

A Family of Robot Control Strategies for Intermittent Dynamical Environments

M. Bühler, D. E. Koditschek, and P. J. Kindlmann¹

Center for Systems Science
Yale University, Department of Electrical Engineering

Abstract

We have developed a formalism for describing and analyzing a very simple representative of a class of robotic tasks which require “dynamical dexterity,” among them the task of juggling. We review our empirical success to date with a new class of control algorithms for this task domain that we call “mirror algorithms.” The formalism for representing the task domain and encoding within it the desired robot behavior enables us to prove that a suitable mirror algorithm is correct with respect to a specified task.

1 Introduction

We are interested in robotic task domains involving dynamical environments. In this paper we consider a simple representative from a range of robotic tasks associated with dexterous capabilities that might be grouped under the general rubric of “juggling”. We understand this term to include those tasks requiring throwing and catching, or (as in this paper) beating and batting, or any other interaction with an object (or multiple objects) which would otherwise fall freely in the earth’s gravitational field. Such tasks share the property of presenting non-trivial dynamical environments whose characteristics change intermittently subject to excitation from the robot. It seems fair to say that the only systematic work in this realm to date has been the pioneering research of Raibert whose careful experimental studies verify the correctness of his elegant control scheme [5]. Our strategy for research in this area is to study carefully a very simple experimental apparatus and develop a theoretical perspective which both explains the particular empirical experience and generalizes to a synthesis procedure over the larger task domain. The present paper reviews our own experimental results and presents a body of theory which accomplishes the first goal.

The paper is organized as follows. The experimental setup and a simplified mathematical model are presented in Section 2. Analysis of the “contact geometry” between robot and environment gives rise to an impact model — the “environmental control system” — with respect to which the “vertical one-juggle” task is formally defined and proven to be achievable in Section 3. Next, in Section 4, we offer a review of previous results, introduce a family of robot control strategies arising from a “mirror geometry” in the phase space of the robot-environment pair, demonstrate that this family solves the environmental control problem — our rigorous formulation of the juggling task expressed in terms of the contact

¹This work has been supported in part by PMI Motion Technologies, GMF Robotics Corporation, INMOS Corporation and the National Science Foundation under a Presidential Young Investigator Award held by the second author.

geometry — and present experimental data attesting to the physical validity of this strategy. Essentially, we have designed our controller so that the desired juggling pattern is an attracting periodic orbit of the closed loop robot-environment dynamics. The conclusion offers some speculations upon the larger implications of this work for robotic tasks in more general intermittent dynamical environments.

2 The Empirical and Analytical Setting

In this section we introduce our experimental apparatus — the Yale Planar Juggling Robot — and develop a simplistic mathematical model of the physics relevant to the juggling task.

2.1 Mechanical and Computational Setup

The physical apparatus consists of a puck, which slides on an inclined plane and is batted successively by a simple “robot”: a bar with billiard cushion rotating in the juggling plane as depicted in Figure 1.

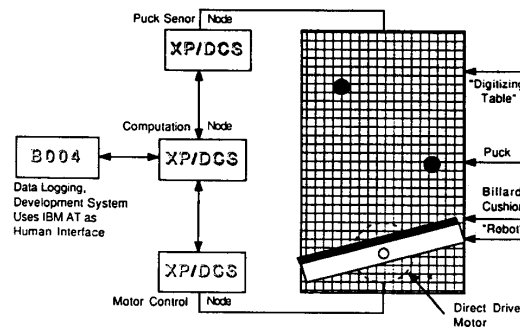


Figure 1: The Yale Juggler

All intelligent sensor and controller functions are performed by a four node distributed computational network formed from Yale XP/DCS control nodes [4]. In addition to the 10 MIPS, 1.5 MFLOPS and fast (20, 10 or 5 Mbits/s) interprocessor communications rate contributed by the INMOS Transputer, our XP/DCS boardset features fiber optics support, fast RAM, as well as support for fast and extensive I/O via a bidirectional latched 32 bit I/O bus.

In order to move the bar according to some puck dependent control algorithm, the puck's position and velocity must be measured. Presently, this is accomplished by placing an oscillator inside the puck and burying a grid in the juggling plane, thus imitating a big digitizing tablet. The sensor node measures induced voltages and computes the puck positions from the zero and first order moments. These, in turn, feed into a standard linear observer to reduce measurement noise in position and estimate velocity. The output of the observer is communicated asynchronously via fiber optics to the juggle planning and control node at a rate of 1 kHz. There, a reference trajectory for the motor is computed according to the mirror algorithm that we describe below, and communicated via a second fiber optic link to the Motor Controller Node. The latter issues torque commands to the motor at a rate of 2KHz, performing noise filtering and numerous additional housekeeping and safety checks as well. The fourth node is used as a logging system and human interface. At the time of writing we are completing a real time vision system based on the XP/DCS in order to move off the plane and juggle in three space.

2.2 A Simplified Mathematical Model

Locate a frame of reference, \mathcal{F}_0 , at the center of the robot shaft, with x -axis perpendicular to the plane, and z -axis defined by the projection of a vertical ray pointed directly into the earth's gravitational field onto the plane, as depicted in Figure 2. Define q so that it measures the angle of the right hand portion of the robot's bar (with the hitting surface — the billiard cushion — facing up) away from the x -axis on the juggling plane.

2.2.1 The Robot and Environment Models

The configuration space of the entire problem is the cross product, $\mathcal{C} \triangleq \mathcal{B} \times \mathcal{Q}$, of the environment and the robot configurations. We will model the robot's configuration space as $\mathcal{Q} \triangleq [-\pi/2, \pi/2] \subset \mathbb{R}$, which we restrict to a half revolution, since for present purposes, it will suffice to consider only those locations of the bar in the right half of the juggling plane for which the hitting billiard cushion is facing up. We will represent the location of the falling body on the plane $\mathcal{B} = \mathbb{R}^2$ with the coordinates (b_1, b_2) denoting, respectively, the position of its centroid relative to the "horizontal" (y) and "vertical" (z) axes of the reference frame, \mathcal{F}_0 .

In isolation, the robot's dynamics occur in its phase space, $\mathcal{R} \triangleq T\mathcal{Q} \approx \mathcal{Q} \times \mathbb{R}$, of angular positions and velocities, and may be modeled simply by the equations

$$\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} \triangleq \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} r_2 \\ \frac{v}{\rho} \end{bmatrix}, \quad (1)$$

(where v denotes the commanded torque from the motor control node and ρ denotes the moment of inertia of the bar) since the motor, with its high bandwidth and power, low shaft friction and inertia deployed in the absence of any transmission comes close to providing a source of "pure torque".¹

¹Unfortunately, the large mass of this motor mitigates against its role in a multi-jointed direct drive robot.

In isolation, the puck's dynamics occur in its phase space, $\mathcal{W} \triangleq TB \approx \mathcal{B} \times \mathbb{R}^2$, and may be modeled simply by the equations

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix} \triangleq \begin{bmatrix} \dot{b} \\ \ddot{b} \end{bmatrix} = \begin{bmatrix} w_2 \\ a \end{bmatrix} \triangleq n(w), \quad (2)$$

(where $a = [0, -\gamma]^T$) since we assume that the puck is a point of unit mass sliding on a frictionless surface. In fact this idealized model is overly simplistic, since there is noticeable coulomb friction on the sliding plane. One of the objectives of our study is to develop a control procedure which is robust enough to succeed even in the face of such unmodeled dynamics, and we will use only n from (2) in the formal analysis. However, in the sequel, we will find it interesting to compare numerical simulations of the robot control laws in the idealized environment, n , with the same strategies run in the more realistic simulation model with friction, as against empirical data.

Finally, the set of all possible impact configurations may be described by a smooth surface, \mathcal{I} , in the puck-robot configuration space, \mathcal{C} , as formalized in [3, Lemma 2.1]. An impact configuration, $(q, b) \in \mathcal{I}$, implicitly defines the robot's "virtual gripper" — the point of contact on the billiard cushion — and it is useful to define a new "virtual gripper frame", \mathcal{F}_1 whose origin is in the body's center, b , whose x -axis is parallel to that of \mathcal{F}_0 , but whose y -axis is aligned with the robot bar, all depicted in Figure 2. The new frame has a representation with respect to the "base frame" given by

$${}^0\mathcal{F}_1 = \begin{bmatrix} R & b \\ 0^T & 1 \end{bmatrix}; \quad R \triangleq \begin{bmatrix} \cos q & -\sin q \\ \sin q & \cos q \end{bmatrix}. \quad (3)$$

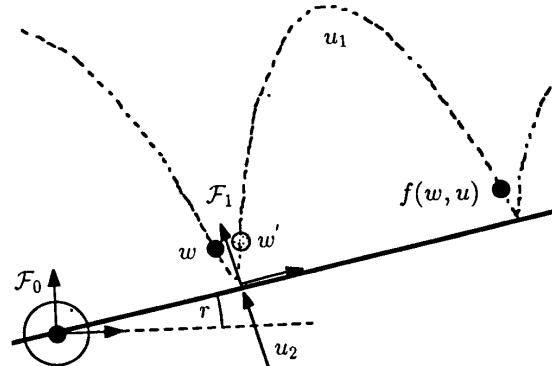


Figure 2: The Impact Event

2.2.2 The Impact Model

We now develop a simplified model of the dynamics of repeated puck-robot impacts based upon the following assumptions. First, we assume here that all interactions between ball and robot during impact can be adequately modeled as an instantaneous event: à posteriori velocities are related to à priori velocities via a simple "coefficient of restitution", $\alpha \in (0, 1)$ [6]. Second, we assume that the robot mass is sufficiently large as to make the puck's mass negligible. Finally, we neglect puck spin and it is assumed that the puck's velocity component parallel to the robot bar is unchanged by the impact.

Under these assumptions, the a posteriori velocity of the body after impact, b' is related to the a priori velocity of the body b , and that of the robot's virtual gripper, $u_2 \triangleq \|b\| \cdot \dot{q}$, in the \mathcal{F}_1 coordinates as

$$\dot{b}' = \begin{bmatrix} 1 & 0 \\ 0 & -\alpha \end{bmatrix} \dot{b} + \begin{bmatrix} 0 \\ 1 + \alpha \end{bmatrix} u_2 = \bar{C} \dot{b} + \bar{c} u_2,$$

and this is expressed in \mathcal{F}_0 coordinates as

$$\dot{b}' = C\dot{b} + cu_2, \quad (4)$$

where

$$C \triangleq R\bar{C}R^T; \quad c \triangleq R\bar{c}.$$

Recall that R , and, hence, C, c , are all functions of q . But since $q = \theta(b)$ for all impact configurations, $(b, q) \in \mathcal{I}$, we obtain \dot{b}' purely as a function of (b, \dot{b}) , and \dot{q} .

The forward trajectory of the body is now obtained by integrating its motion in \mathcal{W} starting from the initial conditions, $w = (b, \dot{b}')$, according to the isolated dynamics, $n(w)$, given in (2),

$$w(t) = \begin{bmatrix} b + \dot{b}'t - \frac{1}{2}at^2 \\ \dot{b}' - at \end{bmatrix}. \quad (5)$$

3 The Environmental Control Problem

In this section we investigate the response of the puck to all logically possible impact events by examining the *environmental control system*. This results from considering the effect of repeated puck-robot impacts on the future puck trajectory, (5), assuming arbitrarily assigned values for the robot, $(q, \dot{q}) \in \mathcal{R}$, at each impact event, independent of the robot's dynamics, (1). From (5) it is clear that the first time, t_{j+1} , after the j^{th} impact at time t_j , at which the robot and body again make contact, $(b(t_{j+1}), q(t_{j+1})) \in \mathcal{I}$, is a function of the robot's future position trajectory, $q(t_{j+1})$. Moreover, from (4), it is clear that the velocity of the virtual gripper at impact is determined by choice of the robot's velocity at impact, $\dot{q}(t_j)$. In the sequel we will use the term *impact schedule* to denote a sequence of pairs,

$$\{u(t_j)\}_{j=0}^{\infty}; \quad u(t_j) = \begin{bmatrix} u_1(t_j) \\ u_2(t_j) \end{bmatrix} \in \mathcal{U} \triangleq \mathbb{R}^2,$$

where $u_2(t_j)$ denotes the velocity of the virtual gripper at the moment of the j^{th} impact which occurs at time $t = t_j$, and $u_1(t_j) \triangleq t_{j+1} - t_j$ denotes the interval of time which elapsed between that impact event and its successor. An impact schedule gives rise to a sequence of puck states measured just after impact,

$$\{w(t_j)\}_{j=0}^{\infty}; \quad w(t_{j+1}) = f(w(t_j), u(t_j)),$$

where $f: \mathcal{W} \times \mathcal{U} \rightarrow \mathcal{W}$, is derived by substituting (4) into (5) to obtain

$$f(w, u) = \begin{bmatrix} b + (C(b)\dot{b} + c(b)u_2)u_1 - \frac{1}{2}au_1^2 \\ C(b)\dot{b} + c(b)u_2 - au_1 \end{bmatrix}. \quad (6)$$

This nonlinear discrete dynamical control system comprises the environmental control system. An *environmental control problem* results from prescribing some desired sequence of puck states, $\{w^*(t_j)\}_{j=0}^{\infty}$, and asking for an impact sequence, $\{u^*(t_j)\}_{j=0}^{\infty}$, which results in asymptotic convergence of $w(t_j)$ to $w^*(t_j)$.

Clearly, any control problem may be solved by a great variety of controller structures. In this section we shall be solely concerned with solutions via pure feedback compensation: namely, we shall abstract away all physical properties of the robot and presume it to be an "ideal" feedback agent which measures puck states, $w(t_j)$, and delivers control inputs, $u(t_j)$, accordingly. This point of view affords a precise definition of the juggling task in Section 3.1, as well as the demonstration that the task is at least logically achievable in Section 3.2

3.1 The Vertical One-Juggle

Probably the simplest systematic behavior of this environment imaginable (after the rest position), is a periodic vertical motion of the puck in its plane. Specifically, we would like to be able to specify an arbitrary "apex" point in the juggling plane, and from arbitrary initial puck conditions, force the puck to attain a purely vertical periodic trajectory with the specified apex point.

Since a purely vertical trajectory requires zero horizontal velocity, $b_1 = 0$, and a fixed vertical impact velocity, \dot{b}_2^* , from a specified impact height, $b_2 = 0$ implies a specified zenith position, we are led to the following definition. Let the task subspace of the vertical one-juggle be the plane

$$\mathcal{T} \triangleq \text{span} \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} = \{w \in \mathcal{W} : b_1 = 0, b_2 = 0\}.$$

We will say that a feedback law, $g: \mathcal{W} \rightarrow \mathcal{U}$, constitutes a *vertical one-juggle* with respect to the task, $w^* \in \mathcal{T}$, if w^* is a fixed point of the closed loop system,

$$w^* = f_g(w^*); \quad f_g(w) \triangleq f(w, g(w)),$$

and is a stable attractor of the resulting discrete dynamics.

Proposition 3.1 ([3]) *Given the discrete dynamical control system, (6), and a point, $w^* \in \mathcal{W}$, there exists a feedback law, $g: \mathcal{W} \rightarrow \mathcal{U}$ such that w^* is a fixed point of the closed loop map, $w^* = f_g(w^*)$, if and only if*

$$(i) \quad w^* \in \mathcal{T};$$

$$(ii) \quad g(w^*) = u^* \triangleq \begin{bmatrix} -2/\gamma \\ -\frac{1-\alpha}{1+\alpha} \end{bmatrix} \dot{b}_2^*.$$

This result shows, on the one hand, that only a point in \mathcal{T} may be fixed by feedback, and, on the other hand, that an appropriate constant, u^* may be found to fix any point of \mathcal{T} .

3.2 Local Stabilizability of the Task Plane

We next observe that the system is locally controllable at almost any point in the vertical one juggle task set. Let \mathcal{T}' denote the zero vertical velocity or zero horizontal position set points in \mathcal{T} ,

$$\mathcal{T}' \triangleq \{(b, \dot{b}) \in \mathcal{W} : b_1 = 0 \text{ or } \dot{b}_2 = 0\}.$$

Proposition 3.2 ([3]) *If*

$$w^* \in \mathcal{T} - \mathcal{T}',$$

and g fixes w^ , $f(w^*, g(w^*)) = w^*$, then system (6) is locally controllable at $(w^*, g(w^*))$.*

Local controllability, of course, implies local stabilizability. For, according to linear control theory, if (A, B) is a completely controllable pair then for any desired set of poles whose complex elements appear in conjugate pairs, $\Lambda = \{\lambda_i\}_{i=1}^4 \subset \mathbb{C}$, there exists a matrix, $K_\Lambda \in \mathbb{R}^{2 \times 4}$ such that the closed loop spectrum achieves that set, $\text{spectrum}(A + BK_\Lambda) = \Lambda$.

Now suppose that A and B denote the fixed system and control input matrices resulting from a local linearization analysis of (6) around a desired set point, w^* (that is, $A = [D_w f](w^*, u^*)$; $B = [D_u f](w^*, u^*)$). If the robot feedback algorithm, g , is chosen to be

$$g(w) \triangleq u^* + K_\Lambda(w - w^*), \quad (7)$$

it follows that any K_Λ for which $\Lambda \subset \mathcal{D}^1 \subset \mathbb{C}$ (the open unit disk in the complex plane) yields a feedback law, g , which achieves the vertical one-juggle as defined in Section 3.1. Thus, Proposition 3.2 demonstrates that the vertical one-juggle as defined in Section 3.1 is logically achievable.

4 Robot Implementation

The preceding analysis employed a geometric representation of the task domain in terms of a discrete dynamical control system on puck velocities over the contact set, \mathcal{I} . That analysis permitted a rigorous definition of the task at hand and the logical assurance of its possibility. We now turn our attention to the *robot control problem* — the synthesis of robot control laws guaranteed to result in impact schedules which accomplish a specified task. To this end, we will introduce a new geometric synthesis procedure defined on the entire cross product puck-robot phase space, $\mathcal{W} \times \mathcal{B}$, which represents the continuous physical trajectories of both rather than the discrete time evolution of their mutual impacts. This synthesis procedure gives rise to a family of robot control algorithms which we demonstrate empirically and prove mathematically to be correct.

Our resort to this new controller geometry is a consequence of the unexpected failure of robot controllers based upon the straightforward algorithm represented by equation 7. In an earlier paper [2] we give a detailed explanation for this empirical result. Roughly speaking, we attribute this failure on the one hand to limitations of local linear controller design, and on the other hand to the inappropriateness of the discrete task geometry for physically viable implementations in the real world of continuous trajectories and torque actuators.

In this section, we will describe our implementation of a successful vertical one-juggle on the physical apparatus sketched in Section 2.1, and review the formal proof of its correctness summarized from the account in [3]. We first introduce the “mirror algorithm” and its relationship to the geometry of the continuous puck-robot phase space, $\mathcal{W} \times \mathcal{R}$, in Section 4.1. In order to prove its correctness, we relate the continuous time geometry of the mirror algorithm to the discrete time geometry of the environmental control system in Section 4.2. Finally, we present data from the physical experiments performed using this algorithm in Section 4.3.

4.1 The Mirror Algorithm and its Implementation

We now introduce a control procedure grounded in the robot’s continuous time framework. Two different ideas are at work. First, we “translate” the desired impact sequence into an analytic function of the continuously changing state of the environment. Thus, the robot is merely required to track a reference trajectory generated online from a mechanical system. This promises to be more robust than a procedure relying explicitly on state measurement at impact. The second idea, borrowing from Raibert [5, 1], is to use the total energy of the environment in this function in order to stabilize the motion around a desired trajectory. If the desired behavior is periodic, then it may be characterized at any instant — for example at the moment of impact — by specifying some constant value for desired total energy.

To better convey the intuitive origins of the new algorithm, we will first discuss the problem of a prismatic one degree of freedom robot in a one degree of freedom environment, and then suggest how our solution scales to the present apparatus.

4.1.1 A Prismatic Robot in a One Degree of Freedom Environment

Suppose there is a single puck (of unit mass) constrained to fall in only the vertical direction, and a piston which moves up and down to strike it precisely. The objective is to bring the puck to a specified periodic orbit via successive impacts. Suppose that r and b measure the height of the robot and the puck, respectively. Notice that the trivial “mirror” law

$$r = -\kappa_{10}b; \quad \kappa_{10} = -\frac{1-\alpha}{1+\alpha}$$

already satisfies the fixed point conditions of Proposition 3.1. Assuming the robot is tracking accurately, impact is guaranteed to occur at zero height. The vertical puck trajectory is completely determined by its total vertical energy,

$$\eta(b, \dot{b}) = \frac{1}{2}\dot{b}^2 + \gamma b.$$

This fact can be conveniently used to enhance κ_{10} by a vertical energy error term to stabilize the above system at a fixed point:

$$r = -\{\kappa_{10} + \kappa_{11}[\eta_d - \eta(b, \dot{b})]\}b = -\kappa(w)b. \quad (8)$$

The reader may note that an impact occurs either in the case of exact tracking $r(t) \equiv b(t)$, or otherwise, only when both the robot and the puck achieve zero height. In the latter case, the robot describes a distorted “mirror” reflection of the puck’s trajectory. The reader may note as well that since we neglect friction during the puck’s flight, we may assume that $\dot{\eta} \equiv 0$, hence,

$$\dot{r} = -\kappa(w)\dot{b}.$$

Since the energy error term stays constant during flight (assuming, as we do, the absence of friction), in the tracking case, the robot would track the object during the entire flight. This can be considered a limiting case for κ_{11} .

Thus, we have a procedure which makes explicit use of the full puck trajectory, completely specifies the robot’s behavior, and results in a provably correct vertical one-juggle, when the robot and environment have the same “cartesian” degrees of freedom.

4.1.2 A Revolute Robot in a Two Degree of Freedom Environment

We now describe the manner in which this idea “scales” to the particular case at hand — the two degree of freedom cartesian environment presented in Section 3.

The basic idea carries over into this environment by just adding linear PD feedback compensation terms for the horizontal component. Define the “puck angle” as $\theta(b) = \text{atan}(b_2/b_1)$.

Now, as opposed to controlling the robot height as a function of puck height, we control the robot angle q as a function of puck angle:

$$q = -\kappa_1(w)\theta + \kappa_2(w).$$

In the particular set of experiments reported here, we have chosen

$$\begin{aligned} \kappa_1(w, w^*) &\triangleq \kappa_{10} + \kappa_{11}[\eta(w^*) - \eta(w)] \\ \kappa_2(w, w^*) &\triangleq \kappa_{21}(b_1 - \hat{b}_1^*) + \kappa_{22}(\dot{b}_1 - \dot{\hat{b}}_1^*) + \end{aligned}$$

$$\frac{1}{2} \frac{(w-w^*)^T M(w-w^*)}{[\kappa_{31} + ((w-w^*)^T M(w-w^*))^2]^2} + \frac{\kappa_{41}(\dot{b}_2 - \dot{\hat{b}}_2^*)}{[\kappa_{42} + (\dot{b}_2 - \dot{\hat{b}}_2^*)^2]^2}, \quad (9)$$

where k_{ij} are fixed constant gains, and M is a symmetric matrix in $\mathbb{R}^{4 \times 4}$.

The first two terms in κ_2 are borrowed from standard linear feedback control theory, implementing proportional derivative feedback. Analyzing the linearized system at a fixed point with just these two terms in κ_2 results in an ill conditioned system — the controllability matrix is nearly singular. The last two terms in this expression were introduced to ensure complete controllability locally without confounding the favorable global properties of the algorithm.

4.2 Analytical Results

This summary of our analytical results is taken from the complete presentation in [3] to which the reader should refer for a more complete discussion as well as all proofs.

When the robot has achieved the reference “mirror” trajectory described above then the puck and robot trajectories lie on a “mirror surface”, $\mathcal{M} \subseteq \mathcal{W} \times \mathcal{R}$, in the cross product phase space,

$$\mathcal{M} \triangleq \left\{ (b, \dot{b}), (q, \dot{q}) \in \mathcal{W} \times \mathcal{R} : (q, \dot{q}) = \left(\mu(b, \dot{b}), (D\mu \cdot n)(b, \dot{b}) \right) \right\},$$

specified as the graph of the function $\mu(w) \triangleq \kappa_1(w) \cdot \theta - \kappa_2(w)$, and its derivative along the motion of the puck (2). Recall θ is the “puck angle”, and κ_i are the gain functions detailed in (9).

Examination of the intersection between \mathcal{M} and the velocities over the contact set, \mathcal{I} , reveals how to choose the gains in κ_1 (9) to achieve the fixed point conditions of Proposition 3.1 for any $w^* \in \mathcal{T} - \mathcal{T}'$. A central result, [3, Proposition 5.2], shows via projection of this intersection onto \mathcal{W} , that the robot’s “mirroring” motion induces a three dimensional invariant submanifold of the environmental control system (6). In consequence, [3, Corollary 5.3], the local stability behavior of any valid vertical one-juggle task, $w^* \in \mathcal{T}'$, may be adjusted by the appropriate choice of gains in κ_2 (9).

4.3 Empirical Results

We now present plots of simulated and experimental data in order to validate our simplified model used for analysis and to illustrate the utility of our analytical results for both the idealized mode as well as the real system.

Figure 3, a “recording” of a successful vertical one-juggle using all parameters derived from analytical procedures, nicely depicts the rapid convergence for initial conditions (in drop-off position) from any region within the puck’s workspace not too close to the origin — a kinematic singularity. Despite departures from the idealized model and the relatively large sensor noise discussed in the beginning of this section, it may be observed from this and the subsequent plots that our algorithm produces steady reliable juggling performance. We have recorded vertical one-juggle runs with hundreds of impacts without encountering any failures.

Figures 4.3 and 4.3 compare the responses of the analytical model with and without friction to the responses of our experimental setup for two different initial conditions. The steady state values in the horizontal position, b_1 , are very close around the desired value for all three curves. The plots of the vertical impact velocity, \dot{b}_2 , demonstrates, first, as we expected, that the effect of the unmodeled friction is a steady state deviation, which second, is rather accurately predicted by the model. Examining the transients, notice that the experimental transient responses for \dot{b}_2 (lower plots) consistently match the responses of the model with friction, as expected. However, for b_1 (upper plots) the experimental transient responses are closer to the much faster transient model responses *without* friction than to those of the model with friction. This favorable discrepancy is not completely understood at present. We suspect that the unmodeled effect of spin, for example, on the impact as well as flight, might be responsible for this benign discrepancy.

Recall that the last two terms in (9) were introduced to arbitrarily specify the local behavior — that is, to place the poles of the linearized system. However, no changes were observed experimentally: the effects of the linear terms are apparently very small and disappear in the measurement noise, or are dominated by unmodeled dynamics. This corroborates our comments in [2] concerning the relative insignificance of local stability properties in the present setting.

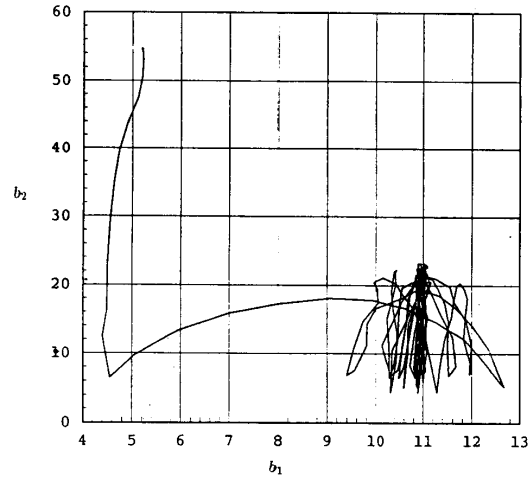


Figure 3: Sample continuous data

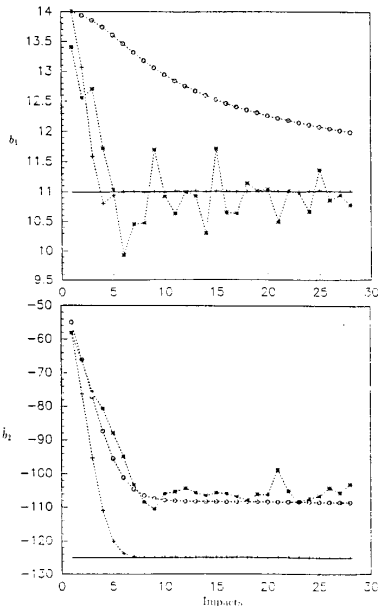


Figure 4: Initial puck position in lower right position of juggling plane

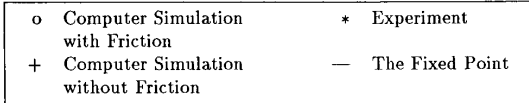
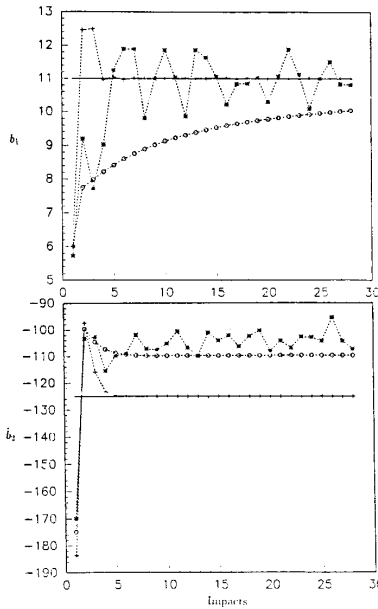


Figure 5: Initial puck position in upper left position of juggling plane

Conclusion

We have shown how the geometry of the impact configurations, Z , leads to a discrete dynamical model of the effect of robot impact strategies upon the behavior of an otherwise free falling puck. This model provides a framework for rigorously defining dexterous robotic tasks — for example, the “vertical one juggle” — and determining their feasibility. Although prescriptions for explicit impact strategies may be extracted from this model as well, it does not seem to offer an empirically viable framework for synthesis of robot control laws.

After many failed attempts to implement a logically correct but physically underconstrained and non-robust algorithm extracted from the discrete dynamics arising out of this “contact geometry,” we were led to a new type of control algorithm based on a completely different “mirror geometry” inhabiting the continuous phase space of the robot-environment pair. Experiments attest to the effectiveness of this control design. Moreover, analysis of the intersection between the mirror surface and the impact surface results in a correctness proof with respect to the discrete dynamical “environmental control system” that formally defines the task.

The central notion of robot controller synthesis via a “mirror geometry” in phase space appears to generalize to other interesting robotic tasks in this domain. For example, we have extended it to the task of catching falling objects and have applied it successfully to the task of juggling two pucks simultaneously as well. Retrospective correctness proofs notwithstanding, the generation of algorithm geometry is completely heuristic at present: each synthesis is empirically hand-tailored to fit the given task. Nevertheless, the analytical tractability of the resulting robot-environment closed loop as demonstrated here raises the hope that sufficient understanding may soon be realized to afford automatic translation of suitably expressed task definitions into provably correct and empirically valid robot controller designs.

In the longer term, we believe these ideas will have still wider application. For example, analytical techniques similar to those employed here result in correctness proofs for (simplified versions of) Raibert’s empirically verified legged locomotion algorithms [1]. Our juggler and Raibert’s hopper “settle down” to a characteristic steady state pattern because that pattern is an attracting periodic orbit of the closed loop robot-environment dynamics. Very likely, similar “natural” control mechanisms would make good candidates for gait regulation and other more complex tasks requiring controlled intermittent collisions with a dynamical environment.

References

- [1] M. Bühler and D. E. Koditschek. Analysis of a simplified hopping robot. In *IEEE International Conference on Robotics and Automation*, pages 817–819, Philadelphia, PA, Apr 1988.
- [2] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. A one degree of freedom juggler in a two degree of freedom environment. In *Proc. IEEE Conference on Intelligent Systems and Robots*, page , Tokyo, Japan, Oct 1988.
- [3] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. *Robotics in an Intermittent Dynamical Environment: A Prelude to Juggling*. Technical Report 8812, Center for Systems Science, Yale University, Jun 1988 (Revised Mar 1989).
- [4] F. Levin, M. Bühler, and D. E. Koditschek. The Yale Real-Time Distributed Control Node. In *Second Annual Workshop on Parallel Computing*, Oregon State University, Portland, OR, Apr 1988.
- [5] Marc H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.
- [6] J. L. Synge and B. A. Griffith. *Principles of Mechanics*. McGraw Hill, London, 1959.