple repetitive throwing experiment, in this paper we introduce the possibility of switching control actions based on one-bit sensors.

3. Manipulation with some feedback

As previously mentioned, dynamic part manipulation has complex equations of motion due to friction and impacts. If a part on the robotic arm shown in Fig. 2 is described by its position $x \in \mathbb{R}^+$ on the arm and its orientation $i \in \mathbb{N}$ (which is discrete and finite because there are only n stable resting orientations corresponding to the sides of a polygon), then its configuration space is $\mathbb{R}^+ \times \mathbb{N} \mod n$. Our goal is to find a sequence of throws ($\dot{\theta}_r > 0$) that will guarantee the parts ends up in a desired final configuration from any initial configuration with as little sensing as possible. Our approach to this problem is to constructively show that using a control defined by $\dot{\theta}_r$ (the throwing velocity) the system is controllable on $\mathbb{R}^+ \times \mathbb{N}$. That is, it can reach any (x_f, i_f) from any initial (x_0, i_0). By doing this constructively we also arrive at an algorithm that guarantees we arrive at (x_f, i_f) only using two bits of sensed information.

In the following analysis we again assume that after a throw the part will either stay on the stable side it initially rests on before the throw or that it will rotate to the next stable side. The outline of the procedure is as follows. First we show that for a constant throwing velocity $\dot{\theta}_r$ the part will eventually rest on all stable sides. Then we show that once the part is on a given side, there exists a sequence of throws that move it to a desired position on the arm. To actually implement a control to take the part to a desired configuration, we require two one-bit sensors: one to tell us when the part is on the correct side and another to tell us when it has (approximately) reached its destination.

Lemma 3.1 *For a given $\dot{\theta}_r$ and i_0 and limit set $S \subseteq \mathbb{R}^+ \times \mathbb{N} \mod n$, for all i_f there exists a k and $x > 0$ such that $T^k(x_0, i_0) = (x, i_f)$.*

Proof: Note that in Eqs. (1) and (3), $x(t)$ can be determined by first solving Eq. (3) for t and then substituting in to Eq. (1). We can locally solve this by replacing $\sin(\dot{\theta}_r t)u$ with $\dot{\theta}_r t u$ and replacing $\cos(\dot{\theta}_r t)v$ with v. Doing so, we get that

$$x(t) = R + \frac{2}{g}(h - v)(R + u)\dot{\theta}_r^2.$$

This implies that $x(t)$ depends linearly on R and quadratically on $\dot{\theta} r$. This, along with the fact that $x > 0$, implies that for any x_0, $x_{i+1} - x_i > x_i - x_{i-1}$. Therefore, every throw is at least ϵ in length and is monotonically increasing, implying there are no equilibria (as was also shown in [4]) before reaching the critical r_0 where it flips from side i_0 to side $i_0 + 1$, and that it reaches r_0 in finite time. This is true for all sides, so the part must eventually rest on all sides. ■

Lemma 3.2 *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 $\dot{\theta}_r$ inputs (i.e., $\{\dot{\theta}_r^1, \dot{\theta}_r^2, \cdots, \dot{\theta}_r^k\}$) such that $T^k(x_0, i_0) = (x_f, i_f)$.*

Proof: The implicitly defined throw map (in Eq. (3)) defines the relationship between inputs $\dot{\theta}_r$ and outputs x. We want to know if we can locally solve for $\dot{\theta}_r$ given a desired x. Since x can be determined by first solving Eq. (3) for t and then plugging into Eq. (1), this amounts to asking if t can always be written in terms of $\dot{\theta}_r$. Taking the derivative of $V_y(t)$ with respect to t, we get

$$\frac{\partial V_y}{\partial t} = gt + R\dot{\theta}_r + \dot{\theta}_r \cos(\dot{\theta}_r t)u - \dot{\theta}_r \sin(\dot{\theta}_r t)v$$

which, evaluated at $t = 0, \dot{\theta}_r = \varepsilon$, is

$$\frac{\partial V_y}{\partial t}|_{\dot{\theta}_r = \varepsilon, t=0} = \varepsilon(R + u)$$

which is positive for all $\varepsilon > 0$. Moreover, the relationship between t and $x(t)$ is linear as well, so by the implicit function theorem [5] and the chain rule, there exists $\delta > 0$ such that for all $x_f \in (x_0, x_0 + \delta)$ there exists $\dot{\theta}_r$ such that $T(x_0, i_0) = (x_f, i_0)$. Now, (in a manner similar to the controllability analysis found in [10]) 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. ■

Notice that the proof of Lemma 3.2 implies that arbitrarily small translations in x can be achieved. Hence, if we only desired that the part be within ϵ of x_f, we only need to choose a sufficiently small $\dot{\theta}_r$ and have some sort of feedback to tell us when the part has (approximately) reached x_f. Lemma 3.1 and 3.2 are combined in Proposition 3.3.

Proposition 3.3 *Let S be a limit set for the input $\dot{\theta}_r^0$ and let x_b be an upper bound on x for any i such that $(x, i) \in S$. Then, from any initial position (x_0, i_0) and any final position (x_f, i_f) such that $x_f > x_b$ there exists a k and a sequence of $\dot{\theta}_r^k$ such that $T^k(x_0, i_0) = (x_f, i_f)$.*

Proof: This is a direct consequence of the two previous Lemmas. Given an initial condition (x_0, i_0) and a final position (x_f, i_f), we can choose any $\dot{\theta}_r$ to get the part to side i_f by Lemma 3.1, we need only choose one such that x_b, the upper bound in x for the limit set, is less than x_f. Then, by Lemma 3.2, we can use a sequence of small $\dot{\theta}_r^k$ to get 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 $x_f > x_b$, the bound on the limit set. This can be tightened somewhat to include the fact that the bound x_b depends on i without changing the proof substantially. Moreover, using ideas from group theory [7], one can show that this system is controllable using the more restricted control set $(\dot{\theta}_r^0 - \varepsilon, \dot{\theta}_r^0 + \varepsilon)$. However, that analysis is beyond the scope of the work presented here.

The proof of Proposition 3.3 constructively provides controllability, thereby actually giving us an explicit algorithm for purposes of control. We first allow the dynamics to drive the part to the desired side i_f and then change the controller to drive the system to x_f. In the case of assembly of m parts, this amounts to solving the same problem for $(\mathbb{R}^+ \times \mathbb{N} \mod n)^m$, so long as no inter-part interactions are considered. We will address both of these cases experimentally in Section 4.

4. Experiments

The following two experiments are simple implementations of some of the preceding ideas. The parts are 30-60-90 triangles with a hypotenuse of 17 cm and 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. In the first one, a single part is controlled to a desired configuration on the arm. In the second one, two parts are assembled into a stack. In each case, the arm approximates an impulsive throw by recoiling the arm 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 tracks the progress of the parts, waiting for them to enter an appropriate state. Then the control law is switched to a more suitable, low amplitude input.

4.1. Single Part Manipulation

Figure 6. A part in its final configuration

By setting the amplitude of the first set of throws high enough, we can ensure that the part will flip onto any given side before the end of the arm, thereby avoiding it falling off the end. Once the part lands on the desired side, the vision system recognizes this fact and the controller switches to lower amplitude throws until the part reaches a desired position on the arm. During this second controller, the amplitude of the throws is only sufficient to jog the part forward and cannot flip the part. In this way, the final position and orientation of the part can be controlled. Fig. 6 shows the stopped arm after the part has reached the desired position and orientation.

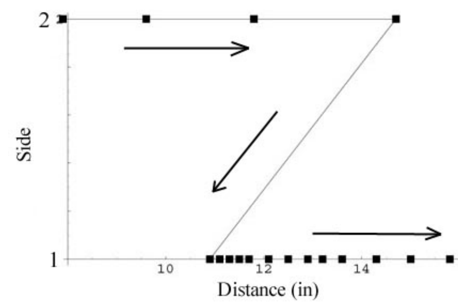

Figure 7. Manipulation with binary feedback

In Fig. 7 is a plot showing the side of the part versus distance on arm. We can see the part starts on side 2 and jogs forward until its critical point. 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 parts onto the arm) and resulted in the part being within an inch of the desired position on the arm in the correct orientation 100% of those times.

4.2. Assembly

In this experiment, the first controller throws two parts with a constant amplitude input until the parts are in the right relative configuration for assembly (the first picture in Fig. 8). This experiment assumes that the parts will land in the "correct" relative configuration after a finite amount of time (this is really an assumption in this case since we have not considered inter-part dynamics in this paper). Once this configuration is detected (via the vision system), the controller switches to lower amplitude throws as well as a lower angle of release. This allows the final assembly of parts to make use of the potential well defined by the downward sloping arm and the perpendicular "wall" at the end of the arm. The parts jog to the end of the arm, and are then squeezed together (the second picture in Fig. 8) until the two right triangles are assembled into a rectangle (the third picture in Fig. 9). This occurs because the further out on the arm a part is, the more rotation it experiences during a throw. Hence, the part that is further out on the arm will rotate more than the part next to it, ensuring that the part next to it will slide underneath. This case involves many impact interactions between the two parts leading to several configurations where the part did not land flat on one side. We ignored these interactions in our analysis from Section 3 and only switched the controller when the parts were in the correct relative configuration.

Figure 8. Three snapshots of the two parts being assembled.

We plot the data using the distance on the arm versus the throw (see Fig.9). We can see that the arm continues to throw the 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 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 and was successful 70% of those times (14/20). The two reasons for failure were that the controller switch occurred before the parts were in the correct orientation due to a vision acquisition error, and be-

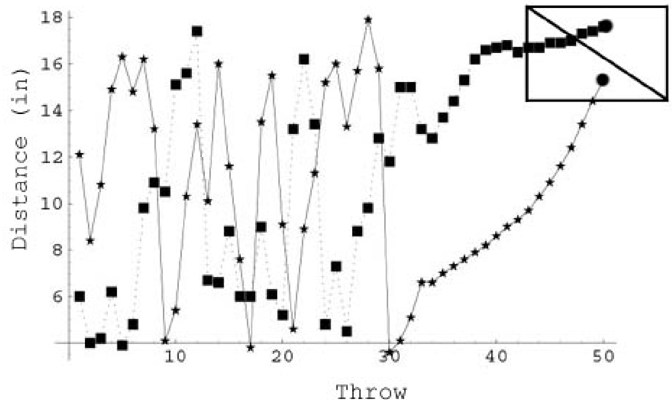

Figure 9. A plot of distance on arm versus number of throws.

cause the arm was lowered too rapidly during the controller switch. These problems can be solved by input shaping for the motor motions, waiting longer before capturing data, or improving our vision system. Perhaps most importantly, the use of an actual trapping mechanism at the end of the arm would help reduce sensitivity to the inter-part dynamics.

5. Conclusions

As we move away from the monolithic robot approach to assembly, we face the question of how to most usefully represent assembly problems for purposes of analysis and analytical design. In self-assembly in particular the sheer number of degrees of freedom and the complexity of the mechanics describing the coupling between those degrees of freedom seems to ensure that classical quantitative analysis should give way to qualitative, high-level analysis. 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.

Our ongoing work includes studying the limit sets and how they relate to part dynamics. In particular, it is possible to "reshape" the limit sets by changing the mechanical design of the parts, and doing so can allow one to choose where the critical points of that limit set (i.e., where the part transitions from jogging to flipping) are located. This, in principle, provides a tractable method for creating true self-assembling systems.

References

[1] 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.

[2] M. A. Erdmann. Understanding action and sensing by designing action-based sensors.

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

[4] 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.

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

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

[7] I. Mezic. Controllability of group translations. In *Proc. American Control Conference*, 2001.

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

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

[10] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[11] 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.

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