Part Pose Statistics: Estimators and Experiments

Ken Goldberg[*], Brian Mirtich[†], Yan Zhuang[‡], John Craig[§], Brian Carlisle[¶], and John Canny[||]

Abstract

Many of the most fundamental examples in probability involve the pose statistics of coins and dice as they are dropped on a flat surface. For these parts, the probability assigned to each stable face is justified based on part symmetry, although most gamblers are familiar with the possibility of loaded dice. Our goal is to develop a science base for part feeding, where parts arrive in random orientations. We consider the following problem: given part geometry and parameters such as center of mass, estimate the probability of encountering each stable pose of the part.

We describe three estimators for solving this problem for polyhedral parts with known center of mass. The first estimator uses a quasi-static motion model that is computed in time $O(n \log n)$ for a part with n vertices. The second, estimator has the same time complexity but takes into account a measure of dynamic stability based on perturbation. The third estimator uses repeated Monte Carlo experiments with a mechanics simulation package. Using a robot and computer vision system, we performed 3595 physical drop experiments using four real parts and determined the final orientation. We compare this data to the results from each estimator. We believe this is the first paper to systematically compare alternative estimators and to correlate their performance with statistically significant experiments on industrial parts.

1 Introduction

Our motivation for studying pose statistics is to develop a science base for part feeding. *Part feeders*, which singulate and orient parts prior to packing and insertion, are critical components of autmated assembly lines and one of the biggest obstacles to flexible assembly. Flexible assembly systems can be rapidly reconfigured to handle new or changed parts, which can dramatically reduce the time and costs needed to bring new products to market.

We consider the following problem, treating one part in isolation and assuming that the worksurface is flat and much larger than the part. For a rigid part P with known center of mass and inertia tensor, denote the n faces of its convex hull H by F_1, $\ldots$, F_n.

[*] Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, goldberg@ieor.berkeley.edu. This work was supported in part by the National Science Foundation under IRI-9612491 and Presidential Faculty Fellow Award IRI-9553197.

[†] Mitsubishi Electric Research Laboratory (MERL), Cambridge, MA 02139, mirtich@merl.com, Mirtich was supported in part by NSF grant #FD93-19412.

[‡] Computer Science Department, University of California, Berkeley, CA 94720–1776, yzhuang@cs.berkeley.edu. Zhuang was supported by the National Science Foundation under IRI-9612491 and Presidential Faculty Fellow Award IRI-9553197.

[§] Adept Technology, Inc. jjc@silma.com

[¶] Adept Technology, Inc.

[||] Computer Science Department, University of California, Berkeley, CA 94720-1776, (510) 642-9955, jfc@cs.berkeley.edu. Canny is supported in part by NSF grant #FD93-19412.

We treat part orientations as equivalent if one can be transformed to another with a rotation about the gravity axis in the world frame. We refer to the set of equivalent orientations as a *pose* of the part. To represent each pose, we attach a body frame to the part with origin at its center of mass. The unit gravity vector g in this frame corresponds to a point on the unit sphere in the body frame. Such a point uniquely defines a pose of the part. Thus we represent the space of part poses with S, the unit sphere. Let $\mathcal{P}_{\mathrm{initial}}$ be the initial probability density function on this space of poses and $\mathcal{P}_{\mathrm{final}}$ be the probability density function after the part comes to rest on the worksurface. For a polyhedral part, the part must come to rest on one of the faces of of its convex hull, so $\mathcal{P}_{\mathrm{final}}$ can be specified by $p_1, \ldots, p_n$, where p_i is the probability that part P has the final pose with face F_i in contact with the worksurface.

Estimating Pose Statistics (EPS) *Assume part P is repeatedly dropped from a known distribution of poses, $\mathcal{P}_{\mathrm{initial}}$ onto a flat worksurface. Estimate $\mathcal{P}_{\mathrm{final}}$.*

We consider three estimators for solving this problem for polyhedral parts. We start with an estimate based on a quasi-static motion model first reported in [38]. Next we describe a *perturbed quasi-static* estimator that incorporates a model of dynamic stability. We then introduce a third estimator based on repeated Monte Carlo simulation experiments using *Impulse*, a mechanics simulation package [26, 27, 25]. We discuss impulse-based simulation, a paradigm for efficient simulation, and present its model for frictional collisions.

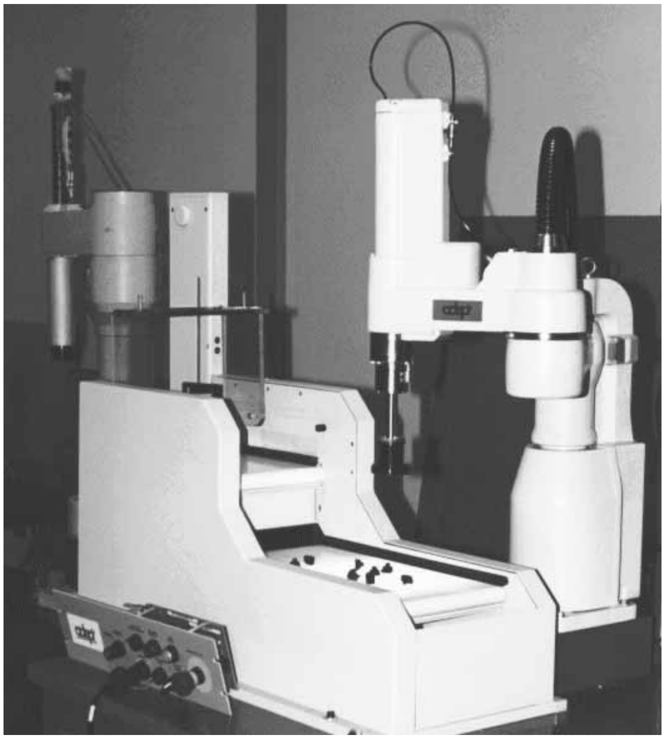

Figure 1: *A flexible parts feeding workcell using machine vision, a high-speed robot arm, and pivoting gripper.*

To evaluate these estimators, we used the robot and computer vision system shown in Figure 1 to perform 3595 physical drop trials using the four real parts shown in Figure 2. In each trial the system determined the part's final orientation. We compare this data to the results from each estimator. (This data was used as a benchmark for an alternative simulation system as described in [12]).

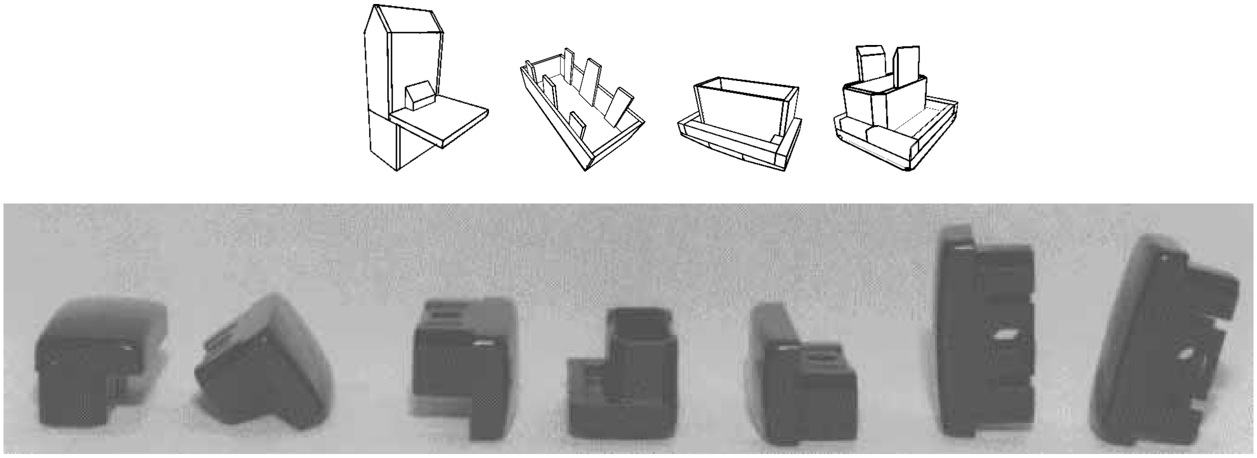

Figure 2: Top: CAD models of the four parts used in the experiments. From left to right: insulator cap, large white, rectangular black, and square black stereo buttons. Bottom: Photographs of the rectangular black stereo button in its seven stable states.

We find that dynamic simulation provides the most accurate results, but requires significantly more computation time. This paper is a revised and updated version of [24].

2 Related Work

An excellent introduction to mechanical parts feeders can be found in Boothroyd's book [6], which describes vibratory bowl feeders in detail as well as non-vibratory feeders such as the magnetic and revolving hook feeders. Sony introduced a novel approach using random motion of parts over part-specific pallets [35, 28].

Although there is a substantial body of research in feeder design, practitioners still rely on instinct and rules-of-thumb [19]. Part feeders are responsible for up to 30% of the cost and 50% of workcell failures [29, 6]. Thus systematic feeder design remains one of the biggest obstacles to automated manufacturing.

One of the earliest systematic efforts to model part feeding was Erdmann and Mason's analysis of the mechanics of a part moving in a tilting tray [14]. This was followed by a number of efforts to rigorously model mechanics and uncertainty [15, 7, 2, 20, 8]. A closely related example is Peshkin and Sanderson's study of feeding parts on a conveyor belt as they move against a sequence of passive fences [31]; this model was extended in a sequence of papers [1, 37, 4].

A variety of sensor-based (robotic) alternatives to mechanical bowl feeders have been proposed. For example, [22] propose an optical silhouette sensor with air nozzle to reject all but the desired pose on a feeder track. Carlisle *et. al.* [9] proposed a flexible part feeding system that combines machine vision with a high-speed robot arm. In contrast to custom-designed hardware such as the bowl feeder, only software is changed when a new part is to be fed. The idea is that a collection of like parts are randomly scattered on a flat worktable where they are subject to the force of gravity. An overhead vision system determines the position and orientation of each part. The robot arm then picks up each part and moves it into a desired final position and orientation as illustrated in Figure 1. A recent paper [16] outlines how feeder throughput can be estimated based on estimates of pose statistics, conveyor speed, and arm cycle time. Similar feeder designs are described in [10, 39].

To facilitate the design of parts feeders, researchers have considered configuration space models [14, 7, 2, 8], simulation [17, 3, 23], heuristics [19], and genetic algorithms [11]. The latter paper made use of our preliminary results in estimating pose statistics.

Boothroyd noted that the feedrate for a parts feeder is based on pose statistics [6]. He gave a quasi-static estimator for rectangular and cylindrical parts. Sanderson showed that robot assembly can be analyzed in terms of pose statistics [34]. [38] improved on the Boothroyd estimator by treating the convex hull of any polyhedral part and propagating probability from unstable faces. The resulting estimate is a good first approximation to experimental distributions but do not take into account effects such as bouncing, vibrations, collisions, and friction. An estimator based on face area and height of the center of mass was reported in [30] but tested only with rectangular parts.

3 The Quasi-static Estimator

In our first model, we ignore part inertia and velocity, treating part motion as quasi-static. We consider the part's initial pose to be uniformly distributed over the unit sphere S as explained in the previous section. After computing the part's convex hull H, the idea is to project the facets of H onto a sphere centered at the center of mass c. If F_π is the projection of face F, the ratio of the area of F_π to the total surface area of the sphere gives the probability that the part will land on face F under quasi-static conditions.

Assuming triangular faces, the ratio in question is given by

$$A = \frac{\beta_0 + \beta_1 + \beta_2 - \pi}{4\pi} \tag{1}$$

where the β_i are the interior angles of F_π (see Figure 3).

The β_i are computed as follows. Let $d_{c0} = \sqrt{c^2 + v_0^2}$, $d_{c1} = \sqrt{c^2 + v_1^2}$, and $d_{01} = \sqrt{v_0^2 + v_1^2}$. Using standard notation for triangles, let δ_2 be the arc that results from projecting the line from v_0 to v_1 onto the sphere (note that arcs are measured by the angle subtended at the center of the sphere). One can solve for δ_2 using the law of cosines,

$$d_{01}^2 = d_{c0}^2 + d_{c1}^2 - 2 d_{c0} d_{c1} \cos\delta_2, \tag{2}$$

and δ_0 and δ_1 are found similarly. Given all the δ_i, β_2 can be found using the spherical law of cosines,

$$\cos\delta_2 = \cos\delta_0 \cos\delta_1 + \sin\delta_0 \sin\delta_1 \cos\beta_2, \tag{3}$$

and analogous computations give β_0 and β_1.

This procedure results in an initial estimate of each p_i. To treat faces of H that are statically unstable, we project the center of mass onto the plane of each face F_i. If the projected point lies outside face F_i, gravity will cause the part to topple over to adjacent face F_j. In this case p_i is added to p_j, and p_i is set to zero.

To facilitate this propagation, we define the *quasi-static graph* (**QSG**) to be a directed graph in which each node corresponds to one facet of the convex hull H. The QSG has a directed link from node i to node j if and only if facet F_i topples to facet F_j and they share one common edge. Clearly, the QSG is acyclic. We propagate probability along the QSG using a breadth-first traverse.

The number of vertices and edges of P are both $O(n)$. We compute the convex hull in $O(n \log n)$ time. As a planar graph, the convex hull of P has $O(n)$ faces and interior angles. Thus we can compute the projected areas in $O(n)$ time because we only visit each interior angle once and can build the QSG in $O(n)$ time, because we just have to check the center of mass against the edges twice. It takes $O(n)$ time to do a breadth-first traverse of this graph. Therefore the total time to compute the probabilities is $O(n \log n)$.

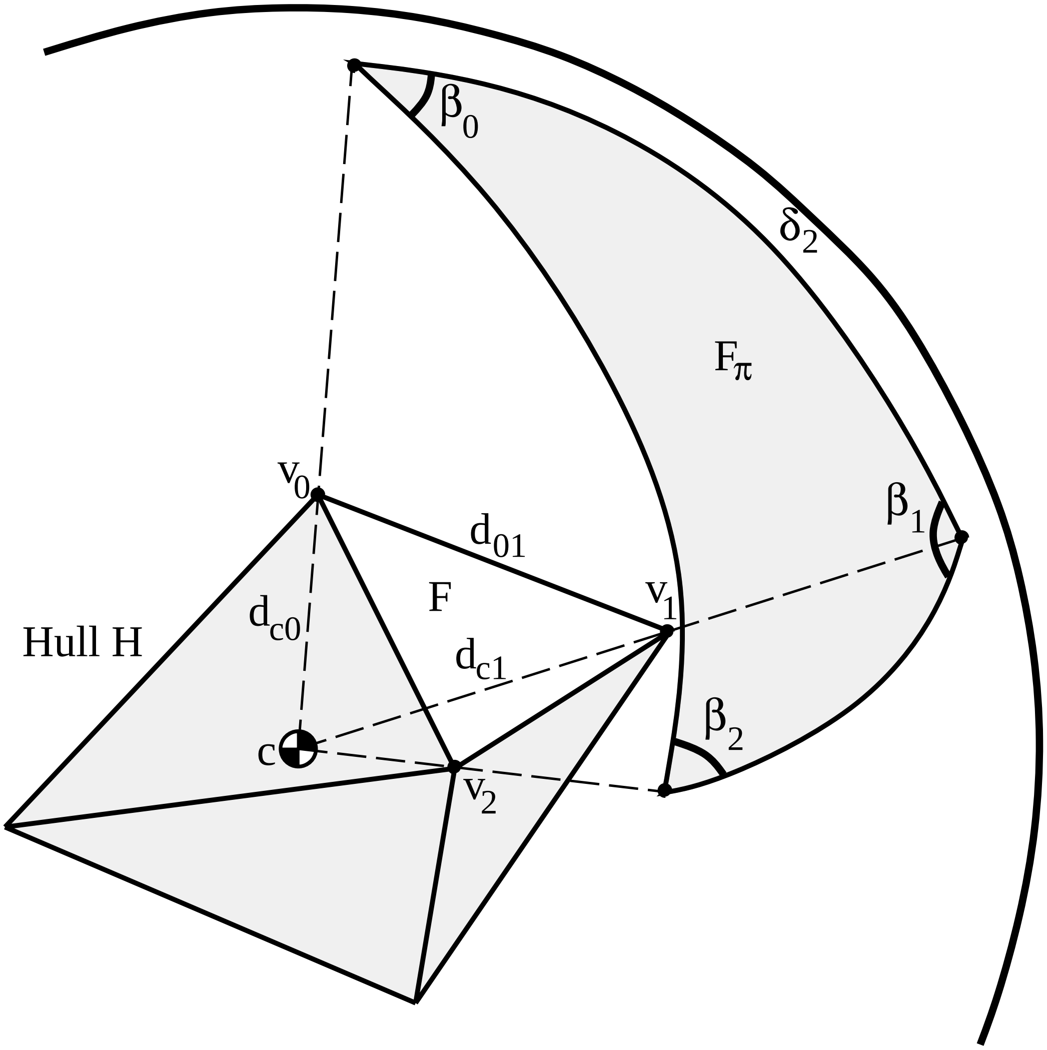

Figure 3: Computing initial probabilities for each face.

4 Perturbed Quasi-Static (PQS) Estimate

Since the quasi-static analysis does not model dynamic disturbances, it often overestimates the probability of landing on a facet that is stable but easily dislodged by small vibration. In this section we describe a modification to the quasi-static estimate that considers a "perturbation region" around each edge of a stable face. Consider two facets of the part's convex hull, F_i and F_j, and the bounding edge e between them. Let g be the downward vector from the part's center of mass. In the quasi-static estimate, we assume that if g intersects F_i when in contact, the part will remain on facet F_i. However, dynamic energy may cause the part to rotate across edge e when g points inside facet F_i but close to edge e (Figure 4). The spherical projection of the perturbation region that falls inside of the face yields a heuristic estimate of how likely the part is to topple from F_i to F_j across their shared edge. We call Δp_{ij} the "perturbation probability" for each pair of adjacent faces (each edge). We use these to compute a "perturbed quasi-static" (PQS) estimate in time $O(n \log n)$ which agrees better with the data from experiments.

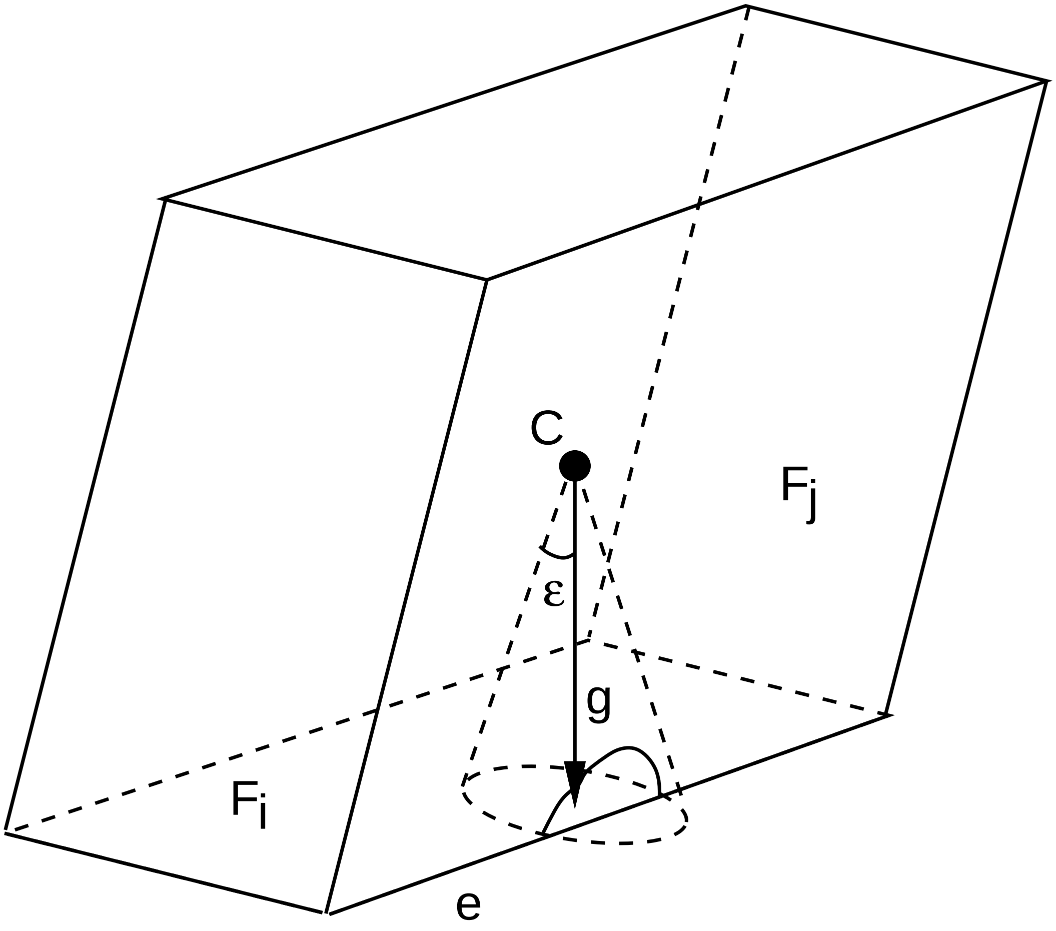

Figure 4: *Under the perturbed-quasi-static model, the part remains on facet F_i, since g intersects F_i. However dynamic effects make it likely that the part will topple onto facet F_j. To model this, we consider a "perturbation region" around the common edge using a cone of disturbance vectors.*

We consider perturbations to the gravity vector that form a right cone of half-angle ϵ with apex at the part's center of mass. The value of ϵ depends on how far dynamic forces can tilt the gravity vector. If we sweep the gravity vector g along the part edge e, the perturbation cone sweeps out a perturbation region around the edge.

To compute the perturbation probability, we consider the triangle formed by edge e and two edges from its endpoints to the projected center of mass on facet F_i. Call this T. When we project T onto the unit sphere, we denote the arc corresponding to e with a. If we translate the plane defined by the center of mass c and the arc a until it intersects the sphere with a new arc a' such that the spherical distance between a' and a is ϵ, the spherical region Σ between a' and a is the spherical projection of the perturbation region (Figure 5(c)).

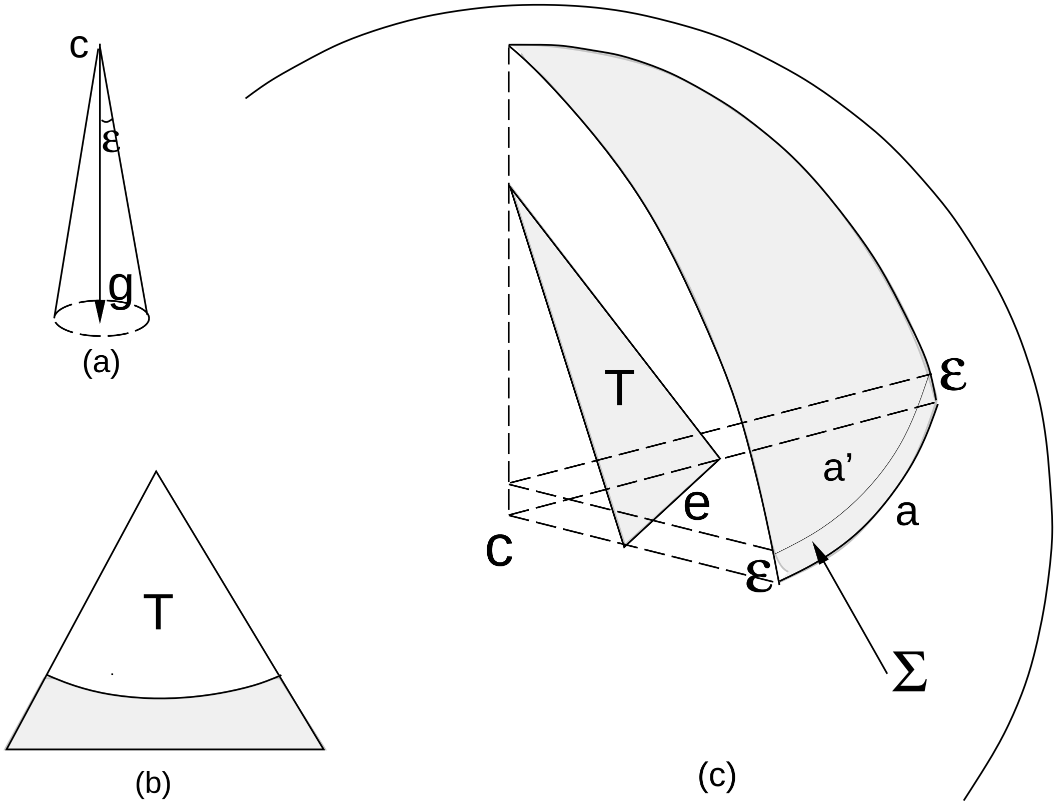

Figure 5: *(a) Perturbations to the gravity vector form a cone. (b) Perturbation region for one part edge. (c) Perturbation probability for each edge, using the PQS model.*

If we denote any point on the unit sphere by a vector $r = (x, y, z)$, which is parameterized by:

$$\begin{cases} x = \sin\varphi\cos\theta \\ y = \sin\varphi\sin\theta \\ z = \cos\varphi \end{cases} \tag{4}$$

then the area of Σ is:

$$\sigma = \int\int_{\Omega} ||N(\varphi,\theta)||d\varphi d\theta, \tag{5}$$

where Ω is the corresponding region in terms of φ and θ, and $N(\varphi,\theta)$ is the fundamental vector product of the surface, $r'_{\varphi} \times r'_{\theta}$. As we cannot solve this integral in closed form, we approximate the projected perturbation region by the area of a rectangle of length $|a|$ and width ϵ.

We transfer perturbation probability between adjacent facets and then propagate down the QSG as in the quasi-static estimate. We transfer Δp_{ij} from facet F_i to adjacent facet F_j if F_j is unstable or if F_i has a lower initial probability under the quasi-static estimate. The first condition insures that the perturbation probability will wind up at a stable facet after propagating through the QSG. Both conditions reflect the intuition that parts will tend to roll toward more stable states. The only extra computation is finding the probabilities of $O(n)$ perturbation regions and transferring the perturbation probabilities. Therefore the PQS estimate has time complexity: $O(n\log n)$.

The parameter ϵ in the PQS estimate depends on how far the part is dropped, how much mechanical energy it can store, the coefficients of friction and restitution, and other physical and dynamic properties. In our physical experiments, the drop heights and materials were constant, so we used data from the first physical experiment to choose a reasonable value: in this case $\epsilon = 20^\circ$. We used this same ϵ value to estimate the distribution for all parts.

The computation time on SPARC20 is less than 1 second for all 4 parts. About 90% of this is used for constructing the convex hull. It is important to keep in mind that the PQS is a heuristic estimate. We do not make any claims that this captures the intricate physics of dynamic collisions. A better heuristic may be possible by more sophisticated propagation of perturbation probabilities which we are now exploring. A full treatment of dynamic effects, at the price of increased computation, is described in the next section.

5 Dynamic simulation

To obtain more accurate pose distribution predictions, one could perform full dynamic simulations of the dropped part over many trials. This seems prohibitive for two reasons. First, the interaction between the part and environment is very collision intensive, and it is notoriously difficult to model the dynamics of collisions with friction [21]. Second, dynamic simulation is much slower than the previous described estimators, and so obtaining a statistically significant number of trials may take too long.

Mirtich and Canny have studied *impulse-based* simulation, a paradigm for dynamic simulation that addresses these problems. The method handles frictional collisions in a natural way, and for general 3D rigid body simulation, the simulator *Impulse* has the fastest execution times reported in the literature [27].

5.1 Computing frictional collisions

Details about *Impulse*, and a comparison of constraint- and impulse-based simulation are in [26, 27]. In the latter paradigm, *all* interactions between simulated bodies are affected through frictional collisions, thus a good collision model is crucial to physical accuracy. Our model is similar to that of Routh [33], although we derive equations which are more amenable to numerical integration. Keller also gives an excellent treatment [18], and Bhatt and Koechling give a classification of frictional collisions, based on the flow patterns of tangential contact velocity [5]. Finally, Wang and Mason have studied two-dimensional impact dynamics for robotic applications, based on Routh's approach [36].

Consider two rigid bodies coming into contact as shown in Figure 6. Each body i has a known mass m_i, inertia tensor $\mathbf{J}_i$, linear center of mass velocity $\mathbf{v}_i$, and an angular velocity $\boldsymbol{\omega}_i$. If $\mathbf{r}_i$ is the offset vector of the contact point relative to body i's center of mass, then the absolute velocity $\mathbf{u}_i$ of the contact point on body i is given by

$$\mathbf{u}_i = \mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{r}_i \tag{6}$$

and the relative contact velocity $\mathbf{u}$ at the contact point is given by

$$\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2 \tag{7}$$

Define a collision coordinate system with the z-axis aligned with the contact normal and directed from body 2 to body 1. If the surfaces are not smooth, the normal can be approximated by the displacement vector between the closest points on the bodies. In this coordinate system, the objects are colliding if $\mathbf{u}$ has negative z (i.e. normal) component. In this case, a pair of collision impulses ($\mathbf{p}$ and $-\mathbf{p}$) must be applied to prevent interpenetration; the goal is to compute $\mathbf{p}$. We assume: infinitesimal collision time, the Coulomb friction model, and Poisson's hypothesis for restitution.

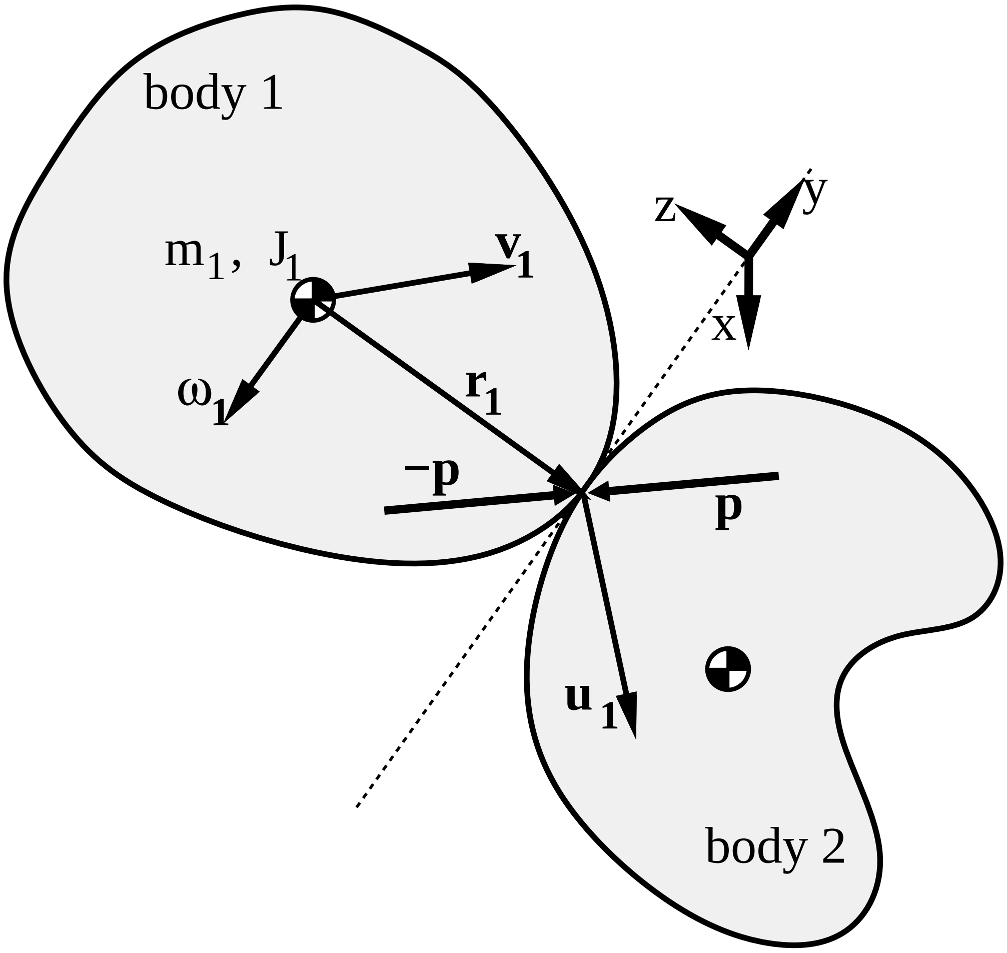

Figure 6: A collision between two rigid bodies.

Infinitesimal collision time implies the positions of the two bodies may be treated as constant during the collision. Since $\mathbf{p}$ is an impulsive force, the velocities of the bodies change during the course of the collision. Because the frictional forces depend on the relative sliding velocity, the velocity profile *during* the collision must be analyzed.

Let γ denote a collision parameter which starts at 0 and increases monotonically during the collision. All body velocities as well as the relative velocity at the contact point are functions of γ. Let $\mathbf{p}(\gamma)$ be the total impulse imparted up to point γ in the collision. From basic physics,

$$\begin{aligned}\Delta\mathbf{v}_1(\gamma) &= \frac{1}{m_1}\mathbf{p}(\gamma) && (8)\\ \Delta\boldsymbol{\omega}_1(\gamma) &= \mathbf{J}_1^{-1}\left[\mathbf{r}_1 \times \mathbf{p}(\gamma)\right]. && (9)\end{aligned}$$

Applying (6) gives

$$\Delta\mathbf{u}_1 = \left[\frac{1}{m_1}\mathbf{I} - \tilde{r}_1\mathbf{J}_1^{-1}\tilde{r}_1\right]\mathbf{p}(\gamma) \qquad (10)$$

where $\mathbf{I}$ is the 3×3 identity matrix and $\tilde{r}_1$ is the canonical skew-symmetric matrix corresponding to $\mathbf{r}_1$. Computing $\Delta\mathbf{u}_2$ analogously ($-\mathbf{p}$ is used instead of $\mathbf{p}$), and applying (7) gives

$$\Delta\mathbf{u} = \underbrace{\left[\left(\frac{1}{m_1} + \frac{1}{m_2}\right)\mathbf{I} - \tilde{r}_1\mathbf{J}_1^{-1}\tilde{r}_1 - \tilde{r}_2\mathbf{J}_2^{-1}\tilde{r}_2\right]}_{\mathbf{K}}\mathbf{p}(\gamma) \tag{11}$$

The 3×3 matrix $\mathbf{K}$ is symmetric. More importantly, the infinitesimal collision time assumption implies $\mathbf{r}_i$ and $\mathbf{J}_i$ are constant during a collision, hence $\mathbf{K}$ is also constant. We can differentiate (11) with respect to γ, obtaining

$$\mathbf{u}' = \mathbf{K}\mathbf{p}' \tag{12}$$

5.1.1 Sliding mode

While the tangential component of $\mathbf{u}$ is nonzero, the bodies are sliding relative to each other, and $\mathbf{p}'$ is completely constrained. Let $\theta(\gamma)$ be the relative direction of sliding during the collision, that is $\theta = \arg(u_x + iu_y)$. Also choose γ to be p_z, the accumulated normal component of impulse. Under Coulomb friction, one finds that

$$\mathbf{p}' = \begin{bmatrix} -\mu\cos\theta \\ -\mu\sin\theta \\ 1 \end{bmatrix}. \tag{13}$$

Expressing the right hand side of (13) in terms of $\mathbf{u}$ and substituting into (12) gives

$$\begin{bmatrix} u_x' \\ u_y' \\ u_z' \end{bmatrix} = \mathbf{K} \begin{bmatrix} -\mu\frac{u_x}{\sqrt{u_x^2+u_y^2}} \\ -\mu\frac{u_y}{\sqrt{u_x^2+u_y^2}} \\ 1 \end{bmatrix}. \tag{14}$$

This nonlinear differential equation for $\mathbf{u}$ is valid as long as the bodies are sliding relative to each other. By integrating the equation with respect to the collision parameter γ (i.e. p_z), one can track $\mathbf{u}$ during the course of the collision. Projections of the trajectories into the u_x-u_y plane are shown in Figure 7 for a particular $\mathbf{K}$.

The basic impulse calculation algorithm proceeds as follows. After computing the initial $\mathbf{u}$ and verifying that u_z is negative, $\mathbf{u}$ is numerically integrated using (14) (p_z is the independent variable). During integration, u_z increases, reaching zero at the point of maximum compression. At this point, p_z is the normal impulse applied during compression, and multiplying it by $(1 + e)$ gives its terminating value, by Poisson's hypothesis for restitution. The integration continues to the terminating value, and $\mathbf{p}$ is recovered by inverting (11).

5.1.2 Sticking mode

Sticking occurs if the relative tangential velocity ever vanishes during integration of (14). In this case, Coulomb friction requires that the frictional force lie within the friction cone, although its direction is not specified. When sticking is detected, the system first checks whether it is a stable sticking condition by setting $\mathbf{u} = (0, 0, \lambda)^T$ in (12), and solving for $\mathbf{p}'$. One can choose λ such that $\mathbf{p}'$ is of the form $\mathbf{p}' = (\alpha, \beta, 1)^T$. If

$$\alpha^2 + \beta^2 \le \mu^2, \tag{15}$$

a frictional force lying within the friction cone can maintain sticking, and so $u_x = u_y = 0$ and $\mathbf{p}' = (\alpha, \beta, 1)^T$ for the duration of the collision.

If $\alpha^2 + \beta^2 > \mu^2$, the friction is not sufficient to maintain sticking, and sliding immediately resumes in a direction θ_e of the ray emanating from the origin in the tangential velocity plot (In Figure 7, $\theta_e \approx 45°$). This ray always exists and is unique in cases of instable sticking.

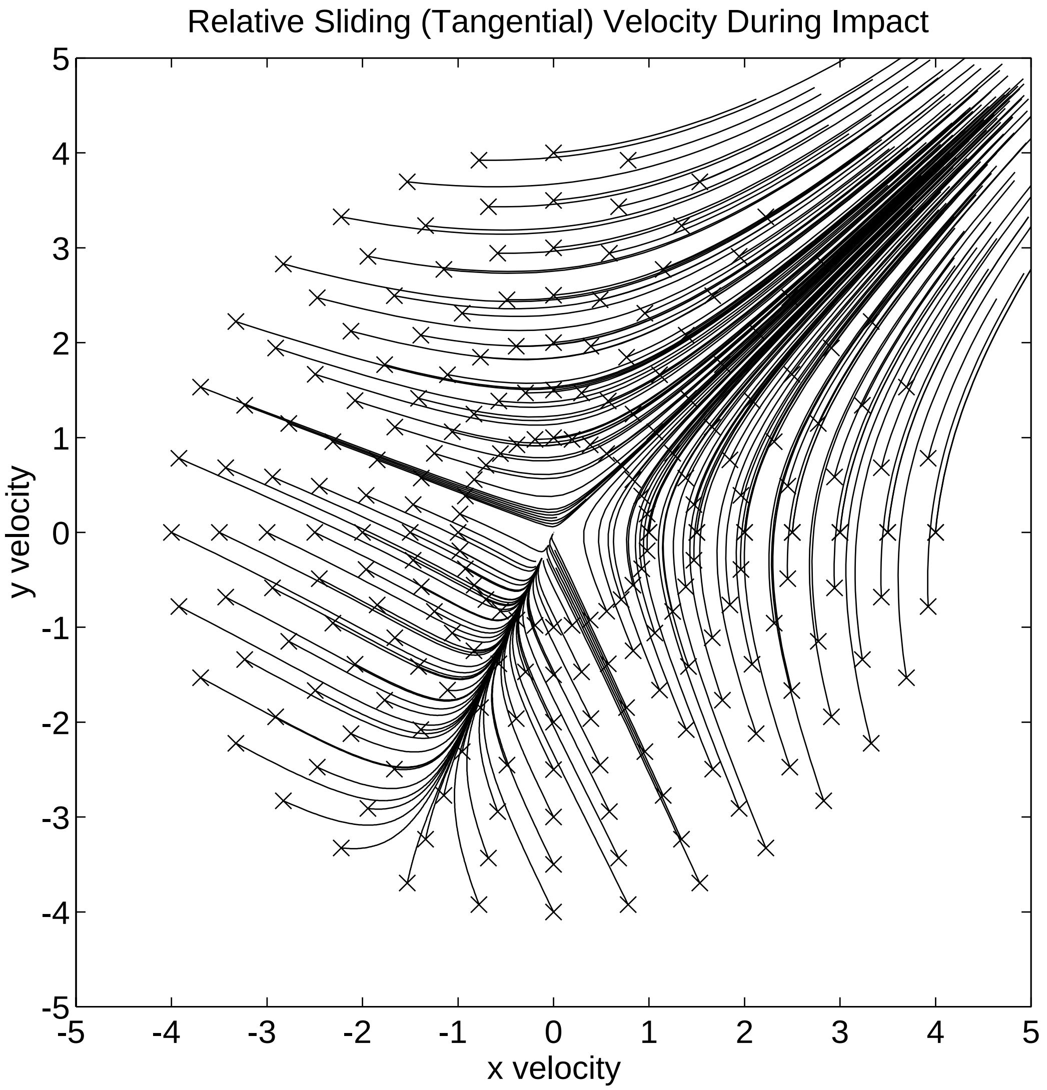

Figure 7: Trajectories of the the tangential components of the system (14) for a particular **K**. The crosses indicate different initial sliding velocities.

5.2 Additional dynamic considerations

The dynamic simulation can take into account parameters that are not considered by the quasi-static and perturbed quasi-static estimators. The flexible feeder system shown in Figure 1 dumps parts from an upper belt onto a lower belt in order to singulate them. Final poses are measured on the lower belt. The precise drop height between belts at the time of the simulation experiments is unknown. We estimate it at 12.0 cm. The horizontal velocity of the parts as they leave the upper belt was estimated at 5.0 cm/s. The coefficients of friction and restitution were both estimated to be 0.3.

Since the parts are in a stable resting state on the upper belt, before being dropped onto the lower belt, the initial distribution of orientations is not uniform, but similar to the final (initially unknown) distribution. To model this, the initial orientations for the first 20 drops are chosen randomly, assuming a uniform distribution over S. Thus we bootstrap the process by estimating the initial pose distribution.

For all remaining drops, the initial (upper belt) poses are chosen from the current estimate of the final pose distribution; the results of each simulation run are then used to re-estimate the initial conditions. A slight perturbation (a rotation of up to 1.5 degrees about a randomly chosen axis) is also applied to the initial pose to introduce noise into the system due to belt vibration. This number is purely a guess; it was not tuned during the experiments, but no attempt was made to estimate it scientifically.

The 20 initial drop tests do not necessarily lead to a unique stationary distribution of intial poses. One improvement over the method used in our experiments would be to verify a proposed initial distribution by performing some small, fixed number of drop tests using it. If the intial and final distributions match well, the initial distribution is at least stationary and therefore a reasonable guess. Otherwsie, the bootstrapping and verification should be repeated. Alternatively, one could perform the boostrapping and verification steps multiple times, and choose the most stationary distribution to seed the initial distribution for the rest of the drop tests.

6 Experimental results

All of the estimators described in this paper were applied to four test parts. The parts were all small, plastic, rigid parts, of the type typically used in automated assembly as shown in Figure 2. For the dynamic simulation, 2000 drops were simulated requiring approximately two hours of computation per part.

Part #1 is an insulator cap purchased at a local hardware store. Parts #2, #3, and #4 are pushbuttons designed for a commercial car stereo system. Geometric models of each part were constructed by measuring the parts with a ruler. Centers of mass and moments of inertia for the parts were computed automatically by *Impulse*.

The system shown in Figure 1 was used to perform 3595 physical drop trials. Tables 1 through 4 show the results. All quantities in the tables are percentages.

The error percentages included in the tables indicate the overall performance of each estimator for each sample part. They are computed as the average deviation of the estimator's predictions from the physical test percentage, weighted by the frequency with which that state actually occurs.

Let $p_1, \ldots, p_n$ represent the probability of each of n states, as measured in the physical test. Let $a_1, \ldots, a_n$ represent the corresponding probabilities computed by one of the estimators. The error percentage for that estimator is given by

$$e = 100 \sum_{i=1}^{n} p_i \, |a_i - p_i|. \tag{16}$$

6.1 Discussion

The quasi-static and perturbed quasi-static estimators are extremely fast, requiring less than a second of computation time for parts with 50 facets. Dynamic simulation is slower; for each part, 2000 drops were simulated, taking approximately two hours per part. The data presented in Tables 1 through 4 bring out several interesting points concerning the accuracy of the estimators' predictions.

The perturbed quasi-static estimator's predictions are consistently more accurate than those of the quasi-static estimator, and the added computation time is negligible. Hence, the perturbed quasi-static estimator should always be chosen over the quasi-static one.

Pose	Quasi- Static	P'turbed Q–S	Dynamic Sim.	Physical Tests[a]
1	30.5	46.5	41.9	46.0
2	37.3	30.2	26.2	27.1
3	19.6	19.8	28.3	19.7
4	8.3	3.5	3.0	5.0
5	4.2	0.0	0.8	2.2
error	10.1	1.2	4.0	–

[a] 1036 trials

Table 1: *Orange insulator cap data.*

Pose	Quasi- Static	P'turbed Q–S	Dynamic Sim.	Physical Tests[a]
1	34.5	48.8	71.7	75.8
2	39.9	30.6	20.9	13.8
3	19.2	20.5	7.4	10.5
4	6.3	0.0	0.1	0.0
error	35.8	23.8	4.4	–

[a] 545 trials

Table 2: *White stereo button data.*

The dynamic simulation estimator is the most accurate for all sample parts, except the insulator cap (Table 1), for which the perturbed quasi-static estimator slightly outperforms it. The dynamic simulation estimator's prediction accuracy is also the most consistent; the composite error is less than 5% in all cases. Nonetheless, a penalty of three to four orders of magnitude in execution time must be paid for this added accuracy; whether this is appropriate or not depends on the situation.

In an interactive setting, where a designer is perhaps editing the CAD model of a part in order to improve feeder throughput,

Pose	Quasi- Static	P'turbed Q–S	Dynamic Sim.	Physical Tests[a]
1	36.2	47.3	54.1	56.0
2	16.0	25.5	24.1	24.5
3	17.4	17.0	14.0	13.6
4	8.1	1.2	1.4	4.4
5	10.6	4.5	5.3	1.4
6	7.5	4.4	1.0	0.3
7	4.3	0.0	0.3	0.0
error	14.0	5.8	1.4	–

[a] 1099 trials

Table 3: *Rectangular black stereo button data.*

Pose	Quasi- Static	P'turbed Q–S	Dynamic Sim.	Physical Tests[a]
1	35.7	46.6	68.4	62.2
2	17.5	15.5	16.6	15.2
3	12.1	17.0	6.1	11.0
4	7.2	8.6	6.0	4.7
5	3.9	1.6	2.7	3.1
6	5.6	1.5	0.0	2.8
7	3.8	3.9	0.3	0.5
8	4.2	1.7	0.0	0.0
9	3.0	2.3	0.0	0.0
10	2..6	0.7	0.0	0.0
11	2.2	0.0	0.0	0.0
12	2.1	0.5	0.0	0.0
error	17.2	10.7	4.8	–

[a] 915 trials

Table 4: *Square black stereo button data.*

the perturbed quasi-static estimator is clearly the best choice. The designer need only wait seconds to see how changing a part's CAD model alters the pose distribution and feeder throughput.

The dynamic simulation estimator is useful for obtaining a more accurate estimate once the design has been determined, or for analyzing the effects of more subtle design changes. It models several factors that are not considered by the standard and perturbed quasi-static estimators, including: friction, collisions with energy loss, mass moments of inertia, height of drop, and initial conditions of the part prior to drop. To study the effects of varying these parameters, dynamic simulation is appropriate.

The quasi-static methods are based on a uniform distribution of initial orientations. In the common case where the true distribution is unknown, this is a reasonable guess. In contrast, dynamic simulation can produce results for other distributions just by choosing the random initial orientations from the distribution. If knowledge about the distribution is known, dynamic simulation has an added accuracy advantage over the quasi-static methods.

Our simulation experiments involved 2000 drop tests. Often, fewer trials may be sufficient, reducing the computational cost of this method. Suppose the true (unknown) probability that a part lands in a particular pose is p. The number of times the part lands in this pose over n trials is a binomial random variable, which may often be approximated by a normal distribution[1]. A *confidence interval* statement is of the form: "p lies within the range $(\mu - \delta, \mu + \delta)$, with $100(1-\alpha)\%$ certainty." Here, μ is the probability estimate obtained from the n trials, δ is the allowable error tolerance, and α is the *level* of the statistical test. Given δ and α, one can bound the number of trials necessary by

$$n = \frac{\Phi^{-1}(1 - \frac{\alpha}{2})}{2\delta}, \tag{17}$$

where $\Phi(x)$ is the cumulative normal distribution function. For example, to pinpoint the probability of a particular final pose to within 5%, with 90% certainty, $\delta = 0.05$ and $\alpha = 0.10$. From (17), 385 trials are sufficient. See [13] for more information.

[1] A common rule of thumb is that the normal approximation is valid if the numbers of successes and failures during the trial series both exceed five [13].

7 Summary

Predicting the pose distribution of rigid parts dropped onto a flat surface is important in evaluating part designs for assembly. These distributions are necessary to estimate feeder throughput, which can then be used to determine how many robots and assembly lines are required to meet specified production rates. This can greatly reduce the time required to set-up or changeover automated factories and hence allow new products to be more rapidly brought to market.

We have presented three estimators for predicting the pose distributions of rigid parts dropped onto a flat surface. We have compared the predictions from these estimators to physical test results, and believe that this is the first systematic comparison of pose estimators with experiments using real industrial parts.

Our results indicate that a perturbed quasi-static estimator, based on a refinement of the quasi-static estimator presented in [38], produces significantly more accurate results, with negligible added computation time. The perturbed quasi-static estimator certainly has the highest accuracy to execution time ratio of all three estimators studied. The third estimator, based on dynamic simulation of the dropped parts using the simulator *Impulse*, generally gives the most accurate predictions, with averaged errors under 5% for all four test parts. This estimator can also be used to study sensitivities to parameters not modeled by the other estimators, such as the coefficient of friction or the initial part velocity. However, this estimator takes one to two hours to generate predictions, as opposed to under a second required by the standard and perturbed quasi-static estimators. In an interactive setting, the quasi-static estimator is the method of choice, providing reasonably good predictions very quickly. The dynamic simulation estimator might find application later in the design cycle, where more careful analysis is required.

This work in estimating pose statistics complements other ongoing work in automated assembly. Rao, Kriegman, and Goldberg have studied the use of a pivoting gripper for Adept's flexible feeder; assuming part pose is known, they give an $O(m^2 n \log n)$ estimator to generate pivot grasps for a part with n faces and m stable configurations [32].

The perturbed quasi-static estimator described in this paper has been incorporated into a commercial simulation package where it is used to predict part behavior for rapid design of modular workcells such as the one shown in figure 1. In the future, part pose statistics can be used in CAD systems, allowing users to alter the design of parts to achieve a desired distribution of poses.

Acknowledgments. Part of this research was performed when Goldberg was at the University of Southern California. Versions of the quasi-static estimator were implemented by Anil Rao and Jeff Wiegley in 1992, Zheng Yeh in 1993, and Yan Zhuang in 1994. The convex hull routine used in our current implementation is due to Ioannis Emiris and John Canny. Thanks to Rob Zanutta and Isam Nessas for help with experiments and to Randy Brost and Bruce Shimano for useful feedback during the research.

References

[1] S. Akella, W. Huang, K. Lynch, and M. Mason. Planar manipulation on a conveyor by a one joint robot with and without sensing. In *Second Workshop on Algorithmic Foundations of Robotics*, Toulouse, France, July 1996.

[2] Srinivas Akella and Matthew T. Mason. An open-loop planner for posing polygonal objects in the plane by pushing. In *International Conference on Robotics and Automation*. IEEE, May 1992.

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

[4] R. P. Berrety, M. Overmars, F. Van der Stappen, and K. Goldberg. On fence design and the complexity of push plans for orienting parts. In *13th Symposium on Computational Geometry*, Nice, France, June 1997. ACM.

[5] Vivek Bhatt and Jeff Koechling. Classifying dynamic behavior during three dimensional frictional rigid body impact. In *Proceedings of International Conference on Robotics and Automation*. IEEE, May 1994.

[6] Geoffrey Boothroyd, Corrado Poli, and Laurence E. Murch. *Automatic Assembly*. Marcel Dekker, Inc., 1982.

[7] Randy C. Brost. *Analysis and Planning of Planar Manipulation Tasks*. PhD thesis, CMU, January 1991.

[8] Mike Caine. The design of shape interactions using motion constraints. In *IEEE International Conference on Robotics and Automation*, 1994.

[9] Brian Carlisle, Ken Goldberg, Anil Rao, and Jeff Wiegley. A pivoting gripper for feeding industrial parts. In *International Conference on Robotics and Automation*. IEEE, May 1994.

[10] G. Causey and Roger Quinn. Design of a flexible parts feeding system. In *IEEE International Conference on Robotics and Automation*, 1997.

[11] A. Christiansen, A. Edwards, and C. A. Coello Coello. Automated design of parts feeders using a genetic algorithm. In *IEEE International Conference on Robotics and Automation (ICRA)*, Minneapolis, MN, 1996.

[12] A. Deguet, A. Joukhadar, and C. Laugier. Models and algorithms for the collision of rigid and deformable bodies. In Pankaj K. Agarwal, Lydia Kavraki, and Matt Mason, editors, *International Workshop on Algorithmic Foundations of Robotics*. A. K. Peters, Ltd, Wellesley, MA, 1998.

[13] Jay L. Devore. *Probability & Statistics for Engineering and the Sciences*. Brooks/Cole Publishing Company, Monterey, California, 1982.

[14] Michael A. Erdmann and Matthew T. Mason. An exploration of sensorless manipulation. In *IEEE International Conference on Robotics and Automation*, 1986. Also appears in IEEE Journal of Robotics and Automation, V. 4.4, August 1988.

[15] Ken Goldberg. *Stochastic Plans for Robotic Manipulation*. PhD thesis, CMU School of Computer Science, August 1990. Copies can be obtained from Catherine Copetas (copetas@cs.cmu.edu, 412-268-8525).

[16] D. Gudmundsson and K. Goldberg. Estimating and optimizing throughput of a robotic part feeder using queueing theory. In *International Conference on Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[17] M. Jakiela and J. Krishnasamy. Computer simulation of vibratory parts feeding and assembly. In *2nd International Conference on Discrete Element Methods*, March 1993.

[18] J. B. Keller. Impact with friction. *Journal of Applied Mechanics*, 53, March 1986.

[19] L. Lim, B. Ngoi, S. Lee, S. Lye, and P. Tan. A computer-aided framework for the selection and sequencing of orientating devices for the vibratory bowl feeder. *International Journal of Production Research*, 32(11), 1994.

[20] Kevin Lynch. The mechanics of fine manipulation by pushing. In *International Conference on Robotics and Automation*. IEEE, May 1992.

[21] Matthew T. Mason, Ken Goldberg, and Yu Wang. Progress in robotic manipulation. In *15th Grantees Conference on Production Research and Technology*. National Science Foundation, January 1989.

[22] G. Maul and N. Jaksic. Sensor-based solution to contiguous and overlapping parts in vibratory bowl feeders. *Journal of Manufacturing Systems*, 13(3), 1994.

[23] G. Maul and M. Thomas. A systems model and simulation of the vibratory bowl feeder. *Journal of Manufacturing Systems*, 16(5), 1997.

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

[25] Brian Mirtich. *Impulse-based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, December 1996.

[26] Brian Mirtich and John Canny. Impulse-based dynamic simulation. In K. Goldberg, D. Halperin, J.C. Latombe, and R. Wilson, editors, *The Algorithmic Foundations of Robotics*. A. K. Peters, Boston, MA, 1995. Proceedings from the workshop held in February, 1994.

[27] Brian Mirtich and John Canny. Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, pages 181–188, New York, 1995. ACM Press.

[28] Paul Moncevicz, Mark Jakiela, and Karl Ulrich. Orientation and insertion of randomly presented parts using viratory agitation. In *ASME 3rd Conference on Flexible Assembly Systems*, September 1991.

[29] James L. Nevins and Daniel E. Whitney. Computer-controlled assembly. *Scientific American*, 1978.

[30] B. K. A. Ngoi, L. E. N. Lim, and S. S. G. Lee. Analyzing the probabilities of natural resting for a component with a virtual resting face. *ASME Journal of Engineering for Industry*, 1995.

[31] Michael A. Peshkin and Art C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5), October 1988.

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

[33] Edward J. Routh. *Elementary Rigid Dynamics*. Macmillan, London, 1905.

[34] A. C. Sanderson. Parts entropy methods for robotic assembly system design. In *International Conference on Robotics and Automation*. IEEE, 1984.

[35] M. Shirai and A. Saito. Parts supply in sony's general purpose assembly system: SMART. *Japan Journal of Advanced Automation Technology*, 1(108), 1989.

[36] Yu Wang and Matthew T. Mason. Modeling impact dynamics for robotic operations. In *Proceedings of International Conference on Robotics and Automation*, pages 678–685. IEEE, May 1987.

[37] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. *Assembly Automation*, 17(2), August 1997.

[38] Jeff Wiegley, Anil Rao, and Ken Goldberg. Computing a statistical distribution of stable poses for a polyhedron. In *The 30th Annual Allerton Conference on Communication, Control, and Computing*. University of Illinois at Urbana-Champaign, October 1992.

[39] W. Wolfson and S. Gordon. Designing a parts feeding system for maximum flexibility. *Assembly Automation*, 17(2), 1997.