From Robotic Juggling to Robotic Parts Feeding

Srinivas Akella Wesley H. Huang Kevin M. Lynch Matthew T. Mason

The Robotics Institute
Carnegie Mellon University

Abstract

This paper describes our recent work on robotic manipulation using a robot with just one motor. The drift field provided by gravity can serve as an uncontrolled motor to allow a single-motor robot to throw objects to itself, a simple type of robotic juggling. Similarly, the drift field provided by a conveyor belt can serve as an uncontrolled motor to allow a single-motor robot to feed parts in an automated manufacturing context. This paper sketches the principles of very simple manipulators, and describes our recent theoretical and experimental results.

1 Introduction

How do we deploy motors to manipulate an object? There are two competing approaches, which can be described as follows:

- "Generalist"
  - At the end of the arm we place a hand that can rigidly grasp the object.
  - So that the hand and the attached object will have general motion, we build an arm with six joints. (Or three joints for a planar task.) We place a motor at each joint.
- "Minimalist"
  - We do not grasp the object at all, the end effector is not a hand, but rather a prod, a paddle, or a bumper.
  - We use the nonlinear coupling among task and arm freedoms to obtain full control of the object with as few joints and motors as possible.

The generalist approach gives simpler software. The relation of arm motion to object motion is very simple. The object's configuration is easily determined from joint position sensors, and desired object motions are transformed to robot motions by a simple kinematic mapping.

The minimalist approach gives simpler hardware. It requires fewer joints and motors, and it does not require a hand. The minimalist approach may in some cases also be more general. In industrial practice, robot hands are not at all general; a different hand may be designed for every different object. In research labs more general hands have been designed, but their complexity exceeds the complexity of the arms, an observation not contradicted by an examination of the human system.

It is interesting to note that the generalist approach, viewed from a broader perspective, is really minimalist. Manipulation tasks may involve many objects. The manipulator works with one object at a time, thus decomposing a task with $6n$ freedoms so that it can be accomplished with just 6 motors.

This paper pursues the minimalist approach. There are two possible motives for minimalism. One motive is to understand the basic principles of manipulation by exploring the extremes, and by exploring the impact of limited resources. Another motive is more economic, to obtain a solution with a lower cost. This paper addresses primarily the former goal, but is informed by the latter goal. In fact, factory automation often employs minimalist solutions.

This paper addresses the minimalist approach by describing our recent work on two different systems. The first system is a single-degree of freedom robot that uses the gravitational field to control three degrees of freedom of an object in the vertical plane. It throws the object to itself, a simple form of juggling. The second system is a single-degree of freedom robot that works in a horizontal plane, using a conveyor belt in place

of gravity. In a sense it too throws the object to itself, and might have applications as a parts feeder in an automated factory.

1.1 Previous work

This paper touches on a number of topics, so that an adequate discussion of our intellectual precedents seems impossible in the space available. The interested reader should refer to [3, 12, 21].

Much of this work was inspired by industrial parts feeders such as bowl feeders and the APOS system (Hitakawa [16]). Related research includes dynamic parts orienting on a vibrating plate (Böhringer *et al.* [7], Swanson *et al.* [31]). Other forms of nonprehensile manipulation are parts orienting by tray-tilting (Erdmann and Mason [13]), tumbling (Sawasaki *et al.* [29]), pivoting (Aiyama *et al.* [2]), tapping (Higuchi [15], Huang *et al.* [17]), two pin manipulation (Abell and Erdmann [1]), and two palm manipulation (Erdmann [12], Zumel and Erdmann [34]).

Dynamic underactuated manipulation is similar to the control of underactuated manipulators, except the unactuated freedoms are controlled through unilateral frictional contacts. Research on underactuated manipulators includes that of Oriolo and Nakamura [27] and Arai and Tachi [5]. In work related to ours, Arai and Khatib [4] demonstrated rolling of a cube on a paddle held by a PUMA. Their motion strategy was handcrafted with the assumption of infinite friction at the rolling contact. In this paper, we automatically find motion strategies which take into account the friction coefficient. Our approach to motion planning uses nonlinear optimization, which has also been used by Witkin and Kass [32], Yen and Nagurka [33], and Chen [9] to find motion strategies for fully-actuated systems. Gradient descent approaches to motion planning for underactuated systems have also been proposed by Divelbiss and Wen [10], Fernandes *et al.* [14], and Sussmann [30].

Minimalism in robotics has also been studied by Donald *et al.* [11], Canny and Goldberg [8], and Böhringer *et al.* [6]. Raibert [28] and McGeer [24] constructed simple, elegant machines that use dynamics for stable locomotion.

The ideas in this paper draw heavily on the control of underactuated nonholonomic systems. A good introduction to nonlinear control is given by Nijmeijer and van der Schaft [26], and nonholonomic robotic systems are discussed in the texts by Latombe [18] and Murray *et al.* [25].

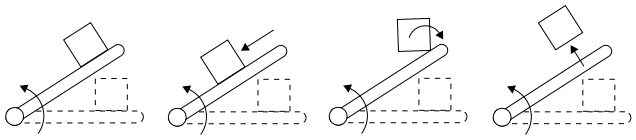

Figure 1: Manipulation phases: dynamic grasp, slip, roll, and free flight.

2 Robotic juggling

Our work on robotic juggling originated with an interest in *dynamic manipulation*: the exploitation of dynamics in manipulation tasks. Our first work [23] focused on the *dynamic grasp*: using the dynamic load of an accelerated object to hold it stably against the end effector of the robot. Since then we expanded our interests to include *rolling* and *free flight*. Our work, previously described in [22], includes analysis of the reachability using nonlinear differential control theory, automatic planning using numerical optimization techniques, and experiments using the NSK arm. The NSK arm is a single degree of freedom arm, working in the vertical plane, powered by a direct-drive NSK Megatorque motor. In this paper we give a brief outline of the planning method and experiments. This section is a condensed revision of [22].

We can define the following phases (Figure 1)

1. *Dynamic grasp.* An object is in a dynamic grasp if it makes line contact with he manipulator and the manipulator accelerates such that the object remains fixed against it.

2. *Slip.* Controlled slip provides control of two state variables, the slipping distance and the slipping velocity.

3. *Roll.* Rolling provides control of two state variables, the rolling angle and angular velocity.

4. *Slip and roll.* Slip and roll occur simultaneously, giving control of up to four state variables.

5. *Free flight.* After the object is released, it follows a one-dimensional path through its state space, parameterized by its time of flight.

We have explored the following dynamic tasks:

1. *Snatch:* transfer an object initially at rest on a table to rest on the manipulator. The manipulator accelerates into the object, transferring control of the object from the table to the manipulator.

Throw: throw the object to a desired goal state. The object is carried with a dynamic grasp and released instantaneously (no slipping or rolling) at a point where the free-flight dynamics will take the object to the goal state (possibly a catch).

Roll: roll a polygonal object sitting on the manipulator from one statically stable edge to another statically stable edge.

Rolling throw: allow the object to begin rolling before throwing it. By controlling the roll angle and velocity before the release, the dimension of the object's accessible state space is increased by two.

The manipulation phases in these tasks are dynamic grasp, roll, and free flight. Slipping contact is not used.

To solve these problems we cast trajectory planning as a constrained nonlinear optimization problem, where the system's initial state and goal state (or state manifold) are specified as constraints to the optimization. The trajectory is also subject to a set of nonlinear equality and inequality constraints arising from constraints on the manipulator motion and the dynamics governing the object's motion relative to the manipulator. Because dynamic nonprehensile manipulation relies on friction between the object and the manipulator, and friction coefficients are often uncertain and varying, the optimization is usually asked to minimize the required friction coefficient for successful manipulation. Unlike other work on optimizing the time or energy of a robot's motion, we are more concerned with making the manipulation maximally robust to variations in the friction coefficient.

Every task is assumed to consist of a sequence of manipulation phases made up of one or more of the following: a (dynamic) grasp phase g, a roll phase r, and a flight phase f. With this notation a throw (as defined above) is denoted gf, a roll is denoted grg, and a rolling throw is denoted grf. A snatch can be either g or rg. We assume that there is no rebound from the impact at the end of a roll (the transition from r to g). The instant the new edge contacts, if the dynamic grasp conditions are met, then the object is assumed to be in a dynamic grasp.

The times of the manipulation phases are t_{g1} for the first dynamic grasp phase g, t_{roll} for the rolling phase r, t_{g2} for the second g phase, and t_{flight} for the flight phase f. If a phase is omitted, its corresponding time duration is zero.

2.1 Design variables

The design variables consist of the variables $\mathbf{x}$ specifying the trajectory of the manipulator; the times of each applicable phase t_{g1}, t_{roll}, t_{g2}, and t_{flight}; and the friction coefficient μ between the object and the manipulator. These variables are not independent; the trajectory of the manipulator implicitly defines the time of each phase. However the problem formulation is much simpler if we make each of these variables explicit and constrain them to be dynamically consistent. Although we cannot control the friction coefficient μ, it is convenient to represent μ as a design variable and explicitly enfore the resulting friction constraints.

For an n-joint robot, $\mathbf{x} = (\mathbf{x}^1, \mathbf{x}^2, \ldots \mathbf{x}^n)$, where $\mathbf{x}^i$ is the vector of knot points for the cubic B-spline position history of joint i. The knot points are evenly spaced in time. The trajectory is C^2 and piecewise cubic, the velocity is C^1 and piecewise quadratic, and the acceleration is C^0 and piecewise linear.

2.2 Constraints

Constraints arise from limitations on the motion of the manipulator and constraints on the motion of the object.

- Manipulator constraints.

  1. Position constraints: joint limits and obstacles are transformed to inequality constraints on arm configuration.
  2. Joint velocity constraints: upper and lower bounds on joint velocity.
  3. Joint torque constraints: upper and lower bounds on joint torque can be transformed to inequality constraints on the design variables.
  4. Torque derivative constraints: upper and lower bounds on joint torque derivatives can be transformed to inequality constraints on the design variables.
  5. Initial state constraints are used to make sure the trajectory starts in the right place.

  Object constraints.

  1. Dynamic grasp constraints (dynamic grasp phase): the object acceleration is constrained so that the object stays

in edge to edge contact with the robot, using a contact force that satisfies Coulomb's law.

2. Rolling friction constraints (rolling phase only): the object acceleration is constrained so that rolling occurs without slip and without breaking contact, using a contact force that satisfies Coulomb's law.
3. Roll angle constraints (rolling phase only): upper and lower bounds on the roll angle prefent the object from penetrating the arm.
4. Roll completed constraints (for rolls only): require the roll to be completed at the appropriate time.
5. Release state constraints (for throws only): require the object to reach the goal submanifold at the appropriate time.

2.3 Objective function

In most of our problems, we minimize the required friction coefficient μ between the object and manipulator.

2.4 Nonlinear Optimization

Sequential quadratic programming is used to solve the nonlinear program. We used CFSQP (C code for Feasible Sequential Quadratic Programming [19]. As with all iterative gradient-based optimization techniques, sequential quadratic programming finds a local optimum which is not necessarily the global optimum. In addition, the finite-dimensional parameterization of the manipulator trajectory artificially limits the space of possible trajectories. The particular local optimum achieved depends on the shape of the feasible space and the initial guess.

2.5 Experiments

We used a one-degree-of-freedom arm powered by an NSK direct-drive motor. The arm link is a hollow aluminum beam 122 cm in length with a 10 cm square cross-section. It also has a palm mounted at a 45 degree angle. The top surface of the arm and the palm are used as manipulation surfaces. To increase friction and damping, these surfaces are covered with a soft foam. Trajectories specified by the planner are directly implemented, without modification. There are unmodeled effects, such as impact rebound and the soft foam, which may cause the plans to fail. In such cases the problem specification can be modified to compensate.

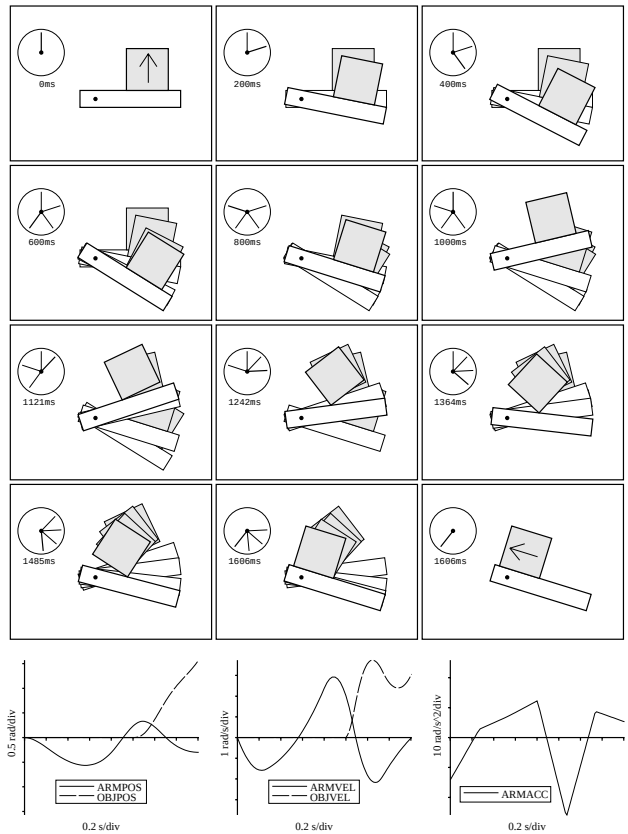

Figure 2: Roll: the trajectory found by the optimization.

To execute a throw, the arm is maximally decelerated at the release point. This causes the object to be released nearly instantaneously. If the arm is also to catch the object, it follows a bang-bang trajectory to reach the catching configuration.

Figures 2 and 3 show a roll. The object is a lightweight square frame constructed of wood. The nine-knot trajectory took 55 iterations and

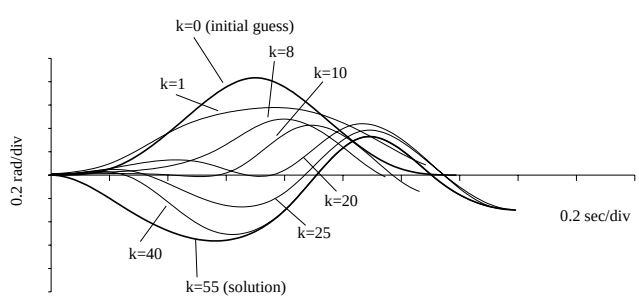

Figure 3: Roll: initial guess, solution, and intermediate iterates.

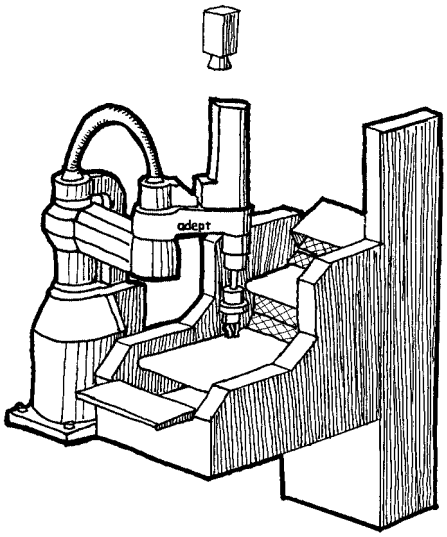

Figure 4: The Adept Flex Feeder System. A SCARA robot picks parts off the middle of three conveyors. These three conveyors, along with an elevator bucket, circulate parts; an overhead camera looks down on the back-lit middle conveyor to determine the position and orientation of parts.

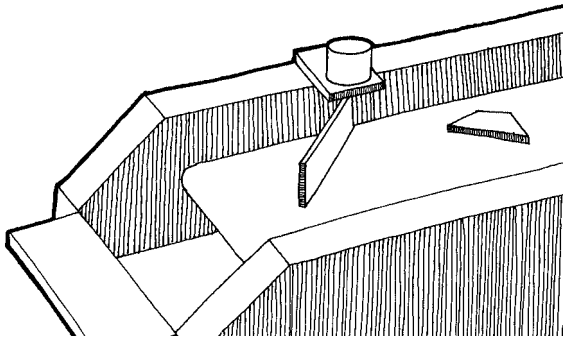

Figure 5: The Flex Feeder with a rotatable fence.

32 seconds on a Sun SPARC 20. The objective is to minimize the impact velocity at the end of the roll. (we also limited the end angle of the arm to make the roll experimentall robust to impact.) Note the windup before the roll.

We have also demonstrated snatches, throws, rolls, and rolling throws. Details are given in [21, 22].

3 Robotic parts feeding

This section describes an approach called "1JOC" (One Joint Over Conveyor, pronounced "one jock"), and is a condensed revision of [3]. Initially the approach was conceived as a variation on the Adept Flex Feeder (see Figure 4). The

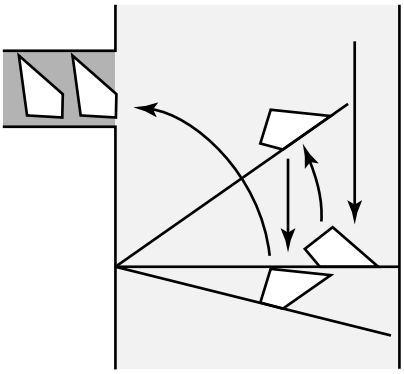

Figure 6: We can feed a part by alternately pushing it with the fence and letting it drift along the conveyor.

Flex Feeder uses a system of conveyors to recirdulate parts, presenting them random orientation to a camera and robotic manipulator. Those parts that are in a graspable configuration may then be picked up by the robot and assembled into a product, placed in a pallet, or otherwise processed.

The question is whether, at least in parts feeding applications, we could replace the four-degree-of-freedom robotic manipulator with a simpler single-degree-of-freedom device. Figures 5 and 6 show a possible variation using a fence driven by a single revolute joint. By a sequence of pushing operations, punctuated by drift along the conveyor, the fence positions and orients a part and directs it into the entry point of a feeder track which carries the part to the next station.

There are many variations on this idea: a 2JOC, multiple 1JOCs working in parallel, curved fences, and so on. However, as an initial study of the fundamental characteristics of the idea, we focus on the simplest version: a straight fence, collinear with the joint axis, working above a constant velocity conveyor.

There are many different measures of manipulation. For example, a system with *small-time local controllability* could move the object along an arbitrary trajectory. A system with *global controllability* could move the object from an arbitrary start to an arbitrary goal. The 1JOC posesses neither of these properties. However, the 1JOC can in principle function as a parts feeding device. We formalize a measure of manipulation called the *feeding property*:

> A system has the *feeding property* over a set of parts $\mathcal{P}$ and set of initial configurations $\mathcal{I}$ if, given any part in $\mathcal{P}$, there is some output configuration $\mathbf{q}$ such that the system can move the part to $\mathbf{q}$ from

any location in $\mathcal{I}$.

Our main results are:

- 1JOC has the feeding property over the set of all polygons.
- For any polygon, a planner can determine a suitable goal, and can construct a sequence of pushes and drifts from any initial point to the goal.

The feeding property is proven for an infinite length fence and infinitely wide conveyor. The planner takes bounds on fence length and conveyor width into account. We have also successfully demonstrated some plans in the laboratory. This paper sketches the development of both results.

We adopt some conventions to simplify the presentation. We assume an origin coincident with the fence pivot, and we assume the belt's motion is in the $-y$ direction. The fence angle is measured with respect to the x axis. If the conveyor moves "down", then the fence at zero degrees is "horizontal". We will only use the right half of the belt, i.e. the half plane $x > 0$. The object's position along the fence is characterized by a *contact radius* r.

3.1 1JOC Primitives

We use four primitive actions, illustrated in Figure 7. A stable push means that the object stays in contact with the fence. A turn starts with a stable push to a fence angle of θ^+, followed by drift, until the object is caught at a fence angle of θ^-, followed by another stable push back to a fence angle of 0. A jog is a sequence of small stable pushes interleaved with drift. In this way it is possible to move the object inward or outward along the fence, decreasing or increasing the contact radius r without changing the angle of the object. The last primitive is a convergent turn, which is only required when an object must be turned through 180 degrees.

The 1JOC manipulates objects as follows. First the fence is held at zero degrees until the object comes to rest on the fence. The object is rotated to a desired edge by a sequence of turns (sometimes including a convergent turn) and jogs. A final jog brings the object to the desired contact radius. That this scenario always works, and just how to choose the parameters, are shown below.

3.2 Feedability

Here we sketch the proof of the feedability property.

- A stable edge is defined is defined to be an edge that will remain in stable contact with the fence, with the fence at zero degrees plus or minus some small angle.
- Every polygon has at least one stable edge. This follows from the observation that the center of gravity is in the interior of the polygon.
- A stable push is possible for any stable edge, provided that the contact radius is higher than some minimum. This minimum contact radius depends on the edge, on the object shape, the center of gravity, the angular rate of the fence relative to the conveyor speed, and the coefficient of friction. Any stable push may be preceded by jogs to move the object out to the minimum contact radius.
- Every polygon has at most one stable edge from which it is impossible to rotate the polygon counter-clockwise to another stable edge. For most polygons this can be accomplished with turns. There is one kind of polygon, with exactly two stable edges, parallel to each other, which requires a convergent turn.
- Given a polygon, we choose the goal configuration as follows. If the there is a stable edge from which it is impossible to rotate counter-clockwise to the next stable edge, then that edge is chosen as the goal edge. Otherwise any stable edge will serve. The goal contact radius is arbitrary.

The feeding property follows in a straightforward way. Given an arrival configuration, the fence can catch the polygon on a stable edge, use jogs as necessary to move the part far enough out on the fence, use turns, and possibly a convergent turn, to rotate to the goal edge, then use jogs again to move the object to the goal radius.

3.3 Planning

Each 1JOC plan is a sequence of turns (possibly including a convergent turn) interleaved with jogs, rotating from the initial edge through a sequence of stable edges to the goal edge. We consider every sequence of stable edges from initial to goal edge that do not rotate more than 360 degrees. For each rotation there is a turn, and there

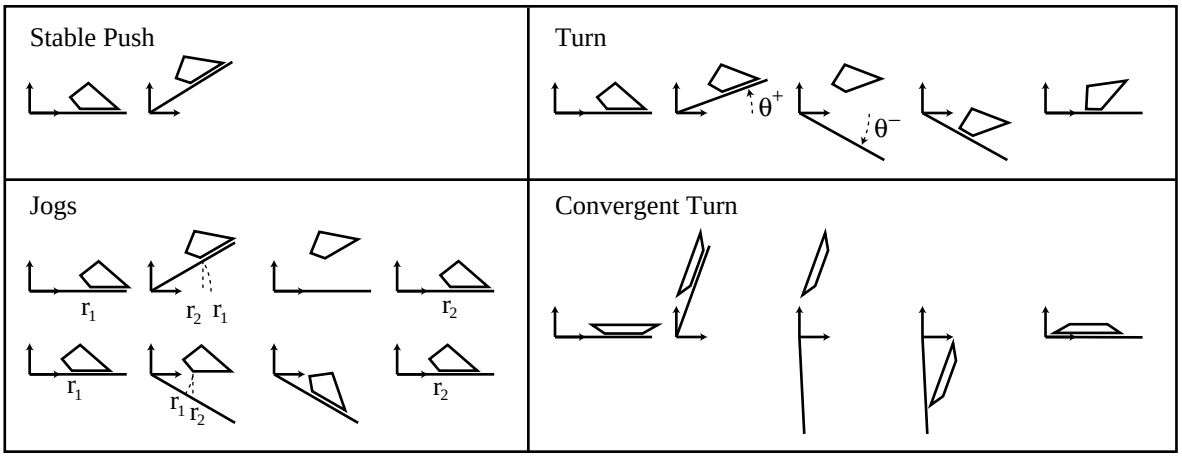

Figure 7: Step by step illustrations of the 1JOC primitives.

may also be a jog. Each turn and each jog has parameters. Given a fence and conveyor of known dimensions, given the shape and center of mass of the polygon, and given the coefficient of friction, we can choose the turn and jog parameters to minimize the time required.

We obtain values for these parameters using a nonlinear programming package GINO [20]. Constraints are formulated as follows:

- The contact radius must always exceed the minimum for any turn.
- The fence rotation must stay in valid ranges.
- The part must stay on the belt.
- The contact radius must not exceed the fence length.
- The fence does not contact the part during drift.
- The start and end configurations must match the given start and goal.

We use an approximation for the time required for the jogs, and find a plan minimizing the time. Details of the jogs are planned afterwards. A feasible solution to the above problem always exists, unless the belt is so narrow or the fence so short that the minimum contact radius cannot be satisfied for stable pushes. When this condition is satisfied, a plan with a maximum of three rotations exists, although it may not be the fastest plan.

4 Conclusion

This paper has described two different systems addressing two different manipulation tasks employing similar principles. In one case the task is governed by the dynamics of Newton, so that the state of the system includes position and rate variables. In the other case the task is governed by quasistatic mechanics, so that the state of the system is just position. The more interesting commonalities are the relation of the available controls and the drift fields, and the coordination of these fields to obtain effective manipulation.

Acknowledgments

Many thanks to Costa Nikou for the implementation of the 1JOC on the Adept robot. We thank Mike Erdmann, Garth Zeglin, and Nina Zumel for early discussions about this paper. We thank Adept, NSK, and SONY for helping us learn about automation and for making equipment available. We thank André Tits and Craig Lawrence for providing the CFSQP software. This work was supported by ARPA and NSF under grant IRI-9318496.

References

[1] T. Abell and M. A. Erdmann. Stably supported rotations of a planar polygon with two frictionless contacts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995.

[2] Y. Aiyama, M. Inaba, and H. Inoue. Pivoting: A new method of graspless manipulation of object by robot fingers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 136–143, Yokohama, Japan, 1993.

[3] S. Akella, W. Huang, K. M. Lynch, and M. T. Mason. Planar manipulation on a con-

veyor with a one joint robot. In *International Symposium on Robotics Research*, 1995.

[4] H. Arai and O. Khatib. Experiments with dynamic skills. In *1994 Japan–USA Symposium on Flexible Automation*, pages 81–84, 1994.

[5] H. Arai and S. Tachi. Position control system of a two degree of freedom manipulator with a passive joint. *IEEE Transactions on Industrial Electronics*, 38(1):15–20, Feb. 1991.

[6] K. Böhringer, R. Brown, B. Donald, J. Jennings, and D. Rus. Distributed robotic manipulation: Experiments in minimalism. In *International Symposium on Experimental Robotics*, 1995.

[7] K. F. Böhringer, V. Bhatt, and K. Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *IEEE International Conference on Robotics and Automation*, 1995.

[8] J. F. Canny and K. Y. Goldberg. "RISC" industrial robotics: Recent results and open problems. In *IEEE International Conference on Robotics and Automation*, pages 1951–1958, 1994.

[9] Y.-C. Chen. Solving robot trajectory planning problems with uniform cubic B-splines. *Optimal Control Applications and Methods*, 12:247–262, 1991.

[10] A. W. Divelbiss and J. Wen. Nonholonomic path planning with inequality constraints. In *IEEE International Conference on Decision and Control*, pages 2712–2717, 1993.

[11] B. R. Donald, J. Jennings, and D. Rus. Information invariants for cooperating autonomous mobile robots. In *International Symposium on Robotics Research*, Hidden Valley, PA, 1993. Cambridge, Mass: MIT Press.

[12] M. A. Erdmann. An exploration of nonprehensile two-palm manipulation: Planning and execution. In *International Symposium on Robotics Research*, 1995.

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

[14] C. Fernandes, L. Gurvits, and Z. Li. Near-optimal nonholonomic motion planning for a system of coupled rigid bodies. *IEEE Transactions on Automatic Control*, 30(3):450–463, Mar. 1994.

[15] T. Higuchi. Application of electromagnetic impulsive force to precise positioning tools in robot systems. In *International Symposium on Robotics Research*, pages 281–285. Cambridge, Mass: MIT Press, 1985.

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

[17] W. Huang, E. P. Krotkov, and M. T. Mason. Impulsive manipulation. In *IEEE International Conference on Robotics and Automation*, 1995.

[18] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[19] C. Lawrence, J. L. Zhou, and A. L. Tits. User's guide for CFSQP version 2.3. Institute for Systems Research 94-16, University of Maryland, 1994.

[20] J. Liebman, L. Lasdon, L. Schrage, and A. Waren. *Modeling and Optimization with GINO*. The Scientific Press, 1986.

[21] K. M. Lynch. *Nonprehensile Robotic Manipulation: Controllability and Planning*. PhD thesis, Carnegie Mellon University, Mar. 1996. CMU-RI-TR-96-05.

[22] K. M. Lynch and M. T. Mason. Dynamic underactuated nonprehensile manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996. To appear.

[23] M. T. Mason and K. M. Lynch. Dynamic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 152–159, Yokohama, Japan, 1993.

[24] T. McGeer. Passive dynamic walking. *International Journal of Robotics Research*, 9(2):62–82, 1990.

[25] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[26] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.

[27] G. Oriolo and Y. Nakamura. Control of mechanical systems with second-order nonholonomic constraints: Underactuated manipulators. In *Conference on Decision and Control*, pages 2398–2403, 1991.

[28] M. H. Raibert. *Legged Robots That Balance*. Cambridge: MIT Press, 1986.

[29] N. Sawasaki, M. Inaba, and H. Inoue. Tumbling objects using a multi-fingered robot. In *Proceedings of the 20th International Symposium on Industrial Robots and Robot Exhibition*, pages 609–616, Tokyo, Japan, 1989.

[30] H. Sussmann. A continuation method for nonholonomic path-finding problems. In *IEEE International Conference on Decision and Control*, pages 2718–2723, 1993.

[31] P. J. Swanson, R. R. Burridge, and D. E. Koditschek. Global asymptotic stability of a passive juggler: A parts feeding strategy. In *IEEE International Conference on Robotics and Automation*, pages 1983–1988, 1995.

[32] A. Witkin and M. Kass. Spacetime constraints. *Computer Graphics*, 22(4):159–168, 1988.

[33] V. Yen and M. L. Nagurka. A suboptimal trajectory planning problem for robotic manipulators. *ISA Transactions*, 27(1):51–59, 1988.

[34] N. B. Zumel and M. A. Erdmann. Balancing of a planar bouncing object. In *IEEE International Conference on Robotics and Automation*, pages 2949–2954, San Diego, CA, 1994.