Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

Designing Parts Feeders Using Dynamic Simulation

Dina R. Berkowitz*
`dberk@cs.berkeley.edu`

John Canny*
`jfc@cs.berkeley.edu`

Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720

Abstract

We consider the problem of designing traditional (e.g. vibratory bowl) feeders for singulating and orienting industrial parts. Our ultimate goal is to prototype new designs using analytically- and geometrically-based methods.

We have developed a tool for designing industrial parts feeders based on dynamic simulation. Our tool allows us to automatically perform multiple feeder design experiments, and to evaluate their outcomes. These results can then be used to compute the probabilities of a Markov model for the feeder. To demonstrate our technique, we present preliminary results for the design of two simple feeders. Our findings suggest that using dynamic simulation is a promising approach for designing parts feeders.

1 Introduction

Vibratory bowl feeders and hopper feeders have proliferated industry as a cost-effective means for reliably orienting parts. These feeders and their transfer conveyors account for nearly one-third of the cost and failure risk of an assembly system [8]. However, the current design of these feeders is a "black art", based merely on modifications to previous designs and empirical debugging rather than on theory and automated design.

Parts are presented to feeders in bulk, and as a result, clustering and entanglement are common. Singulating and orienting parts is a significant problem, due largely to the non-general feeder design solutions that must be developed for the individual parts. The complexity of the parts and the feeder, the number of parts, and the absence of good impact friction models in the literature add to this formidable task. These inflexibilities often result in a 7-12 month turn-around time for each new feeder system, even for feeders that orient the most similar of parts [2].

Our focus in this work is to use dynamic simulation to expedite the design process and to make it more flexible, efficient and robust. We are using Mirtich's novel near real-time impulse-based dynamic simulator, *Impulse* [14; 15]. Impulse was expressly developed to simulate parts feeders. It was designed to represent many colliding rigid bodies, and is founded on a new friction-based simulation paradigm that models impacts more accurately. Impulse's stable pose predictions have been shown to accurately characterize the dynamics found in industrial tasks such as parts feeding [16].

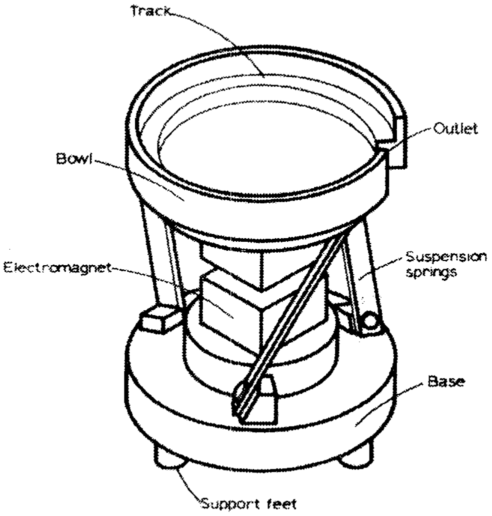

Figure 1: A Typical Vibratory Bowl Feeder for Orienting Industrial Parts. (Reprinted from Boothroyd [4] p. 32 by courtesy of Marcel Dekker, Inc.)

The primary contribution of this paper is to develop parameter enumeration, analysis, and Markov model-building tools. We have developed a tool that allows us to easily generate and test suites of feeder designs with different characteristics and initial conditions. Our tool also evaluates the generated designs automatically. Once a good feeder design has been found, the tool can be used again to simulate a particular feeder over all stable poses of the part. The results of these experiments yield probabilities that allow us to build a Markov model of the feeder.

We have used our tool to simulate simple parts feeders modelled after vibratory bowl feeders. We first describe our parts feeding model. We then summarize how our tool is used in practice to drive design experiments enumerated over a user-specified parameter space. Next we discuss how our tool automatically evaluates a design. Finally, we present a block feeder design case study and some preliminary results of a cap feeder simulation.

---

*Financial support provided by National Science Foundation Grant #FD93-19412 and Advanced Research Projects Agency Grant #B454.

0-7803-2988-4/96 $4.00 © 1996 IEEE

1.1 Related work

Our objective is to formalize the automatic and reliable design of sensorless parts feeders. This problem has been addressed by a number of researchers.

Boothroyd has done seminal work on characterizing industrial parts feeders. With Poli and Murch, he developed a taxonomy of industrial parts and feeders for orienting them [5].

Natarajan introduced several formal paradigms for designing sensorless parts feeders [17]. Motivated by industrial palletizing trays that "sift" parts into desired orientations, Erdmann, Mason, and Vaněček proposed a sensorless table tilting planner that oriented three-dimensional polyhedral parts [10]. The work in both of these papers describes transition graphs similar to our Markov model.

Others have used analytical techniques to uniquely orient streams of singulated parts. Goldberg presented a design for a sensorless programmable parts feeder. The work describes an algorithm that finds a sequence of gripper actions for orienting a given polygonal part [11]. Brokowski, Peshkin, and Goldberg described the use of curved fences above a moving conveyor [6].

It is very time-intensive and costly to build prototypes of feeder designs. Researchers have proposed simulation as a technology for making the designer's job more efficient and effective. Jakiela and Krishnasamy discussed a scheme for two-dimensional simulation of vibratory parts feeding [13]. Caine used the configuration space paradigm to develop an interactive system for simulating and manipulating designs [7]. Our tool uses three-dimensional dynamic simulation to automatically perform feeder experiments and evaluate their outcomes.

Some recent research includes the use of vibratory motion to manipulate parts. Böhringer, Bhatt, and Goldberg developed sensorless strategies for orienting parts by generating and changing the dynamic modes of a vibrating plate on which the parts rest [3]. Christiansen, Edwards, and Coello Coello developed a genetic algorithm to automate the design of traditional vibratory bowl feeders [9]. Our system computes feeder gate probabilities that are similar to the inputs required by their algorithm.

2 Our Feeder Model

Our feeder model is based on vibratory bowl feeders. A typical vibratory bowl feeder consists of a hopper (or bowl) mounted to a heavy base by suspension springs. An electromagnet mounted between the hopper and the base produces vibratory motion constrained by the leaf springs (see Figure 1). The constrained motion results in torsional vibration about the feeder's vertical axis coupled with translational vibration in its vertical direction.

Parts presented in bulk at the bottom of the bowl travel up a helical track as the bowl vibrates. The track has various features such as protrusions and floor cut-outs. These features or "gates" serve to reorient the parts in stages. Each gate has an orientation precondition and postcondition. Parts that do not meet a precondition are rejected and automatically thrown back to the bottom of the bowl to be recycled. The postcondition of each gate is the precondition of the next one; the postcondition of the final gate is the feeder's goal orientation.

There are two fundamental differences between our model and actual vibratory bowl feeders. First, we model a vibratory bowl feeder as a single straight track formed by unravelling the bowl's long helical track. We could model a curved track, but we are also interested in novel narrow-footprint feeder designs using straight tracks. Second, we use gravitational feed rather than vibrational feed. The lack of vibration is forced by limitations of the current simulator (but vibratory capability will be added in the future). However, we believe that neither of these assumptions affects the primary contributions of this paper.

2.1 A Markov Representation

To model the behavior of the system, we represent the effects produced by the gates as transitions in a non-deterministic finite automaton (NFA), where the states correspond to stable part orientations. By labeling the transitions (or edges) of the NFA with probabilities, we get a Markov model of the feeder. This allows us to compute the probability that a part in a particular initial orientation will end up in the desired final orientation. In a similar way, Boothroyd uses probability matrices to compute the effects of feeder gates on parts [4].

As long as a feeder's gates are far enough apart not to interact, we can study their effects independently. By simulating each gate with our tool in isolation, we can compute the probability for each pre- and post-orientation that the gate will convert one into the other. Once we have a Markov model for each gate, we can "chain" the gate models together to get a model for the entire feeder.

Figure 2 shows an example Markov model for the feeder in Figure 3. In this example, the desired behavior of the feeder is to uniquely orient rectangular blocks beginning in one of six initial orientations. The nodes represent quantizations of the stable orientations *Flat Lengthwise, Flat Crosswise, Erect Lengthwise, Erect Crosswise, On-Edge Crosswise, and On-Edge Lengthwise* (labelled FL, FC, EL, EC, OEC, and OEL, respectively) of a rectangular block (see Figure 4). The model is non-deterministic because each discrete state represents infinitely many actual orientations. Asfahl presents a case study of this particular vibratory bowl feeder [1].

In the digraph of Figure 2, transitions between successive levels correspond to feeder gates. The transitions out of the initial state represent the introduction of a block onto the feeder track. They are labelled e (for epsilon) because the parts do not go through any gates at this point. The following phases orient the block as follows: *Wiper Blade* Gate 1 (G1) rejects erect blocks, *Narrowed Track* Gate 2

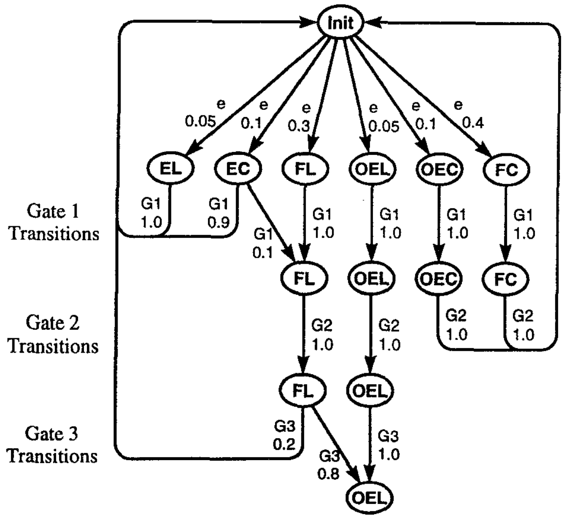

Figure 2: Markov Model Representing Three Consecutive Gates (G1, G2, and G3) in a Vibratory Bowl Feeder.

(G2) rejects both remaining crosswise orientations (FC and OEC) of the block, and *Edge Riser* Gate 3 (G3) lifts flat blocks (FL) to the final On-Edge Lengthwise orientation.

The graph transitions are also labelled with their probabilities of being taken; thus the sum of the probabilities of each node's out-edges is 1.0. The probability that a part will follow a path is the product of the probabilities along that path. Hence, we can compute the probability that a part reaches the final state by summing the probabilities of paths from the start state to the final state.

In this example, there are three paths from the initial to the final state that do not involve a rejection back to the initial state. Summing over these paths, we compute that the probability of success in one pass is 29.8%. The odds of succeeding in at most two passes is 50.7%.

3 Tool Overview

Our tool was developed so that we could easily generate multiple simulations with varying initial conditions and evaluate their output results automatically. The tool's declarative input language is interpreted, so the turn-around time between experiments is low. The tool currently allows us to search and evaluate large spaces of feeder designs by specifying ranges for each of the feeder's parameters. We are working on implementing automatic search techniques driven by our tool's evaluation function.

The tool's input file specifies the configuration of the system to be simulated. This includes descriptions of all of the objects (fixed and moving) and their corresponding characteristic and physical properties, such as geometry, pose, mass, initial linear and angular velocities, friction, and restitution. These characteristics may be parameterized, so that one specification can actually represent an arbitrary number of feeder designs. The values for the parameters are bounded by associating an interval with each one. The tool automatically generates multiple simulations by forming the Cartesian product of the parameter intervals. Each simulation is evaluated by the tool's evaluation function, and the results are presented to the user.

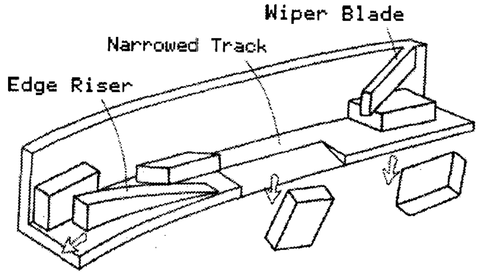

Figure 3: Orientation of Rectangular Blocks in a Vibratory Bowl Feeder. (Reprinted - with slight modifications - from Boothroyd [5] by courtesy of U. Mass., Amherst).

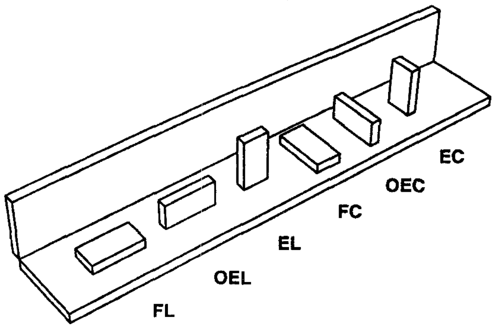

Figure 4: The Six Stable Block Orientations: Flat Lengthwise (FL), On-Edge Lengthwise (OEL), Erect Lengthwise (EL), Flat Crosswise (FC), On-Edge Crosswise (OEC), and Erect Crosswise (EC).

We would like to explore various parameter spaces using dynamic simulation. This will help us to analytically determine what constitutes an effective design for orienting a set of parts.

We have parameterized several aspects of a feeder design; these include the poses of the feeder gates such as the Wiper Blade, Narrowed Track, Edge Riser (a wedge), and *Wall Projection* (see Figures 3 and 5), and feeder physical properties such as friction and restitution. Since the tool explores the Cartesian product of the parameter intervals, the parameter space over which designs are evaluated is multidimensional.

4 Design Evaluation

Our tool automatically evaluates a parts feeder design and returns a real number corresponding to the "goodness" of

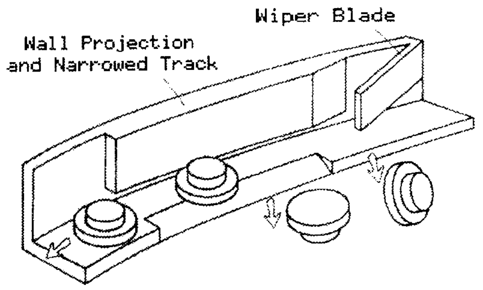

Figure 5: Orientation of Caps in a Vibratory Bowl Feeder. (Reprinted - with slight modifications - from Boothroyd [5] by courtesy of U. Mass., Amherst).

the design. This evaluation is a weighted sum of metrics based on the part's position, orientation, linear velocity and angular velocity as it exits the feeder. The metrics are computed by comparing the part's final state (as it passes through a particular precomputed plane in the feeder) to an "ideal" state. The objective value is the absolute "difference" between the actual and ideal states, so zero indicates a perfect design.

An example ideal state is one in which the part is travelling perfectly aligned along the feeder wall, completely contacting the feeder track, with no velocity components away from the track or wall, and with no angular velocity as it exits the feeder. Currently, the metric weights are chosen based on each metric's potential range and its relative importance in the final weighted sum. For example, the orientation metric is weighted more heavily than the linear velocity magnitude metric, since we are more concerned that the part's final orientation is correct than we are that the part's speed has a particular magnitude.

In addition to evaluating the feeder based on its "local" behavior on a part's trajectory, we use another metric to gain an overall measure of the feeder's design. This general criterion for evaluating a mechanical parts feeder is known as "efficiency", and is defined as follows:

$$efficiency = \frac{F_{out}}{F_{in}} \qquad (1)$$

where F_{out} is the output feed rate of correctly oriented parts and F_{in} is the total input feed rate [1].

Future work will include a more analytic and automatic method for determining metric weights. We are also exploring more sophisticated criteria and formal methods for evaluating feeder designs. For instance, we could take part path efficiency into account; two feeders might be able to produce the same output for a given input, but the number and complexity of motions each oriented part takes might be lower for one of the feeders. Natarajan has considered some computational issues involved in automating the design of parts feeders [17].

5 Results

We have used our tool to simulate two simple feeders: a block feeder with a single gate similar to the Edge Riser of Figure 3, and a cap feeder containing gates similar to the Wall Projection and Narrowed Track of Figure 5. In this section, we describe the results of our experiments.

5.1 A Block Feeder Design Case Study

We designed a simple feeder to orient a rectangular block starting in the initial Flat Lengthwise or On-Edge lengthwise orientation. Our track contains an Edge Riser gate, which flips the block up to the On-Edge Lengthwise orientation.

For each of the six stable orientations of the block (see Figure 4), we simulated a suite of 4,620 trials. Our experiments included two parameters: the angle *WedgeRz* of the Edge Riser rotated about a normal to the track floor, and the initial distance *BlockTx* of the block's center from the Edge Riser measured along the track floor's major axis (see Figure 6).

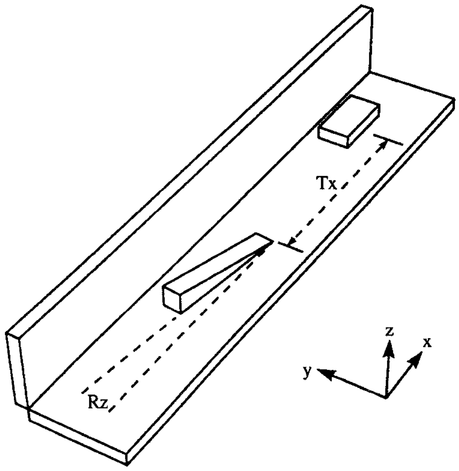

Figure 6: The feeder parameters *WedgeRz* (Rz) and *BlockTx* (Tx).

We varied WedgeRz between -10 and 0 degrees in increments of one degree (with more negative rotations causing the high end of the wedge to be closer to the wall), and BlockTx between 40 and 80 centimeters in steps of two. Each pair of parameters corresponds to a single *experiment*. For each experiment, we ran 20 *trials*, perturbing the initial orientation of the part by a small random amount for each trial. We computed the outcome of each experiment as the mean of the outcomes of its trials.

The results from these experiments correlate fairly well with our predictions of the Edge Riser's behavior on the blocks' trajectories for the various starting orientations. Figure 7 shows the objective value for each pair of parameter values when the block was started in the Flat Lengthwise orientation. As indicated by the trough visible in this figure,

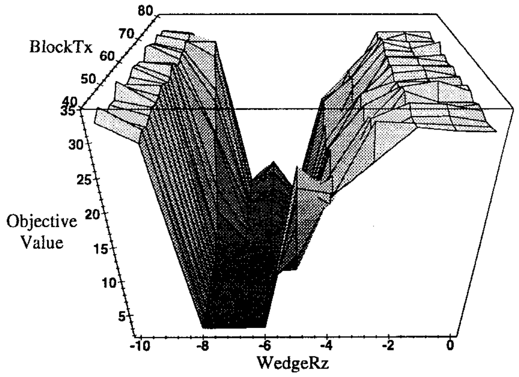

Figure 7: Objective Function Results for the Flat Lengthwise Block Experiment.

a block starting in the Flat Lengthwise orientation is almost ideally oriented for WedgeRz values of -7 to -6, especially for BlockTx values between 40 and 60.

Why didn't the Edge Riser work for other WedgeRz angles? By viewing the simulations for larger and smaller values of WedgeRz, we discovered that the gate was failing for two different reasons. When the wedge angle was too steep, the block was lifted rather than rotated, and flew off the ramp in the Flat Lengthwise orientation. When the wedge angle was too shallow, the edge of the block was lifted, but there was not enough angular momentum to stand it on its edge, so it simply fell back to the Flat Lengthwise orientation.

We do not have enough space to include plots for the five other initial orientations, but there were no surprises. The On-Edge Lengthwise orientation performed perfectly for all WedgeRz values except -10 degrees. By watching simulations for this case, we saw that there was too little clearance for the block to pass, so it was getting wedged between the Edge Riser and the wall. The Erect-Crosswise orientation performed well for only a very small number of trials in which the erect block fell over into the Flat-Lengthwise orientation as it approached the Edge Riser. These results are not surprising since the feeder was specifically designed to raise Flat Lengthwise blocks to the On-Edge Lengthwise orientation. As expected, the other three starting orientations performed poorly in this feeder.

Figure 7 shows the feeder's average performance, but it does not show how much the performance varied within each experiment. Consider the set of successful experiments corresponding to the valley WedgeRz = -7 of the same figure. The top plot of Figure 8 shows the mean values and standard deviations of these experiments.

It is somewhat troubling that some of the standard deviations in the top plot of Figure 8 are so large. However, this is because the trial outcomes for a given experiment tend to be bi-modal. The bottom plot in Figure 8 shows the results for the individual trials. Most results are clustered toward the mean, with an occasional outlier.

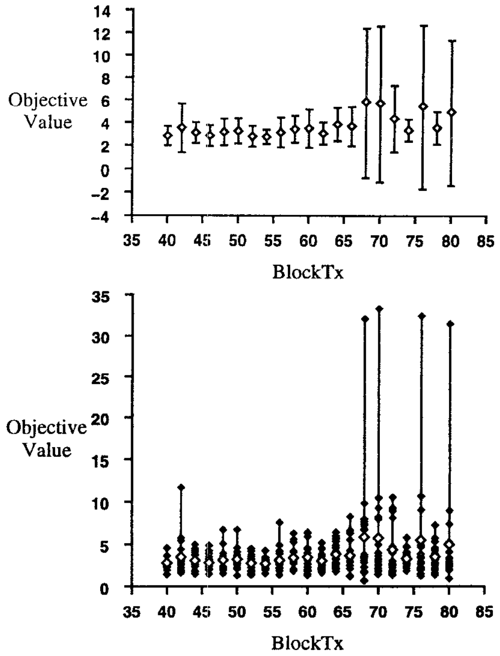

Figure 8: Objective Function Results for the Flat Lengthwise Block Experiment with WedgeRz = −7. The hollow diamonds are the mean evaluation results over 20 slightly perturbed trials. The error bars in the top plot show the standard deviation, while the filled diamonds in the bottom plot show the results of the individual trials.

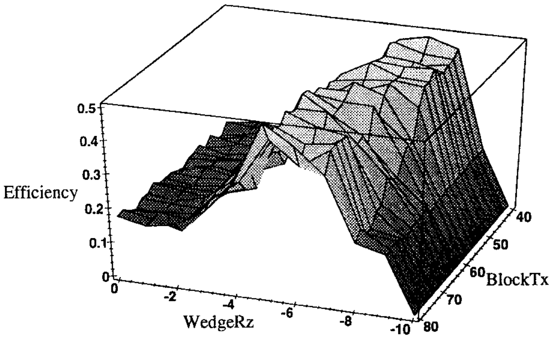

Figure 9: The Efficiency of the Block Feeder. For each feeder configuration, the height of the surface is the efficiency of the feeder over all six initial block orientations.

Figure 9 shows the efficiency (given by Equation 1) of each feeder parameterization over all initial orientations of the block. Recall that the efficiency is the feeder's probability of successfully orienting parts. In constructing this figure, we considered an outcome to be successful if it produced a corresponding mean objective value of at most 5.0 (zero indicating a perfect design). The figure indicates that the efficiency is more directly related to the gate's angle WedgeRz than the block's translation along the feeder track

BlockTx. The best efficiency is approximately 50% for values of WedgeRz between -6 and -7. This low overall efficiency is not surprising given that this one-gated feeder was designed to handle only Flat Lengthwise and On-Edge Lengthwise blocks. Over these two orientations only, the feeder's efficiency is nearly 100%.

5.2 A Cap Feeder

We have designed a simple feeder consisting of a Wall Projection and Narrowed Track (see Figure 5) that orients caps. The parameters for this feeder are the height of the Wall Projection and the width of the Narrowed Track. Preliminary experiments yielded fairly reliable results for orienting caps. We are working on running a more complete suite of trials over a uniform distribution of cap orientations for this feeder.

6 Conclusions

We have presented a tool based on dynamic simulation for doing parameter enumeration, analysis, and Markov model-building of parts feeders. It allows us to easily characterize a parts feeder by a small set of parameters, to simulate feeders over large ranges of parameter values and thousands of experiments, and to automatically analyze the results. These results can be used to determine the edge probabilities of the feeder's Markov model.

Results from two simple parts feeder designs indicate that dynamic simulation may be a feasible method for determining good feeder configurations. The time required to generate and analyze thousands of experiments takes on the order of hours rather than the weeks or months it currently takes to prototype new designs in industry. Simulations of good designs are visual and can be "replayed". Mirtich et al. have shown that Impulse's stable pose experiments accurately predict part dynamics [16]. However, future work is required to show the accuracy of simulation results when run on actual feeders. We are also exploring various search techniques for finding optimal feeder designs.

Acknowledgments

The authors wish to thank Allan Heydon and Greg Nelson for help with their constraint-based drawing editor, Juno-2, which was used to draw Figures 2, 4, 6, and 8 [12]. The authors would also like to thank Brian Mirtich for comments on a draft of the paper.

References

[1] C. Ray Asfahl. *Robots and Manufacturing Automation*. John Wiley and Sons, New York, second edition, 1992.

[2] Bodine. Personal Communication, 1994. Bodine Corporation, located in Bridgeport CT, sells feeder and assembly systems.

[3] Karl-Friedrich Böhringer, Vivek Bhatt, and Kenneth Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *International Conference on Robotics and Automation*. IEEE, May 1995.

[4] Geoffrey Boothroyd. *Assembly Automation and Product Design*. Marcel Dekker, Inc., New York, 1992.

[5] Geoffrey Boothroyd, Corrado R. Poli, and Laurence E. Murch. *Handbook of Feeding and Orienting Techniques for Small Parts*. Department of Mechanical Engineering, University of Massachusetts, Amherst, MA, 1976.

[6] Mike Brokowski, Michael A. Peshkin, and Ken Goldberg. Curved fences for part alignment. In *International Conference on Robotics and Automation*. IEEE, May 1993.

[7] Michael E. Caine. *The Design of Shape from Motion Constraints*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1993. Ph.D. Thesis, AI TR-1424 is a revised version of a thesis submitted to the Department of Mechanical Engineering.

[8] Brian R Carlisle. Robot technology and system integration. In *International Symposium of Experimental Robotics*. Stanford University, June 1995. Keynote address at the symposium.

[9] Alan D. Christiansen, Andrea D. Edwards, and Carlos A. Coello Coello. Automated design of part feeders using a genetic algorithm. In *International Conference on Robotics and Automation*. IEEE, April 1996.

[10] Michael Erdmann, Matthew T. Mason, and George Vaněček Jr. Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10:226–247, August 1993.

[11] Kenneth Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, August 1993.

[12] Allan Heydon and Greg Nelson. The Juno-2 constraint-based drawing editor. Research Report 131a, Digital Systems Research Center, December 1994. See http://www.research.digital.com/SRC/juno-2.

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

[14] 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.

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

[16] Brian Mirtich, Ken Goldberg, Yan Zhuang, John Craig, Rob Zanutta, Brian Carlisle, and John Canny. Estimating pose statistics for robotic part feeders. In *International Conference on Robotics and Automation*. IEEE, April 1996.

[17] Balas K. Natarajan. Some paradigms for the automated design of parts feeders. *International Journal of Robotics Research*, 8(6):98–109, December 1989. Also appeared in Proceedings of the 27th Annual Symposium on Foundations of Computer Science, 1986.