1998 IEEE International Conference on Robotics and Automat ion

The Roles of Shape and Motion in Dynamic Manipulation: The Butterfly Example

Kevin M. Lynch* Naoji Shiroma† Hirohiko Arai‡ Kazuo Tanie‡

*Mechanical Engineering Department
Northwestern University
Evanston, IL 60208 USA

†Institute of Engineering Mechanics
University of Tsukuba
1-1 Tennodai, Tsukuba, 305 Japan

‡Biorobotics Division
Mechanical Engineering Laboratory
Namiki 1-2, Tsukuba, 305 Japan

Abstract

We are studying a juggler's skill called the "butterfly." Starting with a ball resting on the palm of his/her open hand, a skilled juggler can accelerate and shape his/her hand so that the ball rolls up the fingers, over the top, and back down to the back of the hand. This paper describes a robotic implementation of the butterfly. The hand's shape and motion combine to effect the rolling motion of the ball, and we find that the shape and motion parameters enter the dynamic equations in a similar way. We define parameterized spaces of hand shapes and motions, and using a simulation based on the rolling equations, we identify shape and motion solutions that roll the ball from one side of the hand to the other. We describe an implementation of the butterfly on our planar dynamic manipulation testbed FLATLAND. *This example is our first step toward exploring the roles of shape and motion in dynamic manipulation.*

1 Introduction

The robot manipulation problem is to find a set of controls that map the current state of the world to a goal state, where this mapping $\mathcal{M}$ is governed by the laws of physics. Typically the robot controls are specified as motions or forces and the system is engineered to simplify the mapping $\mathcal{M}$, making it easier to construct robot plans. For instance, a grasped object tracks the motion of the hand, simplifying $\mathcal{M}$ to an identity relationship between hand and object motions.

For manipulation by pushing, throwing, tapping, rolling, and catching, however, the mapping $\mathcal{M}$ is a function of manipulator shape, compliance, friction, restitution, etc. If we treat these as design (control) variables, then it is possible to exploit the information embedded in $\mathcal{M}$ to simplify robot hardware. Instead of constructing powerful dexterous robots to force $\mathcal{M}$ to be a simple function, we can construct simple robots which work with, not against, the natural environmental dynamics.

In this paper we consider the relative roles of manipulator shape and motion in dynamic nonprehensile manipulation. By *nonprehensile* we mean that the manipulated object is not grasped. By *dynamic* we mean that the robot exploits dynamics to help control the motion of the part. An example of dynamic nonprehensile manipulation is shooting a basketball. As the hand moves, dynamic forces cause the ball to roll up the hand and off the fingertips into a free-flight trajectory toward the hoop. With dynamic nonprehensile manipulation, a robot can cause motion of the part relative to the end-effector, thereby controlling more part degrees-of-freedom (Lynch and Mason [9]).

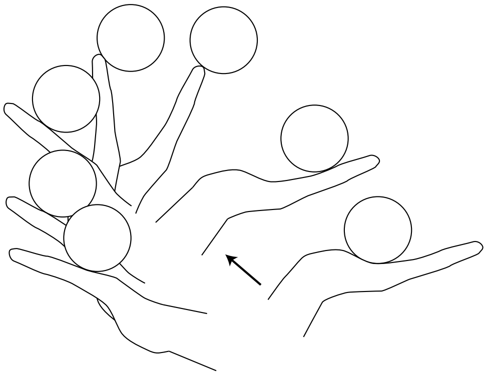

Figure 1: Sketch of a juggler doing a "butterfly."

This paper focuses on rolling manipulation, where an object rolls freely on the surface of the manipulator. We find that the derivative of the curvature of the manipulator surface and the acceleration of the manipulator enter into the dynamic equations of rolling in a similar way. This implies that, at least locally, we can trade freedom in the manipulator trajectory for freedom in the manipulator surface shape. To obtain a desired rolling motion of the object, we may be able to reduce the number of actuators required by properly designing the shape of the surface. This is a type of *dynamic cam*—we can use rolling dynamics and freedom in designing the shape of the manipulation surface to transform simple rotational or translational actuator motions to the desired motion of the object.

2 The Butterfly

We have begun our investigation into the roles of shape and motion in dynamic manipulation by examining a juggler's skill called the "butterfly" (Figure 1). Starting with a ball resting on the palm of his/her open hand, a skilled juggler can accelerate and shape his/her hand so the ball rolls up the fingers, over the top, and down to the back of the hand.

Our goal was to perform a planar version of this skill with a one joint robot, as shown in Figure 2. The design problem is to find a hand shape and 1 DOF revolute motion profile that rolls the disk from one side of the hand to the other while maintaining rolling contact at all times.

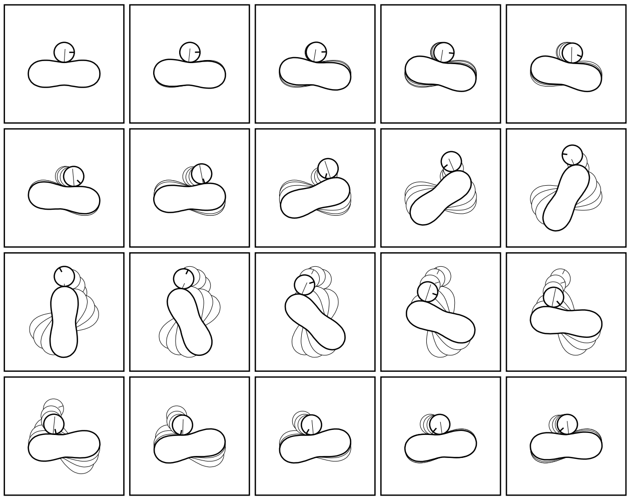

Figure 2: A simulated robotic butterfly. Gravity acts downward. The contact force is shown.

3 Dynamic Rolling Equations

In this section we present the dynamic equations of rolling between a moving surface (the "hand," which could be any manipulator surface) and an object rolling with point contact on the surface. We assume a hard contact model—no torque can be applied about the contact normal, and spin about the contact normal is not explicitly prohibited. Only slip is disallowed.

The kinematic equations of rolling have been derived previously in the context of rolling within a grasp. First-order analysis relating the relative velocity of two objects to the change in contact coordinates has been carried out by Kerr and Roth [7], Montana [10], and Cole *et al.* [3]. Sarkar *et al.* [11] built on Montana's work by deriving the second-order relationship between the relative motion of the contacting bodies and the acceleration of the contact coordinates. This information is used in the dynamic control of rolling motion in a grasp. Cai and Roth [2] derived the equations of motion in a manner allowing higher-order analysis.

Our derivation builds on this work, but differs in two ways. 1) The motion of the object relative to the hand is not specified, but rather is determined by the acceleration (and resultant contact forces) of the hand. The input to the equations of motion is the acceleration of the hand, and the outputs are the acceleration of the object and the acceleration of the contact coordinates. This is a second-order dynamic analysis. 2) Our development applies to both the planar and the spatial case. For this reason, we adopt notation similar to that of Cole *et al.* [3] (see Figure 3). In the spatial case, simple transformations provide the metric tensor, curvature tensor, and torsion form used by Montana [10].

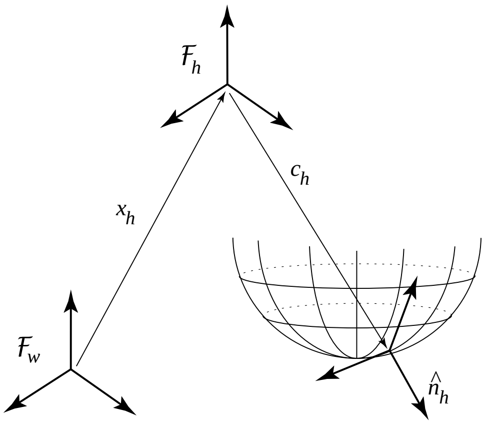

Figure 3: Coordinate conventions for the hand.

The hand is a one-dimensional curve (planar case) or a two-dimensional surface (spatial case). The hand is locally parameterized by s_h, where $s_h \in \mathbf{R}$ in the planar case and $s_h \in \mathbf{R}^2$ in the spatial case. In a coordinate frame $\mathcal{F}_h$ attached to the hand, the contact position is given by $c_h(s_h)$, where $c_h \in \mathbf{R}^2$ in the planar case and $c_h \in \mathbf{R}^3$ in the spatial case. We define the 2x1 (or 3x2) Jacobian

$$J_h = \frac{\partial c_h}{\partial s_h}.$$

The outward-pointing unit contact normal at s_h is given by $\hat{n}_h(s_h)$. In the planar case, $\hat{n}_h$ is simply $J_h/||J_h||$ rotated by 90 degrees so that it is the outward-pointing normal. (Alternatively, considering $J_h/||J_h||$ as a 3-vector with zero third component, $\hat{n}_h$ is this vector crossed with a unit vector out of the plane.) In the spatial case, $\hat{n}_h$ is the normalized cross-product of the two columns of J_h. The 2x1 (or 3x2) Jacobian of $\hat{n}_h$ defines the curvature of the surface and is written

$$K_h = \frac{\partial \hat{n}_h}{\partial s_h}.$$

The contact point in a world frame $\mathcal{F}_w$ is given by

$$x_h + R_h c_h(s_h),$$

where x_h is the location of $\mathcal{F}_h$ in $\mathcal{F}_w$ and R_h is a rotation matrix ($x_h \in \mathbf{R}^n$, $R_h \in SO(n)$ where $n = 2$ in the planar case and $n = 3$ in the spatial case). The time derivatives of x_h are written $\dot{x}_h = v_h$ and $\ddot{x}_h = a_h$. The angular velocity and angular acceleration of $\mathcal{F}_h$ in $\mathcal{F}_w$ are written ω_h and α_h, respectively. In the planar case, we have

$$\dot{R}_h = (\omega_h \times) R_h = \begin{pmatrix} 0 & -\omega_h \\ \omega_h & 0 \end{pmatrix} R_h,$$

and in the spatial case, we have

$$\dot{R}_h = (\omega_h \times) R_h = \begin{pmatrix} 0 & -\omega_{h3} & \omega_{h2} \\ \omega_{h3} & 0 & -\omega_{h1} \\ -\omega_{h2} & \omega_{h1} & 0 \end{pmatrix} R_h.$$

The matrix $(\alpha_h \times)$ may be defined similarly to $(\omega_h \times)$.

We can make similar definitions for the object being manipulated, replacing the subscript "h" (for hand) with the subscript "o" (for object). The object frame $\mathcal{F}_o$ is fixed to the center of mass of the object. The mass of the object is m and its inertia matrix is I_o expressed in $\mathcal{F}_o$.

We are now ready to derive the rolling equations.

1) Contact position constraint. The contact points on the object and the hand must be coincident:

$$x_h + R_h c_h - (x_o + R_o c_o) = 0.$$

This constraint differentiates twice to yield the following two (three) linear equations in the planar (spatial) case.

$$\begin{aligned} a_h + (\alpha_h \times + \omega_h \times \omega_h \times) R_h c_h + 2\omega_h \times R_h J_h \dot{s}_h \\ + R_h (J_h \ddot{s}_h + \dot{J}_h \dot{s}_h) - (a_o + (\alpha_o \times + \omega_o \times \omega_o \times) R_o c_o \\ + 2\omega_o \times R_o J_o \dot{s}_o + R_o (J_o \ddot{s}_o + \dot{J}_o \dot{s}_o)) = 0. \end{aligned} \qquad (1)$$

2) Contact normal constraint. The unit contact normals must be opposite:

$$R_o \hat{n}_o + R_h \hat{n}_h = 0.$$

This constraint differentiates twice to yield the following two (three) linear equations in the planar (spatial) case. One of these equations is redundant because $\hat{n}_o$ and $\hat{n}_h$ are constrained to be unit.

$$\begin{aligned} (\alpha_o \times + \omega_o \times \omega_o \times) R_o \hat{n}_o + 2\omega_o \times R_o K_o \dot{s}_o \\ + R_o (K_o \ddot{s}_o + \dot{K}_o \dot{s}_o) + (\alpha_h \times + \omega_h \times \omega_h \times) R_h \hat{n}_h \\ + 2\omega_h \times R_h K_h \dot{s}_h + R_h (K_h \ddot{s}_h + \dot{K}_h \dot{s}_h) = 0. \end{aligned} \qquad (2)$$

3) Rolling constraint. To maintain rolling, the acceleration of the contact points on the object and the hand must be equal when projected to the contact tangent space. This yields one (two) linear equations in the planar (spatial) case.

$$\begin{aligned} R_h J_h^T (a_h + (\alpha_h \times + \omega_h \times \omega_h \times) R_h c_h \\ - a_o - (\alpha_o \times + \omega_o \times \omega_o \times) R_o c_o) = 0. \end{aligned} \qquad (3)$$

4) Force constraint. The contact force passes through the contact point. This yields one (three) linear equations in the planar (spatial) case:

$$\tau - r \times f = 0$$

or equivalently,

$$I_w \alpha_o + \omega_o \times I_w \omega_o - R_o c_o \times m a_o = 0. \qquad (4)$$

$I_w = R_o I_o R_o^T$ is the object inertia matrix expressed in the world frame $\mathcal{F}_w$. The $\omega_o \times I_w \omega_o$ term vanishes in the planar case.

We would like to solve Equations (1)–(4) for a_o, α_o, $\ddot{s}_o$, and $\ddot{s}_h$ as a function of the state of the system and the inputs a_h and α_h. (We assume the input is the acceleration of the hand, not force/torque.) Rearranging the equations, we get the form

$$\mathbf{Ax} = \mathbf{b},$$

where **A** is 6x5 (11x10), **x** (the variables to be solved for) is a 5-vector (10-vector), and **b** is a 6-vector (11-vector) in the planar (spatial) case. These equations may be solved by premultiplying each side by $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$, the pseudo-inverse of **A**, allowing us to simulate dynamic nonprehensile rolling manipulation of smooth objects. In practice we also add a "snapping" routine to make sure the object stays on the surface of the hand. We have written a rolling simulator in C. The simulation enforces rolling contact; we check the contact force implied by a_o and α_o to see if contact and friction constraints are satisfied.

The curvature derivative shape information in the term $\dot{K}_h = \frac{\partial K_h}{\partial s_h} \dot{s}_h$ (Equation 2) gives us a way to design the shape of the hand to affect the rolling motion in a way similar to the acceleration of the hand. This curvature derivative information integrates to yield the shape of the hand.

4 Designing the Shape and Motion

Using the rolling simulation, we would like to design a hand shape and motion to solve the planar butterfly task. Below we describe a space of shapes and motions from which we will find a solution.

4.1 Shape Space

We would like a symmetric hand with a stable well (local minimum in a gravity field) when the disk is at the beginning or end of the roll, as in Figure 2. We chose simple polynomial functions satisfying these conditions:

$$x(s_h) = a + c s_h^2 + e s_h^4$$

$$y(s_h) = b s_h + d s_h^3 + f s_h^5,$$

where $s_h \in [-1, 1]$ describes the right half of the hand. (The other half is the mirror image.) Note that $x(s_h)$ and $y(s_h)$ are even and odd functions, respectively. The six coefficients must satisfy the two independent equations $x(-1) = x(1) = 0$ and $y'(-1) = y'(1) = 0$, where the constraint on $y'(\pm 1)$ forces the hand to be flat at these points. We chose $a = 18$, $b = -18$, leaving a two-dimensional shape space which can be parameterized by c and d (Figure 4). Essentially d controls the depth of the well and c controls whether or not the curvature is pushed out to the ends of the hand. For our experiments, we chose $a = 18.0\text{cm}$, $b = -18.0\text{cm}$, $c = -25.2\text{cm}$, $d = 21.6\text{cm}$, $e = 7.2\text{cm}$, and $f = -9.36\text{cm}$.

Two important points are worth noting. First, not all points in the shape space correspond to appropriate hands. Some choices of c and d yield self-intersecting curves, curves without wells, or wells with curvature too high to accommodate the disk (radius of 5.1cm in the experiments). Second, the curvature is discontinuous at $s_h = \pm 1$, because we have required only that $\hat{n}_h$ be continuous there. The rolling equations are ill-defined at $s_h = \pm 1$. This problem can be solved by increasing the order of the shape polynomials and forcing them to satisfy continuous curvature and curvature derivative constraints at $s_h = \pm 1$. Here we use the low order polynomials and simply take care in the simulation near these singularities.

4.2 Motion Space

We would like to find a symmetric hand motion that dips the hand to begin the disk rolling, rotates 180 degrees to roll

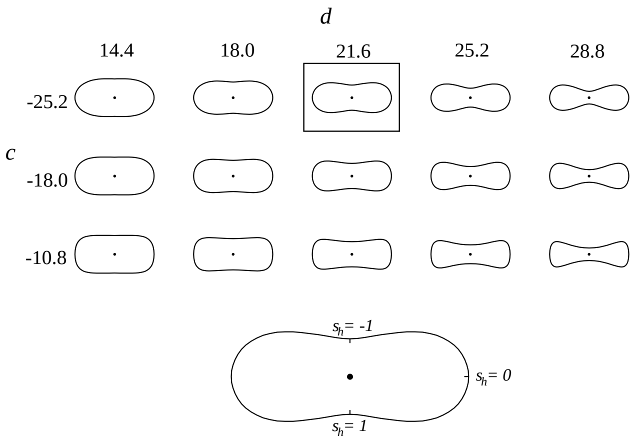

Figure 4: Some points in the two-dimensional hand shape space for $a = 18$, $b = -18$, and the shape we chose for our experiments.

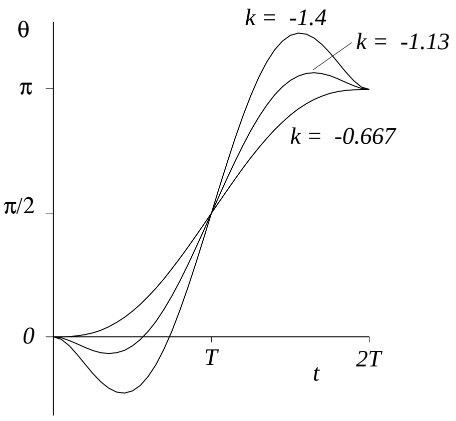

Figure 5: Example motion profiles for different values of k.

the disk to the other side of the hand, overshoots to stop the rolling, and finally settles in a horizontal position. We chose the following motion profile:

$$\theta(t) = \frac{\pi(j(t-T) + k(t-T)^3 + l(t-T)^5)}{2(jT + kT^3 + lT^5)} + \frac{\pi}{2}, t \in [0, 2T]$$

where $2T$ is the total time of the motion. This function rotates the hand from 0 to π. We require $\dot{\theta}(0) = \dot{\theta}(2T) = 0$ and we set $j = 1$. The remaining two-dimensional motion space can be parameterized by k and T, where k determines how far the hand initially dips to begin rolling of the disk. $k = -2/3$ yields zero dip, and the dip increases with increasingly negative values of k. Example motion profiles are shown in Figure 5.

4.3 Simulation

Using the simulation with the hand shape chosen in Section 4.1 and a disk of uniform mass and a radius of $5.1\mathrm{cm}$, we found a one-dimensional locus of solutions to the butterfly problem in the two-dimensional $k - 2T$ motion space. (For a given motion, we could similarly find a one-dimensional locus of solutions in the two-dimensional $c - d$ shape space.)

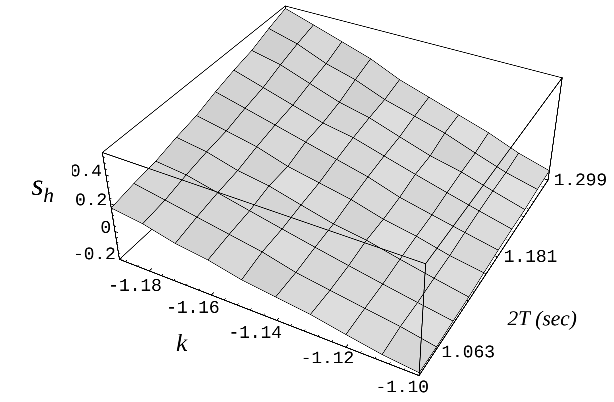

Figure 6: Surface plot of the contact parameter s_h at time T during the butterfly as a function of k and $2T$. The locus of $k - 2T$ solutions to the butterfly corresponds to the points where $s_h = 0$.

A "solution" is any motion that takes the disk to exactly zero velocity on the opposite side of the hand. Because the hand shape and the motion are symmetric, the motion of the disk on the hand is symmetric for any solution.

In a neighborhood of a solution in the $k - 2T$ space, increasing the initial dip (smaller k) or the motion time $2T$ causes the disk to roll further (past the goal position). Figure 6 shows the hand contact parameter s_h at time T during the butterfly as a function of k and $2T$. Because of the symmetry of solutions, a necessary condition for a solution motion is that $s_h = 0$ at time T. The smoothness and monotonicity of the s_h surface of Figure 6 allow us to quickly converge on the locus of points satisfying this condition.

In full gravity ($9.8m/s^2$), the fastest solution which maintains contact at all times is $2T = 1.095s$, with $k = -1.1493$. At this speed, the contact force becomes zero when the disk reaches its apex. At any higher speed the hand will throw the disk. This is the time-optimal rolling solution in the $k - 2T$ motion space.

Figure 2 shows the solution for $2T = 1.181s$ and $k = -1.1393$. In this example, the contact force at the apex of the roll is 17.5% of the gravitational force. This motion requires a contact friction coefficient of 0.306 to maintain rolling contact. As we increase $2T$, the value of k that solves the butterfly problem increases, implying a smaller initial dip (Figure 7). The friction coefficient required to maintain rolling goes to zero as $2T$ goes to infinity. This is because there is a locus of (θ, s_h) equilibrium configurations connecting the start configuration $(0, -1)$ and the goal configuration $(\pi, 1)$ where the contact force is normal to the surface and through the disk center of mass. Some of these configurations (in the wells) are stable, others are unstable.

5 Experimental Testbed: FLATLAND

We implemented the butterfly on FLATLAND (Figure 8). We built FLATLAND to be a general testbed for experiments in planar dynamic manipulation. FLATLAND consists of a tiltable air table, allowing us to perform planar dynamic manipulation experiments with variable gravity; a set of ma-

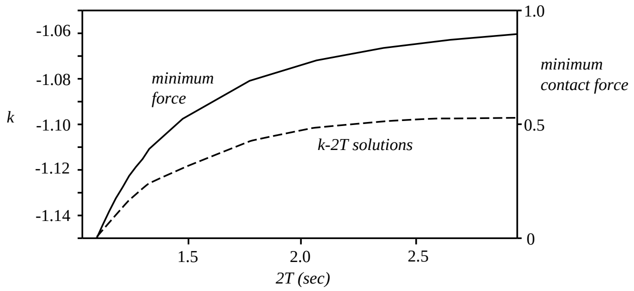

Figure 7: Plot of $k - 2T$ solutions to the butterfly problem and the corresponding minimum contact force as a ratio to the gravitational force.

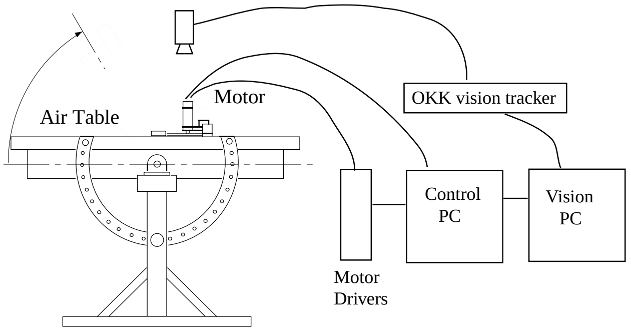

Figure 8: The FLATLAND experimental setup.

nipulator modules that can be configured as 1 or 2 DOF robots mounted on rails around or over the air table; and a 30 Hz OKK vision tracking system to track the motion of objects on the table. To the underside of each manipulator link is attached a *manipulation surface*. The manipulation surfaces are the parts of the robot actually making contact with the laminar objects floating on the table, and they can be changed easily to allow us to experiment with new manipulator shapes (Figure 9). The manipulators are controlled by a Pentium PC which receives vision data from an NEC PC98 which processes the data from the OKK vision tracker.

The hand was cut from aluminum by a CNC machine and attached to a manipulator module configured as a 1 DOF robot. The disk is made of plastic and is encircled by rubber O-rings to increase friction with the aluminum manipulation surface. We set the air table to a 5 degree angle, introducing a time-scaling $\kappa = \sqrt{1/\sin 5°} = 3.387$ from the full-gravity rolling solutions. (Increasing the time-scale permits more vision data during the roll.) The hand was made to follow the trajectory $2T = \kappa(1.181s) = 4.0s$, $k = -1.1393$, as shown in the simulation in Figure 2. Even without feedback, the butterfly often successfully rolled the disk to the goal well. By simply scaling $2T$, the butterfly was performed at different table angles.

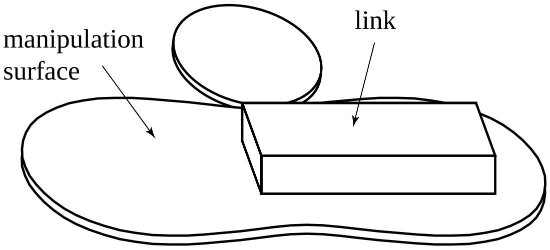

Figure 9: Manipulation surfaces attached to the robot links actually make contact with objects on the air table.

The open-loop butterfly is not robust. Often the disk rolls too far, overshooting the final position, or does not roll far enough, never reaching the goal configuration. In both cases the hand drops the disk. To make execution robust, we implemented vision feedback control. A simple estimator was used to estimate the disk's position between vision data frames, and this data was used in a 1 kHz control loop. The control follows three stages:

1. Perform the initial dipping motion open-loop to get the disk rolling along the hand.

2. Once the disk has passed a certain point (typically $s_h > -0.9$), simply servo the hand toward the angle and angular velocity in the planned trajectory that corresponds to the disk's current position on the hand. In other words, the contact parameter s_h drives the hand's motion. During this stage s_h is monotonically increasing, so there is a one-to-one mapping between s_h and the planned manipulator angle and angular velocity. The control law is written

$$\ddot{\theta} = k_p(\theta_p(s_{h,a}) - \theta_a) + k_d(\dot{\theta}_p(s_{h,a}) - \dot{\theta}_a),$$

   where $k_p, k_d > 0$, $s_{h,a}$ is the actual contact parameter (from vision feedback), θ_a is the actual hand angle, and $\theta_p(s_{h,a})$ is the planned hand angle when the disk is at $s_{h,a}$. (In our control system, the commanded acceleration $\ddot{\theta}$ is used to calculate a new reference position θ and velocity $\dot{\theta}$, which are then used in a PD controller to calculate joint torque. This approach suppresses nonlinearities in the actuator due to friction, etc.)

3. Once the disk has passed a certain point (typically $s_h > 0.9$), perform the final overshoot motion open-loop.

Despite the simplicity of the controller, it significantly stabilizes the rolling motion to small errors in initial conditions. Figure 10 shows experimental data for a butterfly under feedback control with $2T = 4.0s$, $k = -1.1393$. The θ and s_h trajectories are symmetric, but the execution time is extended to approximately $4.23s$. The controller slowed the motion of the hand in stage 2 to allow the disk more rolling time to compensate for errors.

If the rolling velocity after stage 1 is too large, the hand may throw the disk slightly when the disk reaches the top of the hand. (The butterfly often succeeds despite this, as the hand catches the disk and continues on the trajectory.) A slower nominal motion, which maintains a higher nominal contact force, could alleviate this problem. Another failure mode is when the disk does not have enough velocity from the initial dip to reach the top of the hand. We are currently investigating nonlinear feedback controllers based on the rolling dynamics to make the butterfly more robust.

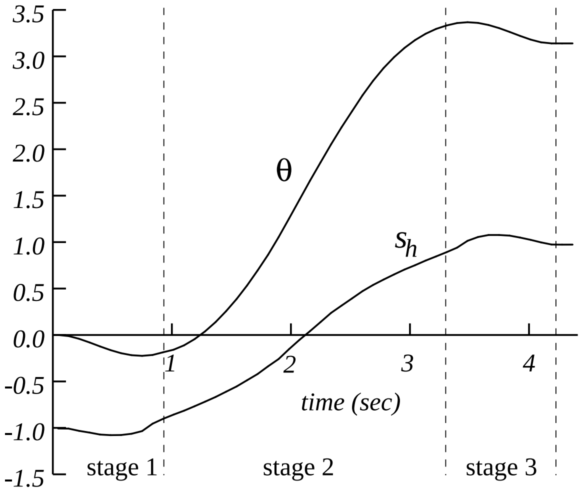

Figure 10: Experimental data for a butterfly with $2T = 4.0s$, $k = -1.1393$, and the table set at a 5° angle. The controller slows the execution of the hand trajectory during stage 2 to allow the disk more rolling time to compensate for errors. The total execution time is approximately $4.23s$.

6 Discussion

The butterfly system resembles a revolute two-joint robot with an unactuated second joint (Arai and Tachi [1]; Suzuki *et al.* [12]). Both systems are subject to a second-order nonholonomic constraint from a passive pivot joint. Three important differences are: 1) The pivot point is rolling along the surface of the manipulator with the butterfly. 2) The butterfly is performed in a gravity field. This allows equilibrium configurations to be stabilized, but trajectories cannot be time-scaled. The time scale is chosen by gravity. 3) There is a limited friction cone of contact forces that can be applied into the disk. Contact force constraints make the control problem particularly challenging.

We have broken the control of the butterfly into two stages, as suggested by the nonholonomic nature of the system: planning a nominal trajectory and feedback stabilization of that trajectory. Figure 6 suggests that gradient-descent approaches could be applicable to the trajectory planning problem; see, for example, (Divelbiss and Wen [4]; Fernandes *et al.* [5]; Lynch and Mason [8]). The optimization could also solve simultaneously for shape parameters according to some cost function weighting shape and motion. Since the motion of the disk is solved for by simulation, care must be taken to avoid numerical problems, especially if finite differences are used to approximate gradients.

7 Conclusion

We have derived the dynamic equations of nonprehensile rolling and used a simulation based on them to find shape and motion solutions to the planar butterfly problem. We have successfully implemented a solution on FLATLAND, our testbed for dynamic manipulation. Feedback control is used to stabilize the planned trajectories. Future work includes automatic trajectory planning for dynamic rolling manipulation; more robust nonlinear feedback control; designing manipulation surfaces to achieve a desired rolling motion with a low degree-of-freedom robot; and investigating the roles of shape and motion in other kinds of dynamics, such as impact in vibratory parts feeding.

Acknowledgments

This work was performed while the first author was an STA postdoctoral fellow at the Biorobotics Division of the Mechanical Engineering Laboratory. We thank the Science and Technology Agency of Japan and the Robotics Department of MEL for their support. We especially thank Garth Zeglin and Matt Mason for suggesting the butterfly problem. Their air table setup at Carnegie Mellon inspired many of the ideas used in FLATLAND.

References

[1] 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.

[2] C. Cai and B. Roth. On the spatial motion of a rigid body with point contact. In *IEEE International Conference on Robotics and Automation*, pages 686–695, 1987.

[3] A. B. A. Cole, J. E. Hauser, and S. S. Sastry. Kinematics and control of multifingered hands with rolling contact. *IEEE Transactions on Automatic Control*, 34(4):398–404, Apr. 1989.

[4] A. W. Divelbiss and J. Wen. A global approach to nonholonomic motion planning. In *IEEE International Conference on Decision and Control*, pages 1597–1602, 1992.

[5] C. Fernandes, L. Gurvits, and Z. Li. Attitude control of a space platform/manipulator system using internal motion. *International Journal of Robotics Research*, 13(4):289–304, 1994.

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

[7] J. Kerr and B. Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4(4):3–17, 1986.

[8] K. M. Lynch and M. T. Mason. Dynamic underactuated nonprehensile manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 889–896, 1996.

[9] K. M. Lynch and M. T. Mason. Dynamic manipulation with a one joint robot. In *IEEE International Conference on Robotics and Automation*, pages 359–366, 1997.

[10] D. J. Montana. The kinematics of contact and grasp. *International Journal of Robotics Research*, 7(3):17–32, June 1988.

[11] N. Sarkar, X. Yun, and V. Kumar. Control of contact interactions with acatastatic nonholonomic constraints. *International Journal of Robotics Research*, 16(3):357–374, June 1997.

[12] T. Suzuki, M. Koinuma, and Y. Nakamura. Chaos and nonlinear control of a nonholonomic free-joint manipulator. In *IEEE International Conference on Robotics and Automation*, pages 2668–2675, 1996.