




1990

A Simple Juggling Robot: Theory and Experimentation

M. Buehler

Follow this and additional works at: http://repository.upenn.edu/ease_papers

 Part of the [Electrical and Computer Engineering Commons](#), and the [Systems Engineering Commons](#)

Recommended Citation

M. Buehler, "A Simple Juggling Robot: Theory and Experimentation", *Experimental Robotics* 1, 35-73. January 1990.

Preprint version. Published in *Lecture Notes in Control and Information Science*, Volume 1, 1989, pages 35-73.

Publisher URL: <http://link.springer.com/book/10.1007/BFb0042509>

NOTE: At the time of publication, author Daniel Koditschek was affiliated with Yale University. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ease_papers/669

For more information, please contact repository@pobox.upenn.edu.

A Simple Juggling Robot: Theory and Experimentation

Abstract

We have developed a formalism for describing and analyzing a very simple representative of a class of robotic tasks which involve repeated robot-environment interactions, 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.” These new nonlinear feedback algorithms were motivated strongly by experimental insights after the failure of local controllers based upon a linearized analysis. We offer here a proof that a suitable mirror algorithm is correct with respect to the local version of the specified task — the “vertical one-juggle” — but observe that the resulting ability to place poles of the local linearized system does not achieve noticeably superior transient performance in experiments. We discuss the further analysis and experimentation that should provide a theoretical basis for improving performance.

For more information: [Kod*Lab](#)

Disciplines

Electrical and Computer Engineering | Engineering | Systems Engineering

Comments

Preprint version. Published in *Lecture Notes in Control and Information Science*, Volume 1, 1989, pages 35-73.
Publisher URL: <http://link.springer.com/book/10.1007/BFb0042509>

NOTE: At the time of publication, author Daniel Koditschek was affiliated with

Yale University. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

A Simple Juggling Robot: Theory and Experimentation

M. Bühler, D. E. Koditschek, and P. J. Kindlmann *
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 involve repeated robot-environment interactions, 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." These new nonlinear feedback algorithms were motivated strongly by experimental insights after the failure of local controllers based upon a linearized analysis. We offer here a proof that a suitable mirror algorithm is correct with respect to the local version of a specified task — the "vertical one-juggle" — but observe that the resulting ability to place poles of the local linearized system does not achieve noticeably superior transient performance in experiments. We discuss the further analysis and experimentation that should provide a theoretical basis for improving performance.

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

1 Introduction

Progress in robotics, as in any other science, requires an interplay of theory and experimentation. The inevitable tension between these opposites pulls analysis away from sterile abstractions and pragmatism from blind tinkering. This paper, presented in a symposium devoted to that interplay, offers an account of research partaking strongly of both modes of inquiry in the effort to achieve a level of dexterity relatively new to robotics: the ability to manipulate intermittent dynamical environments. We have built a one degree of freedom robot that can “juggle” two degree of freedom bodies falling otherwise freely in the earth’s gravitational field and bring them to a desired periodic orbit from a wide range of initial conditions and in-flight perturbations by a controlled schedule of impacts. Our efforts might be seen as a case study in this productive tension at work.

We began with an abstract formalism [4] for describing and analyzing the particular task under investigation, the “vertical one juggle.” This representation suggested a simple control strategy arising from straightforward application of standard tools of linear systems theory. The strategy did not work when implemented on our physical apparatus [6]. A very different strategy born of entirely intuitive reasoning and some physical tinkering achieved notable experimental success in the vertical one juggle [5] and generalized equally successfully to a variety of interesting extensions [7]. We have been able to furnish a local correctness proof of the empirically validated strategy with respect to the original formal model, and will present this proof in the present paper. We will show as well how the same theory entitles us (after further analytical tinkering with the successful strategy) to claim complete control over the local transients. However, the data we present reveals little empirical distinction between the theoretically slower and faster local transient gain settings. A paper presently in preparation shows how a return to the formalism affords a global correctness proof of the successful scheme assuming a radically simplified model of the robot in its dynamical environment. Further experimentation will reveal whether or not the control over global transient behavior resulting from this newest analysis is physically discernible.

The direct comparisons we display in the paper between simulation studies and experimental data show that our original formal model is empirically relevant despite its many simplifications and blindness to known physical effects. Thus, we have reached a critical juncture in our program of research. If our recent global analysis of the radically simplified model does not result in experimentally observable improvements in the intuitively conceived control strategy then we will be forced to pursue a global analysis of the physically validated model that may well prove intractable. In contrast, empirically successful predictions of this simplified model would strongly suggest that a generalizable body of physically relevant theory for robotics in intermittent dynamical environments is at hand. For this “simplistic” global theory both contains the linearized analysis and has a formal claim to a certain “universality” within discrete dynamical systems theory [8]. Thus, we would be well along the road to a geometric formalism that translates abstract human goals into automatic robot controllers.

We have no doubt that such a body of theory is both essential and possible. The first systematic work in this domain of robotic tasks has been the pioneering research of Raibert whose careful experimental studies verify the correctness of his elegant control strategies for legged locomotion [14]. Our analysis of simplified models of Raibert's hopper [9] uses the same global tools as in the simplified model of our juggler to make strong assertions about the transient and limit behavior of his machines that also remain to be empirically validated. On the other hand, McGeer has successfully used the kind of local linearized analysis that failed in our experiments to build passive (unpowered) walking robots [13, 12], and feels that similarly tractable analysis should suffice for controlling running machines as well [11]. Wang [18] has proposed to use the same local techniques for studying *open loop* robot control strategies in intermittent dynamical environments although his ideas remain to be tested as well. Research by Atkeson et al. on juggling [1] suggests that task level learning methods may relieve dynamics based (or any other parametric) controller synthesis methods of the need to achieve precise performance requirements as long as a basically functioning system has been assured. Thus, increasing numbers of researchers have begun to explore the problems of robotics in intermittent dynamical environments with increasingly successful results.

This paper is organized as follows. The experimental setup and original mathematical model are reviewed 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 via local linear analysis. Section 4 accounts for the failure of linear algorithm founded on this contact geometry: resulting impact schedules, while provably correct, are not robust and produce basins of attraction around the desired task point that are insufficiently large to be physically observable. Section 4.3 then motivates the intuitively developed successful class of nonlinear control algorithms. Briefly, trajectories in the puck phase space, are projected down to the robot phase space, yielding explicit robot reference trajectories. The projected trajectories have an intuitively appealing character which lead us to name this procedure the *mirror algorithm*. When our puck-robot system is forced toward this surface, that is, when we force the robot to track the trajectory specified by a mirror algorithm applied to a free falling puck, the result is a successful vertical one-juggle. In Section 5 we provide the local linear analysis of why our robot control strategy works. Specifically, we are able to prove that the family of environmental control laws arising from the robot "mirror algorithm" is correct with respect to the vertical one-juggle task set as defined in the original paper [4]. We first demonstrate that this family has the effect of inducing a lower dimensional invariant submanifold of the nonlinear discrete puck dynamics — the impact dynamics originally studied in [4]. We next show that the mirror algorithm family ensures that the local linearized dynamics around any valid task point, when restricted to this invariant submanifold, defines a completely controllable linear time invariant system. We finally observe that the new construction includes enough design freedom to afford pole placement with respect to that local linearized reduced dimensional control system. Thus, for any valid vertical one-juggle, we are able to synthesize a mirror law which attains the desired task while theoretically achieving arbitrarily specified local transient behavior. 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

This section describes our experimental setup and presents a simplified mathematical model which enables us to pose and solve robot juggling tasks as formal problems in control theory. The physical apparatus we use to obtain the data reported in this paper is depicted in Figure 1, described in Section 2.1. The mathematical models are presented in Section 2.2.

2.1 Experimental Apparatus

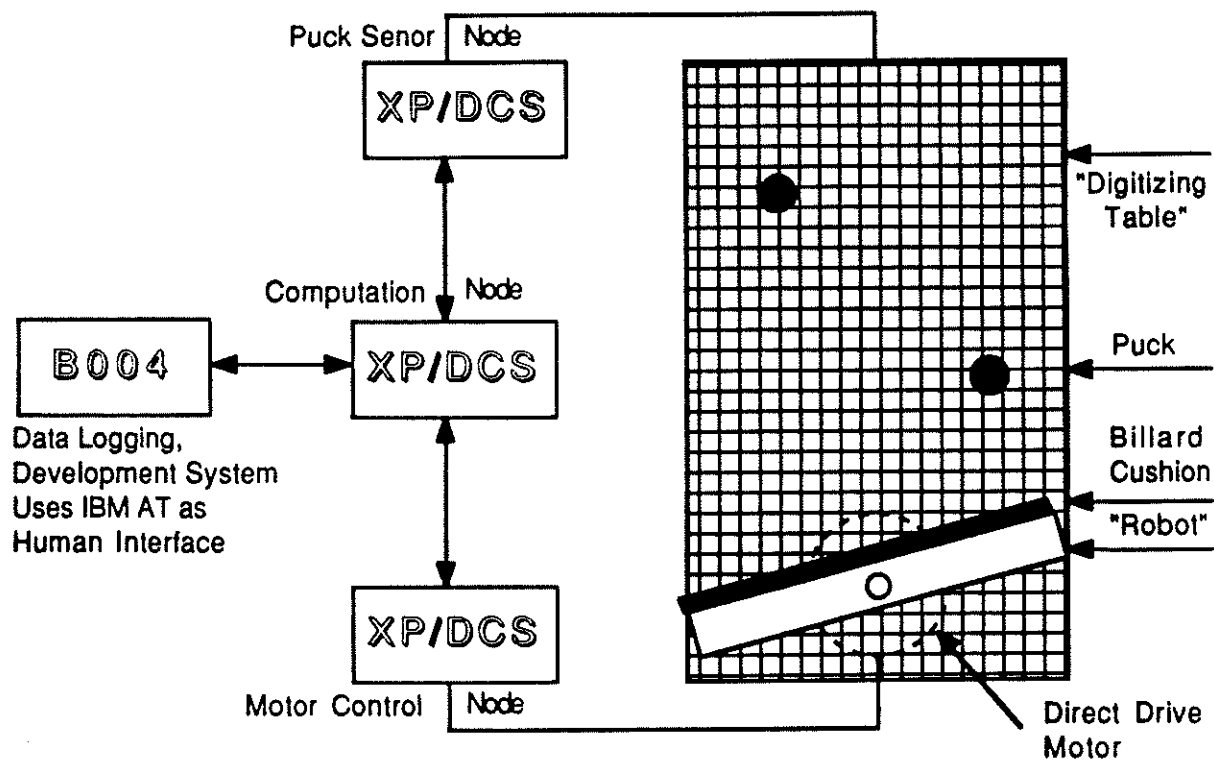


Figure 1: The Yale Juggler

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.

2.1.1 A Distributed Real Time Controller

All intelligent sensor and controller functions are performed by a four node distributed computational network formed from the INMOS transputer based Yale XP/DCS control node [10]. The INMOS T800 is a 32 bit 10 MIPS RISC chip with an integral floating point

unit capable of a sustained 1.5 Mflop computational rate. Each transputer provides four independent 5, 10 or 20 Mbits/sec serial DMA channels, so any node can communicate with up to four neighbors while simultaneously executing its own program with no effect upon either computational or communications rate (after initial start up overhead). Our XP/DCS CPU board complements the transputer's modular and flexible character by providing fast external memory, support for the four serial communication links, two fiber optic links, and an I/O expansion connector. The board's backplane connector is pin compatible with the INMOS ITEM Development System. The plug-in XP/DCS I/O board enhances the transputer's computational and communication power with a bidirectional latched 32 bit I/O bus with full handshaking support. Half of this board is allotted to a prototyping area allowing for easy customization to specific I/O needs. The cost of each mother/daughter board set at the time of writing is slightly over \$2000.

2.1.2 Distribution of Computational Resources to Mechanical Hardware

In order to move the bar according to some puck dependent control algorithm, the puck's position and velocity in both directions on the plane 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. On the back of the plane, a simplified XP/DCS system, the Puck Sensor Node, is used as a smart sensor. It measures the voltages induced in the sensing grid by the puck. The puck position in the plane is computed from the zero and first order moments. This information is used to estimate the puck's state: we use a standard linear observer to reduce measurement noise in position and velocity data. Each puck state measurement is communicated asynchronously via fiber optics to the Computation Node. This sampling and communication process is performed at a rate of 1kHz (when tracking one puck). We constructed an XP/DCS based real time (60Hz) stereo vision system that will enable us to move off the plane into three space soon.

The main node is the Computation Node which receives puck state information from the sensor node, reports logging data to the logging node, implements the control algorithm and issues the resulting desired robot states to the Motor Control Node. Various additional tasks like detecting the puck motion status (up, top, down, impact), predicting puck states (both used for extracting logging data, not for control) as well as extensive error checking and housekeeping tasks have to be performed on this node as well. The sampling time can vary between 500 and 1000 μ s. The Motor Control Node is dedicated to commanding a high torque dc servo actuator (donated to the robotics laboratory by PMI Motion Technologies) at a rate of 2kHz.

The experiences with the XP/DCS, the transputer and the development environment derived from this application are very encouraging. No single number can capture the ease of use and the little time spent with system overhead. Given the T800's intrinsic floating point capability, and the mathematical function library, formulas were programmed (in OCCAM, the "C-like" native compiler) almost directly from the blackboard with no

attempt at code optimization. In spite of substantial calculations, and a great deal of data logging and error handling overhead, very high sampling rates were achieved. The system operates capably in a high EMI environment in consequence of the low cost 5Mbits/sec fiber optic units from Hewlett Packard built into the Yale XP/DCS boards.

2.2 Mathematical Models

It is clear that this simple apparatus entails a multitude of complex physical effects: collisions between two partially elastic bodies; stiction, coulomb friction, and spin effects of the puck at impact; bouncing of the puck perpendicular to the plane; and so on. Yet since we desire a theory of robotics in intermittent dynamical environments, it is critical that our mathematical models of any particular physical setup be analytically tractable: simple enough to be amenable to formal proof; abstract enough to permit insights into a broader range of robotic devices and environments. In this section we present a simplified model which abstracts away as much of the extraneous physics as is possible while retaining the essential aspects of a robot and environment whose dynamics are intermittently coupled. The vindication of our claim to have retained the essential aspects, of course, is only possible by recourse to physical experiment, and will be provided in Section 4.

The central model of the robot and the environment is developed in Section 2.2.1: it preserves the disparity in degrees of freedom between them but abstracts away most of the dynamical complications introduced by spin, collision, and friction. Section 2.2.2 presents a greatly simplified account of the manner in which collisions between the robot and environment affect their behavior.

2.2.1 A Revolute Robot in a Two Degree of Freedom Environment

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 r 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.

The configuration space of the entire problem is the cross product, $\mathcal{C} \triangleq \mathcal{B} \times \mathcal{R}$, of the body and the robot configurations. 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 . For present purposes, without restriction of generality, we will only consider the right half of the juggling plane, $b_1 > 0$. We will model the robot's configuration space as $\mathcal{R} \triangleq [-\pi/2, \pi/2] \subset \mathbb{R}$, which we restrict to a half revolution where the hitting billiard

cushion is facing up on the right side of the juggline plane.

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

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} \triangleq \begin{bmatrix} \dot{r} \\ \dot{\tau} \end{bmatrix} = \begin{bmatrix} v_2 \\ \frac{\tau}{\rho} \end{bmatrix}, \quad (1)$$

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

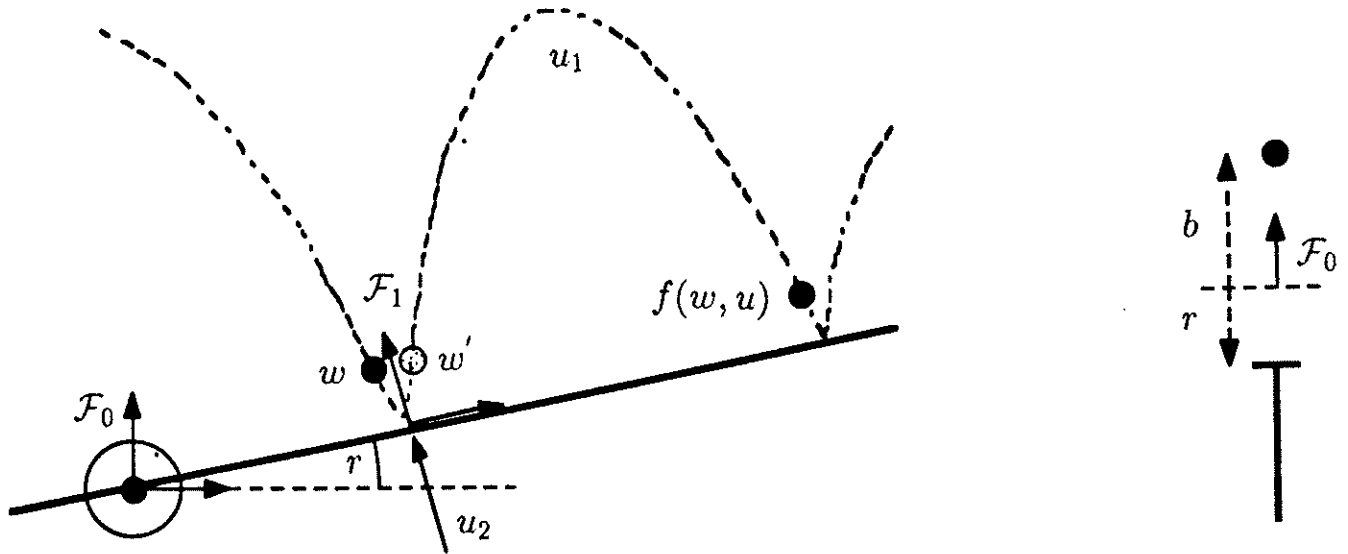


Figure 2: The Impact Event

The absence of gravitational torques on the robot's bar reflects its symmetric distribution of mass about the joint.

In isolation, the puck's dynamics occur in its phase space, $\mathcal{W} \triangleq T\mathcal{B} \approx \mathcal{B} \times \mathbb{R}^2$, and may be modeled simply by the equations

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

(where $a = [0, -\gamma]^T$, and $\gamma = \gamma_{grav} \sin \beta$ denotes the projection of the gravitational constant of acceleration, γ_{grav} , onto the sliding plane inclined away from vertical by the angle β) since we assume that the puck is a point of unit mass sliding on a frictionless surface.

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

In fact, this idealized model is overly simplistic since there is noticeable coulomb friction on the sliding plane, and a closer approximation to reality would replace n in (2) by

$$n'(w) \triangleq \begin{bmatrix} w_2 \\ c(w_2) + a \end{bmatrix}; \quad c(w_2) \triangleq \gamma_{coul} \begin{bmatrix} \text{step}(\dot{b}_1) - \text{step}(-\dot{b}_1) \\ \text{step}(\dot{b}_2) - \text{step}(-\dot{b}_2) \end{bmatrix},$$

where step denotes the unit step function. 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, n' , as against empirical data.

The set of all possible impact configurations may be described by a smooth surface in the puck-robot configuration space, \mathcal{C} , as formalized by the following statement which gives an implicit and explicit representation.

Lemma 2.1 *The set $\mathcal{I} \subset \mathcal{C}$ of configurations where the robot is in contact with the puck is a smooth submanifold specified as the zero set, $\iota^{-1}[0]$, of a scalar valued map, $\iota : \mathcal{C} \rightarrow \mathbb{R}$, which may be globally parametrized by a function of the body configurations, $\tilde{\theta} : \mathcal{B} \rightarrow \mathcal{C}$.*

Proof: Using the reference frame, \mathcal{F}_0 , the puck, $(b_1, b_2) \in \mathcal{B}$, contacts the bar, $r \in \mathcal{R}$ if and only if $\tan r = b_2/b_1$. Thus, denoting the “body angle” by,

$$\theta(b_1, b_2) \triangleq \text{atan}(b_2/b_1), \quad (3)$$

it is clear that \mathcal{I} is the graph of θ in \mathcal{C} . That is to say, \mathcal{I} is simply the image of the map

$$\tilde{\theta}(b) \triangleq \begin{bmatrix} b \\ \theta(b) \end{bmatrix},$$

or, alternatively, the zero level set of the map

$$\iota(b, r) \triangleq r - \theta(b).$$

□

An impact configuration, $(r, b) \in \mathcal{I}$, 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 (in two dimensional homogeneous coordinates) by

$${}^0\mathcal{F}_1 = \begin{bmatrix} R & b \\ 0^T & 1 \end{bmatrix}; \quad R \triangleq \begin{bmatrix} \cos r & -\sin r \\ \sin r & \cos r \end{bmatrix}. \quad (4)$$

2.2.2 The Impact and Flight Models

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 modeled as an instantaneous event: à posteriori velocities are related to à priori velocities via a simple “coefficient of restitution,” $\alpha \in (0, 1)$ [17]. We have previously considered dynamical interactions in a one degree of freedom problem [3] for which this assumption is invalid: we believe that a similar treatment will be possible in the present more complicated situation, but have not yet addressed the matter carefully. Second, we assume that the robot mass is sufficiently large as to make the puck’s mass negligible. We are obviously more interested in the general situation wherein this assumption is invalid, and, moreover, wherein the actual mass of the body is poorly known. Such problems will be considered in future experiments and analysis.

More formally, within the puck-robot phase space,²

$$\mathcal{X} \triangleq \mathcal{TC} = \mathcal{W} \times \mathcal{V},$$

consider the set of all possible velocities at each impact configuration,

$$T_I\mathcal{C} \triangleq \bigcup_{(b,r) \in \mathcal{I}} T_{(b,r)}\mathcal{C} = \left\{ ((b,r), (\dot{b}, \dot{r})) \in \mathcal{X} : (b,r) \in \mathcal{I} \right\}. \quad (5)$$

The first assumption amounts to the hypothesis of a “collision map,” $c : T_I\mathcal{C} \rightarrow T_I\mathcal{C}$ which takes an à priori phase at contact, $x \in T_I\mathcal{C}$, into a new phase, $x' = c(x) \in T_I\mathcal{C}$, in the same contact configuration — in coordinates, $(b', r') = (b, r)$. The second assumption may be expressed in coordinates as the hypothesis that $\dot{r}' = \dot{r}$. It is left to describe a map

$$\dot{b}' = c((b,r), (\dot{b}, \dot{r})).$$

Consider, first, the simplified case shown in Figure 2. The à posteriori velocity of the body after impact, \dot{b}' is related to the à priori velocity of the body \dot{b} , and that of the robot’s gripper, \dot{r} as

$$\dot{b}' = -\alpha \dot{b} + (1 + \alpha) \dot{r}. \quad (6)$$

In the full model the velocity of the revolute robot’s virtual gripper is $u_2 \triangleq \|b\| \cdot \dot{r}$ and will be considered in the subsequent section a robot control input. Let us now further assume that the puck’s velocity component parallel to the robot bar is unchanged by the impact. Then the à posteriori velocity of the two degree of freedom body after impact, \dot{b}' , is related to the à priori velocity of the body, \dot{b} , and that of the robot’s virtual gripper, in the \mathcal{F}_1 coordinates as

$$\dot{b}' = \begin{bmatrix} 1 & 0 \\ 0 & -\alpha \end{bmatrix} \dot{b} + \begin{bmatrix} 0 \\ (1 + \alpha) \end{bmatrix} \|b\| \dot{r} = \bar{C} \dot{b} + \bar{c} u_2, \quad (7)$$

²Throughout the paper, given a smooth manifold \mathcal{M} , we will denote its “tangent bundle” — the union over \mathcal{M} of the tangent vectors over each of its elements — by $T\mathcal{M}$.

which, expressed with respect to \mathcal{F}_0 , the world frame, is

$$\dot{b}' = C\dot{b} + cu_2 \triangleq c((b, r), (\dot{b}, \dot{r})). \quad (8)$$

where

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

Note that for each fixed configuration, $(b, r) \in \mathcal{I}$, c is linear in the velocities.³ Note that R , and, hence, C, c , are all functions on the configuration space. According to Lemma 2.1, \mathcal{I} is an immersion of \mathcal{B} : in other words, we may substitute $\theta(b)$ for all instances of r in (8), so that c is independent of the robot configuration value. It depends only upon b (in a nonlinear fashion) \dot{b} and \dot{r} (in a linear fashion as has been already noted). The independence of c from r enables us to consider the past and future history of the body at time of collision with no regard to the robot's behavior other than its velocity at the moment of impact.

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

$$w(t) = \begin{bmatrix} b + c(b, \dot{b}, \dot{r})t + \frac{1}{2}at^2 \\ c(b, \dot{b}, \dot{r}) + at \end{bmatrix}. \quad (9)$$

³In formal parlance, c is a *vector bundle morphism* of $T_{\mathcal{I}}C$.

3 The Environmental Control Problem

We may now 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 and is described completely by the discrete impact map (9), assuming arbitrarily assigned values for the robot inputs, $(r, \dot{r}) \in \mathcal{V}$, at each impact event. In other words, we treat the robot as an independent external “agent of control” and consider the various puck behaviors resulting from the robot’s “perfect” actions.

The formal control problem is developed in Section 3.1. The particular control task considered in the sequel is stated formally and shown to be logically achievable in Section 3.2. This mode of investigation exhibits some parallels to the seminal work of Schaefer and Cannon [15]. In both cases the problem involves an inherently unstable mechanical system — the stabilization of inverted pendulum setpoints and the stabilization of periodic juggling motions. In both cases, the linearized controllability analysis provides formal verification of the intuitive knowledge that the problems are solvable. There exist basic differences as well. In Schaefer and Cannon’s work, linear control theory is also employed successfully to solve the task. In contrast, in this paper we will show that linear theory does not provide a satisfactory design tool.

3.1 The Impact Schedule

From (9) 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, $(b(t_{j+1}), r(t_{j+1})) \in \mathcal{I}$, is a function of the robot’s future position trajectory, $r(t_{j+1})$. Moreover, (8) shows that the velocity of the virtual gripper at impact is determined by choice of the robot’s velocity at impact, $\dot{r}(t_j)$. In the sequel we will use the term *impact schedule* to denote a sequence of pairs,

$$\{u(t_j)\}_{j=0}^{\infty}; \quad u(t_j) = \begin{bmatrix} u_1(t_j) \\ u_2(t_j) \end{bmatrix} \in \mathcal{U} \triangleq \mathbb{R}^2, \quad (10)$$

in the robot’s *action set*, \mathcal{U} , where $u_2(t_j) = \|b\| \cdot \dot{r}$ denotes the velocity of the virtual gripper at the moment of the j^{th} impact which occurs at time 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 before 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 (8) into (9) to obtain

$$f(w, u) = \begin{bmatrix} w_1 + \dot{b}'(w)u_1 + \frac{1}{2}au_1^2 \\ \dot{b}'(w) + au_1 \end{bmatrix}. \quad (11)$$

where, recalling from (8),

$$\dot{b}'(w) = C(w_1)w_2 + c(w_1)u_2.$$

This nonlinear discrete dynamical control system constitutes 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 paper we shall be solely concerned with solutions via pure feedback compensation. In this particular section, 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 as well as the demonstration that the task is at least logically achievable in Proposition 3.1.

3.2 The Vertical One-Juggle

Before proceeding with the theoretical discussion, it seems worthwhile to provide a more intuitive explanation of what it means to stabilize a fixed point of the environmental control system (11). 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 want to be able to specify an arbitrary "apex" point, and from arbitrary initial puck conditions, force the puck to attain a periodic trajectory which impacts at zero position and passes through that apex point. This corresponds exactly to the choice of an appropriate fixed point, w^* , of (11).

To see this, consider first the one degree of freedom environment, $\mathcal{B} = \mathbb{R}$ as depicted in Figure 2. Selecting $w^* = (b^*, \dot{b}^*)$ as the desired constant set point indicates that we want the impact to occur at the position b^* and with the velocity just before impact given by \dot{b}^* . If w^* is truly a fixed point of the closed loop dynamics, then the velocity just after impact must be $-\dot{b}^*$, and this "escape velocity" leads to a free flight puck trajectory whose apex occurs at the height $b_{apex} = b^* + \frac{\dot{b}^{*2}}{2\gamma}$, assuming the simple ballistic model of free flight with no friction (2). Thus, a constant w^* "encodes" a periodic puck trajectory which passes forever through a specified apex point, b_{apex} .

In the two degree of freedom environment, $\mathcal{B} = \mathbb{R}^2$, the same notion applies, only in this case, a purely vertical steady state trajectory requires zero horizontal velocity, $\dot{b}_1^* = 0$. Again, a fixed vertical impact velocity, \dot{b}_2^* , from a specified impact height, b_2^* , implies a specified apex position. However, since the robot has only one degree of freedom, impacts at a position from which the virtual gripper frame is not oriented identically to

the base frame will impart some non-zero horizontal velocity, violating the condition that the steady state trajectory be purely vertical. Thus, in contrast to the one degree of freedom case, it is clear that not every constant set point, w^* is attainable in steady state for the two degree of freedom environment.

We may now restate these observations more formally. A robot feedback strategy is a map, $g : \mathcal{W} \rightarrow \mathcal{U}$, from the body's state to the robot's action set, \mathcal{U} (10), resulting in the impact strategy $u(t_j) = g(w(t_j))$. The robot-environment closed loop dynamical system is formed from the composition of f (11) with g ,

$$w(t_{j+1}) = f_g(w(t_j)); \quad f_g(w) \triangleq f(w, g(w)). \quad (12)$$

According to the definition in the previous section, a robot feedback strategy, g , accomplishes the constant set point environmental control problem, $w(t_{j+1}) = w(t_j) = w^*$, if and only if w^* is an asymptotically stable fixed point of f_g . We are now led to ask first which set points, w^* , can be made fixed points of (12) and second, of these, which can be made asymptotically stable by an appropriate choice of g . Given the specific juggling task at hand, we are only interested in a subset of the full puck impact phase space. As mentioned before we are limiting the setpoints to the right hand side $b_1 > 0$ of the juggling plane. Also, it only makes sense to admit negative vertical puck velocities $\dot{b}_2 < 0$ just before impact — i. e. those which point towards the robot's hitting bar. Thus we define the *working* puck phase space $\tilde{\mathcal{W}} \subset \mathcal{W}$ as

$$\tilde{\mathcal{W}} \triangleq \{w \in \mathcal{W} : b_1 > 0, \dot{b}_2 < 0\}.$$

In answer to the first question, our previous analysis reveals that only a subset of points in the task set, $\mathcal{T} \subseteq \tilde{\mathcal{W}}$, may be made fixed points of the closed loop system.

Proposition 3.1 *Given the discrete dynamical control system, (11), and a point, $w^* \in \tilde{\mathcal{W}}$, there exists a robot feedback strategy, $g : \mathcal{W} \rightarrow \mathcal{U}$ such that w^* is a fixed point of the closed loop map, f_g (12) if and only if*

(i) $w^* \in \mathcal{T}$, the vertical one-juggle task set, where

$$\mathcal{T} \triangleq \{w \in \tilde{\mathcal{W}} : b_2 = 0, \dot{b}_1 = 0\}.$$

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

Proof: Consider the fixed point condition

$$w = f(w, g(w)) = \begin{bmatrix} b + \dot{b}' u_1 + \frac{1}{2} a u_1^2 \\ \dot{b}' + a u_1 \end{bmatrix}$$

Elimination of $\dot{b}' = -\frac{1}{2}au_1^*$ directly results in

$$\begin{aligned} (1) \quad & \dot{b}_1^* = 0 \\ (2) \quad & u_1^*(w^*) = -\frac{2}{\gamma}\dot{b}_2^* \end{aligned}$$

With these conditions and

$$\dot{b}' = C\dot{b} + cu_2 = -\frac{1}{2}au_1$$

we obtain

$$\begin{aligned} (3) \quad & \dot{b}_2^* = 0 \\ (4) \quad & u_2^*(w^*) = -\frac{1-\alpha}{1+\alpha}\dot{b}_2^* \end{aligned}$$

where conditions (1),(3)[(2),(4)] are equivalent to (i)[(ii)].

□

Notice that within the allowable puck phase space, \mathcal{T} corresponds exactly to those constant set points which our previous intuitive thinking led us to understand would cause a purely vertical periodic puck trajectory which returns to the same apex point again and again. Thus we are led to call any set point $w^* \in \mathcal{T}$ a *vertical one-juggle task*, and say that a robot feedback strategy, g , constitutes a *vertical one-juggle* if such a point is an asymptotically stable fixed point of (12). In answer to the second question, our previous analysis showed that all points in the vertical one-juggle task set may be achieved.

Proposition 3.2 *If*

$$w^* \in \mathcal{T}$$

and g fixes w^ , $f_g(w^*) = w^*$, (12) then system (11) is locally controllable at $(w^*, g(w^*))$.*

Proof: According to Proposition 3.1, g fixes w^* if and only if $g(w^*) = u^*$. Thus the system is locally controllable if and only if

$$\begin{aligned} A &= [D_w f](w^*, g(w^*)) = [D_w f](w^*, u^*) \\ B &= [D_u f](w^*, g(w^*)) = [D_u f](w^*, u^*) \end{aligned} \quad (13)$$

comprise a completely controllable pair.

Taking partial derivatives of (11) gives

$$A = \begin{bmatrix} 1 & -\frac{4}{\gamma}\frac{\dot{b}_2^{*2}}{\dot{b}_1^*} & -\frac{2}{\gamma}\dot{b}_2^* & 0 \\ 0 & 1 & 0 & \frac{2}{\gamma}\alpha\dot{b}_2^* \\ 0 & 2\frac{\dot{b}_2^*}{\dot{b}_1^*} & 1 & 0 \\ 0 & 0 & 0 & -\alpha \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 \\ \dot{b}_2^* & -\frac{2}{\gamma}(1+\alpha)\dot{b}_2^* \\ 0 & 0 \\ -\gamma & 1+\alpha \end{bmatrix}$$

It suffices to show that four of the eight columns of the matrix (B, AB, A^2B, A^3B) are linearly independent. The four columns (B, A^2B) we consider are

$$\begin{bmatrix} 0 & 0 & \frac{4b_2^*{}^3}{\gamma b_1^*} (2\alpha - 3) & \frac{8b_2^*{}^3}{\gamma^2 b_1^*} (1 + \alpha)(3 - \alpha) \\ b_2^* & -\frac{2}{\gamma} b_2^* (1 + \alpha) & b_2^* (2\alpha^2 - 2\alpha + 1) & -\frac{2}{\gamma} b_2^* (1 - \alpha + \alpha^2) \\ 0 & 0 & 4\frac{b_2^*{}^2}{b_1^*} (1 - \alpha) & -\frac{4b_2^*{}^2}{\gamma b_1^*} (1 + \alpha)(2 - \alpha) \\ -\gamma & 1 + \alpha & -\alpha^2 \gamma & \alpha^2 (1 + \alpha) \end{bmatrix}.$$

The determinant of this matrix,

$$\frac{16}{\gamma^2} \frac{b_2^*{}^6}{b_1^*{}^2} \alpha (1 + \alpha)^2$$

is nonzero for any $w^* \in \mathcal{T}$.

□

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,

$$\Lambda = \{\lambda_i\}_{i=1}^n \subset \mathbb{C}$$

there exists a matrix, $K_\Lambda \in \mathbb{R}^{2 \times n}$ such that the closed loop spectrum achieves that set,

$$\text{spectrum}(A + BK_\Lambda) = \Lambda,$$

where n is the dimension of the dynamical system. Now suppose that the feedback algorithm, g , is chosen to be

$$g(w) \triangleq u^* + K_\Lambda(w - w^*). \quad (14)$$

Since

$$\begin{aligned} [D_w f_g](w^*) &= [D_w f](w^*, g(w^*)) + [D_u f](w^*, g(w^*)) [D_w g](w^*) \\ &= A + BK_\Lambda, \end{aligned}$$

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 3.2.

We have now shown that the vertical one-juggle is logically achievable. This discrete analysis formally confirms the intuition that only state information at impact should be required for a successful juggling algorithm — that full trajectory information is redundant.

However, we have said nothing yet concerning the ability of the robot to realize any particular feedback strategy, g , much less one which stabilizes a desired set point, w^* : it is completely up to the designer to solve the robot control problem and achieve an approximation to the required impact strategy.

4 Robot Implementation: Controller Synthesis in Task Geometry

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 that result in impact schedules which accomplish a specified task.

4.1 The Robot Control Problem

The robot control problem can be formulated as follows. Given a desired impact schedule — a solution to the environmental control problem, $\{u^*(t_j)\}_{j=0}^{\infty}$, which results in asymptotic convergence of $w(t_j)$ to $w^*(t_j)$, find an appropriate continuous robot control input $\tau(t)$ that forces the robot with the dynamics

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} \triangleq \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} v_2 \\ \frac{\tau}{\rho} \end{bmatrix}, \quad (15)$$

to implement the desired discrete impact sequence.

4.2 Direct Implementation of a Discrete Time Environmental Controller

The controllability analysis in Section 3.2 not only demonstrates that our task is possible: it also furnishes a robot control synthesis procedure based upon standard linear time invariant control theory. It seemed entirely plausible to us that a physical implementation of this straightforward procedure would result in empirical success. Indeed, our appeal to the mathematical formalism of Section 3 was partially motivated by the expectation that controller design could be based upon such well understood theoretical principles. To our surprise, these controllers failed when implemented on the experimental apparatus.

For a variety of task points, $w^* \in \mathcal{T}$, we chose a variety of stable spectra, Λ , in the open unit disk, \mathcal{D}^1 , determined K_Λ using a numerical procedure, and determined an ideal impact schedule according to the affine feedback law, g , described by (14). We then induced our robot to deliver a close approximation of this impact schedule via a reasonable ad hoc procedure. In no case of this series of experiments did we observe a successful vertical one-juggle. In Section 4.2.1 we describe in detail the implementation procedure. In Section 4.2.2 and 4.2.3 we offer a critique of this procedure and suggest several reasons for this unexpected outcome.

4.2.1 Implementation Procedure

Of course, the key to success or failure of even a practicable feedback scheme stemming from the discrete analysis would depend upon the details of how the robot is commanded to implement the impact schedule required by (14). For, as we have previously described, the analysis of the environmental control problem in Section 3, provides an abstract characterization of what is logically possible assuming that the robot is a perfect "feedback agent." It remains entirely silent concerning the manner in which a particular impact schedule is achieved. Several different implementation procedures were attempted; the best performance seemed to result from the following procedure:

1. Since the feedback algorithm (14) is based upon the states of the puck just before impact, a "start-up" procedure at time $j = 1$ is required. For simplicity, we start from the desired apex point, and assume that the first impact occurs at $\hat{w}_1 = w^*$: the robot is commanded to hit the puck with a velocity obtained from applying the feedback law (14) to the estimated impact state: $u_{j=1} = g(\hat{w}_1)$. Of course, in this case, $g(w^*)$ yields $u_{2,j=1} = u_2^*$ given in Proposition 3.1.
2. Just before the actual first impact, we measure and estimate the true state of the puck before impact, w_1 , and evaluate the feedback law, $g(w_1)$ to get a desired time interval to next hit, $\hat{u}_{1,j=1}$. Now we use the dynamical model (11) to predict \hat{w}_2 with u_2 set to the actual measured impact velocity, and u_1 set at the nominal value, $\hat{u}_{1,j=1}$.
3. The desired velocity of the next impact, $u_{2,j=2}$, is determined by applying the feedback law to the predicted next impact point, $g(\hat{w}_2)$. Now \hat{w}_2 prescribes the robot's angle, r , at second impact, and, together with $u_{2,j=2}$, prescribes the robot's angular velocity, \dot{r} at second impact.
4. To implement the desired robot states for the second impact, we use a simple open loop fixed torque control strategy which works as follows. Acceleration from a rest position to \dot{r} takes time t_{acc} . Together with r , this yields the robot's rest position. During the puck flight, its states are predicted t_{acc} in advance. When the predicted position crosses r , the robot starts accelerating until the second impact occurs. Measurements made after the fact show that this strategy yields satisfactory accuracy.
5. The procedure continues using the measured estimate of the forthcoming impact, w_j , to obtain a value for $u_{1,j}$ from $g(w_j)$, (14), and a predicted next impact state, \hat{w}_{j+1} , from the model (11), to generate a value for $u_{2,j+1}$ from $g(\hat{w}_{j+1})$.

4.2.2 Limitations of Linear Control Theory

The local nature of our task definition in Section 3.2 — asymptotic stability of a fixed point — afforded an easy proof that it may actually be attained, based upon standard

arguments from linear control systems theory. However, (11) is a highly nonlinear control system. The determination that a particular equilibrium state is asymptotically stable provides very little help in estimating the domain of attraction and the domain of "containment" around that state. The former is comprised of those initial conditions which tend asymptotically toward the equilibrium state, while the latter is comprised of those initial conditions which are guaranteed to remain in some specified neighborhood of the equilibrium state. Since our juggling plane, in contrast to its analytical model developed in Section 2.2, is not truly infinite, achieving a particular containment region will be crucial to the success of any real juggling strategy, regardless of its domain of attraction.

In fact, numerical simulation showed that the domain of containment of the idealized closed loop discrete dynamical system resulting from (14) was in the order of our sensor resolution and thus unacceptably small. For certain arbitrarily chosen values of w^* , pole assignments, Λ , could be found that resulted in a large domain of attraction. However, we could find no pole assignments for physically realizable settings of w^* which achieved a domain of attraction whose diameter along the b_2 axis of \mathcal{W} was greater than 6 % of the value b_2^* . In all of these simulations, the trajectories within the domain of attraction leave the physical boundaries of actual juggling plane. For typical settings, the domain of containment within the juggling plane was no larger than 2 % of the desired fixed point magnitude. This is smaller than the error tolerance of the puck position sensing system. We conclude that the linearized analysis of even this simplified discrete nonlinear system is inadequate to the desired task. In formal terms, the local definition of the task in 3.2 may be too weak.

Moreover, since the parameters of the ultimate closed loop system with spectrum Λ are chosen according to a numerical procedure that finds a matrix K_Λ , as a function of the pair (A, B) , a controllable but *ill conditioned* pair may result in large departures of the closed loop parameters from their desired values either because the numerical procedure is very sensitive or because of small departures in implementation from the numerically determined value of K_Λ . Thus, pole placement in the face of ill conditioned data is not robust. Indeed, for all values of w^* examined, numerical tests revealed that the pair (A, B) , while completely controllable as guaranteed by Proposition 3.2, was poorly conditioned. We conclude that our experimental gain settings were not sufficiently close approximations of K_Λ even to guarantee stability, much less containment.

4.2.3 Limitations of the Discrete Impact Model

As mentioned at the end of Section 3, an initially attractive feature of the discrete analysis is that it confirms the intuition that only state information at impact should be required for a successful juggling algorithm — that full trajectory information is redundant. Conceptual appeal notwithstanding, in reality state information at or very near the impact event is exceedingly difficult to measure. Moreover, just as they necessarily rely upon the accuracy of the impact state measurement, algorithms resulting from the discrete analysis critically depend upon the accuracy of the impact parameters and idealizations of the

model — the coefficient of restitution; the validity of the zero friction assumption; a zero puck diameter — because this information is explicitly used to compute the next desired impact states and the next robot inputs. It is necessary, for example, to correct for the error between the next predicted and the actual impact position. How to do that, is left to the designer. These features do not contribute to a practicable scheme.

More fundamentally, as pointed out in the very beginning of the paper, this algorithm only specifies desired robot control inputs to the environment at the moment of impact. It is completely up to the designer to solve the robot control problem, and this leads to the kind of ad hoc implementations to which we resorted as described in Section 4.2.1.

Thus, even had the linearized analysis resulted in a robust feedback law with a physically realizable region of containment we feel that it would fail to provide a satisfactory synthetic framework in the present task domain. We were led to seek, instead, some means of generating successful and provably correct juggling behavior in the continuous time framework of the robot itself.

4.3 A New Algorithm

We now introduce a procedure which meets the desiderata of the previous paragraph. We introduce a new synthesis procedure defined on the entire cross product puck-robot phase space, $\mathcal{W} \times \mathcal{V}$, 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 here and prove mathematically (in Section 5) to be correct. For reasons that will soon become apparent, we call this synthesis procedure the “mirror algorithm.”

4.3.1 Intuitive Motivation

Intuitively, two different ideas are combined to produce an algorithm that is implemented on the robot by recourse to standard trajectory tracking techniques. First, we “reflect” the desired periodic puck trajectory in \mathcal{W} into a “distorted mirror image” in \mathcal{V} : the “distortion” is so designed that the “cross product” trajectory of the puck and robot reflection in $\mathcal{X} = \mathcal{W} \times \mathcal{V}$ intersects the impact set, $T_I\mathcal{C}$, with characteristic tangent vectors, (\dot{b}, \dot{r}) , whose image under the collision map, c (8), is “favorable” to the task at hand. Second, we borrow from Raibert [14, 3] the idea of modifying the robot’s trajectory by “servoing” on the discrepancy between the (constant) total mechanical energy of the puck in its desired steady state, and the currently measured value.

To better convey the nature of the new algorithm, we will first discuss the one degree of freedom case (Figure 2). Let the puck drop from the desired steady state height corresponding to a steady state value of $w^* = (0, \dot{b}^*) \in \mathcal{T} \subseteq \mathcal{W}$. Suppose the robot tracks

exactly the “distorted mirror” trajectory of the puck,

$$r = -\kappa_{10}b,$$

where κ_{10} is a constant. Contact between the two can occur only when $(r, b) = (0, 0)$. Therefore, in this case, the contact configurations are limited to a single configuration $\mathcal{I} = (r = 0, b = 0)$. Since $\dot{r} = -\kappa_{10}\dot{b}$, an impact at this configuration occurs at the phase point $s = ((0, 0), (\dot{b}, -\kappa_{10}\dot{b})) \in T\mathcal{I}\mathcal{C} \subseteq \mathcal{X}$. Now solving the equation $c(s) = -\dot{b}^*$ for κ_{10} , yields a choice of that constant,

$$\kappa_{10} = \frac{1 - \alpha}{1 + \alpha},$$

which ensures a return of the puck to the original height. Thus a properly tuned “distortion constant,” κ_{10} will maintain a correct puck trajectory in its proper periodic course — formally, as we will soon show, it achieves the fixed point condition of Proposition 3.1.

In the absence of friction, the desired steady state periodic puck trajectory is completely determined by its total vertical energy,

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

in this case,

$$\eta^* \triangleq \eta(w^*) = \frac{1}{2}\dot{b}^{*2}.$$

This suggests the addition to the the original mirror trajectory,

$$r = -\kappa(w)b; \quad \kappa(w) \triangleq \kappa_{10} + \kappa_{11}[\eta^* - \eta(w)], \quad (16)$$

of a term which “servos” around the desired steady state energy level. Since we neglect friction during the puck’s flight, we may assume that $\dot{\eta} \equiv 0$, hence,

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

At steady state, $\eta(w) = \eta^*$, the fixed point condition is still preserved. However, deviations of η away from η^* cause proportionately harder or softer robot impacts than the steady state condition requires. It is plausible that these proportionally adjusted deviations will cause convergence toward η^* : we will shortly prove that this is indeed the case.

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

$$\theta(b) \triangleq \text{atan} \frac{b_2}{b_1}.$$

Now, as opposed to controlling the robot height as a function of puck height, we control the robot angle r as a function of puck angle θ as shown in (17) where κ_{ij} are fixed scalar 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 first terms in κ_2 results in an ill conditioned system. The last two terms in this expression were introduced to ensure complete controllability locally without interfering with the global nature of the algorithm.

Thus, implementing a mirror algorithm is an exercise in robot trajectory tracking wherein the reference trajectory is a function of the puck's state.

$$\begin{aligned}
 r_d(t) &= -\kappa_1(w)\theta + \kappa_2(w). \\
 \kappa_1(w, w^*) &\triangleq \kappa_{10} + \kappa_{11}[\eta(w^*) - \eta(w)] \\
 \kappa_2(w, w^*) &\triangleq \kappa_{21}(b_1 - b_1^*) + \kappa_{22}(\dot{b}_1 - \dot{b}_1^*) + \\
 &\quad \frac{1}{2} \frac{(w-w^*)^T M(w-w^*)}{[\kappa_{31} + ((w-w^*)^T M(w-w^*))^2]^2} + \frac{\kappa_{41}(\dot{b}_2 - \dot{b}_2^*)}{[\kappa_{42} + (b_2 - b_2^*)^2]^2},
 \end{aligned} \tag{17}$$

4.3.2 Data Quality

Before discussing the plots, some remarks about the quality of the experimental data are called for. The inductive puck position sensor as described in Section 2.1, is accurate to about 0.5 inch, due to a systematic measurement error resulting from the nonlinear distribution of the puck's generated flux across the measurement loops as well as measurement noise in the electromagnetically sensitive inductive grid. The latter is exacerbated during operation due to noise emitted by the high power motor, most of which fortunately can be filtered out using a standard observer. This explains the quality of the data in Figure 3. For the impact data, two further detrimental effects occur. First, the impact states are somewhat difficult to obtain accurately: during the short but not truly infinitesimal impact event, the puck-robot system is moving, making it difficult to decide, when to take the measurement *during* the time of contact; in contrast, the velocity data must be measured *just before* the impact. Second, the solid metal robot distorts the magnetic field of the puck at time of impact. Of course all of these effects can be compensated for in a very time consuming approach. A simple satisfactory solution to compensate for the errors introduced at impact is to use a very slow filter for estimating the puck states and arguing that the dynamics of these effects is much faster, resulting in an improved measurement. Of course it has to be checked that the slow filter has converged to the true puck states well before the impact. Concluding this paragraph we can say that first, the actual puck trajectory during flight (Figure 3) is obviously smooth, and second, that the steady state error band around the desired impact states is actually smaller than depicted on the Figures 4 to 7.

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 model as well as the real system. In Section 5.2 we describe how some gain settings in (17) are derived readily from our analysis.

4.3.3 Validation of the Simplified Model

Figure 3, a continuous “recording” of a vertical one-juggle nicely depicts the rapid convergence typical for initial conditions (in drop-off position) from any region within the puck’s workspace not too close to the origin — a kinematic singularity. Due to the nonzero puck radius and robot juggling bar dimensions, the vertical impact position b_2^* , the sum of the two, is nonzero as well, as can be seen in this plot. Despite departures from the idealized model and the relatively large sensor noise discussed in the previous 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.

Next, we present data plots of the puck states just before impact. Each experimental data curve displays statistical information (mean plus/minus one standard deviation) obtained from 20 successive runs (without handpicking). We feel that this presentation offers a closer rendering of true performance than one based upon a handpicked best run. Such presentation methods are standard practice in other natural sciences, but seem only slowly to be entering the field of robotics [13].

Figures 4 to 7 compare the responses of the analytical model with and without friction to the responses of our experimental setup (with friction) for two different initial conditions. These two initial conditions for the puck impact states result from dropping the puck from two extremes of the right juggling half plane, the upper left (initial condition one) and the lower right corner (initial condition two). The steady state values in the horizontal position, b_1 , are very close around the desired value for all curves. The plots of the vertical impact velocity, \dot{b}_2 , demonstrate, 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 at impact might be responsible for this benign discrepancy.

4.3.4 Confirming the Limited Utility of Linear Analysis

Recall that the last two terms in (17) were introduced to arbitrarily specify the local behavior — that is, to place the poles of the linearized system around the fixed point.

For the experiments reported here, they were placed at $[0, \pm 0.1j]$ for fast convergence and very close to the unit circle at $[0.9, 0.9 \pm 0.1j]$ for slow convergence. Figures 4 to 7 show a side by side comparison of the "fast" system with the "slow" system for the two initial conditions. The experimental data confirms our limited ability to affect the local behavior. There is no clear improvement in the steady state error of the fast system. Our ability to place the local poles is demonstrated more drastically by the simulated data in Figure 8: We place the poles outside the unit circle and verify the unstable local dynamics. The same plot also reveals the strong nonlinear (global) properties of the mirror algorithm: the injected instable behavior is bounded to a small neighborhood around the fixed point. However, while we can verify control over local transients via simulations, the experimentally observed changes are rather small when taking into account the large difference in pole settings. For this system the interesting dynamic behavior seems to result from its global properties and therefore the local linear behavior is only of limited usefulness. Further remarks about the utility of the linear analysis can be found at the beginning of Section 5.

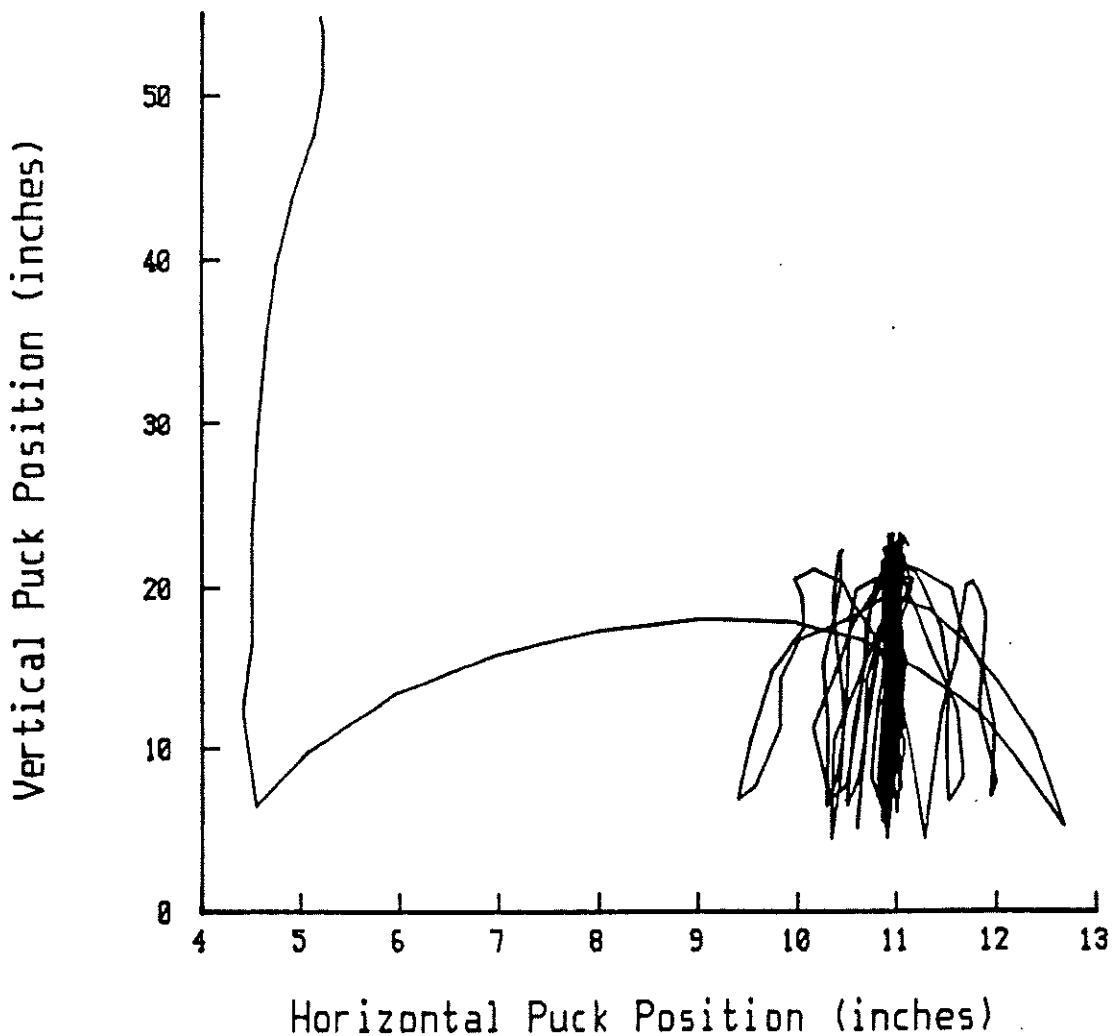


Figure 3: Sample continuous data

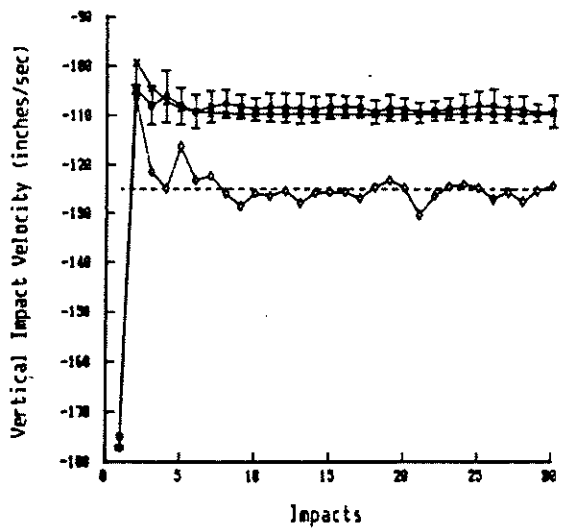
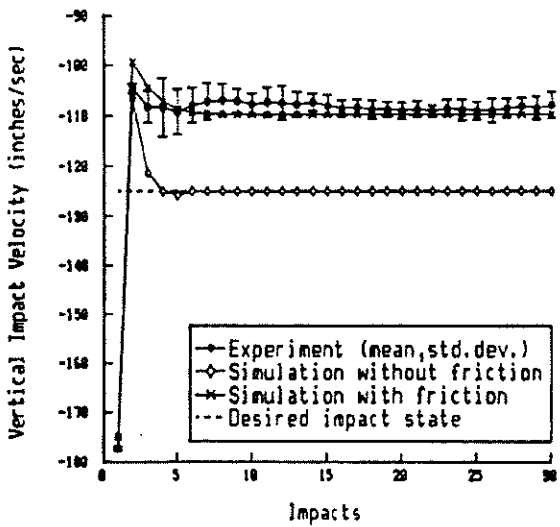
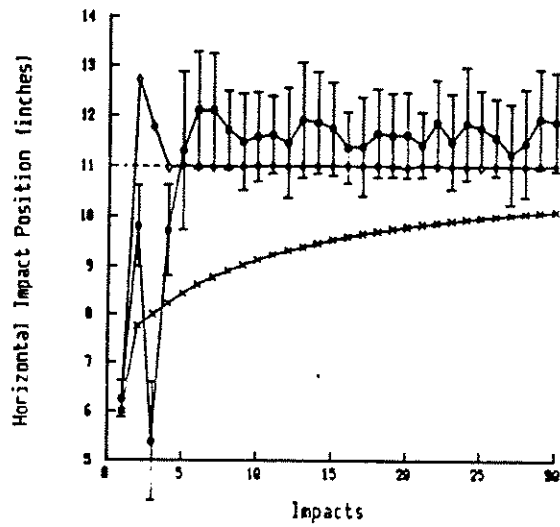
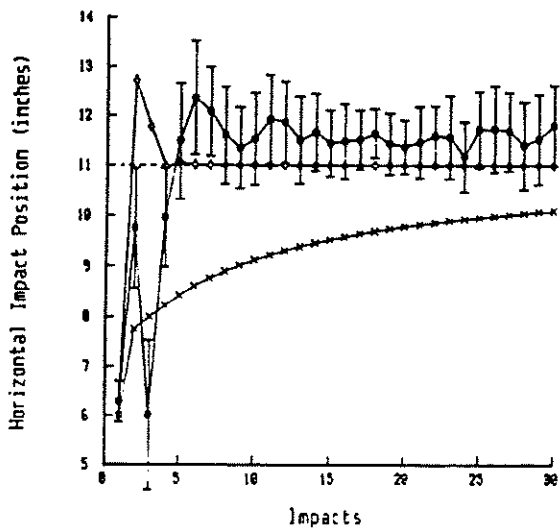


Figure 4: Fast local transient setting, Figure 5: Slow local transient setting, Initial condition one.

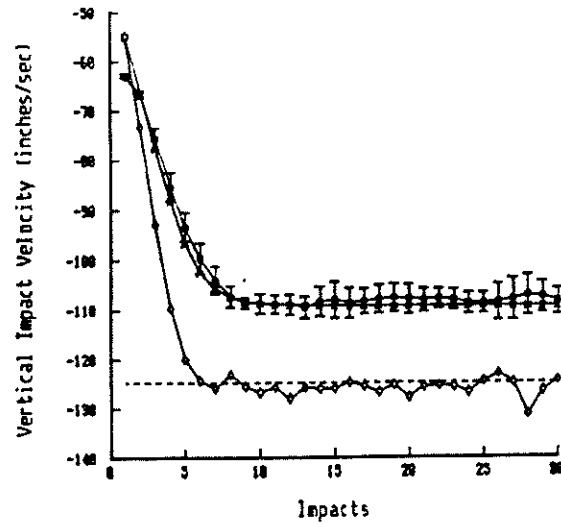
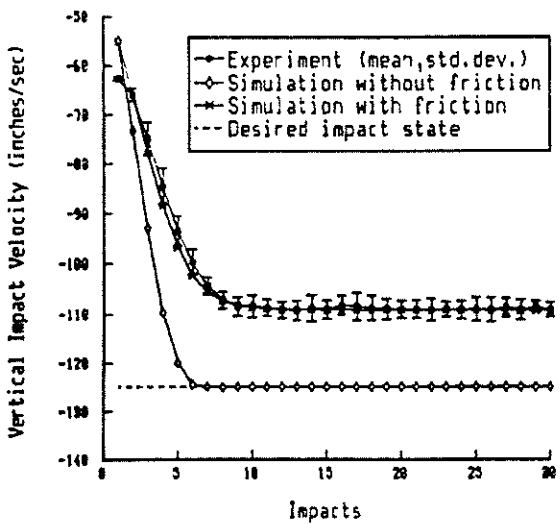
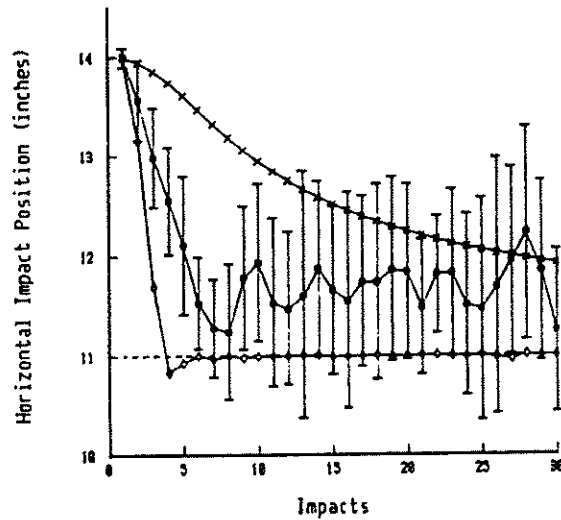
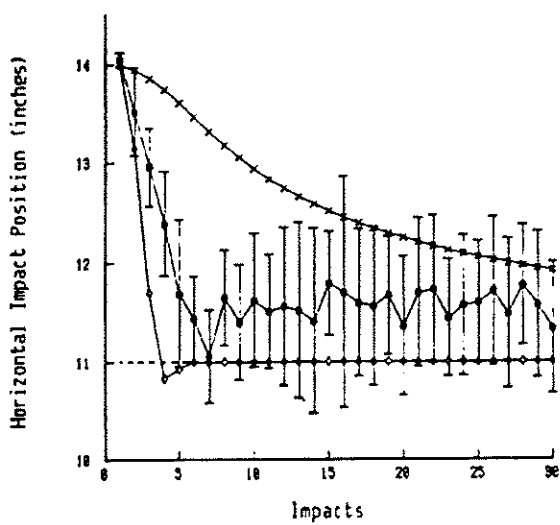
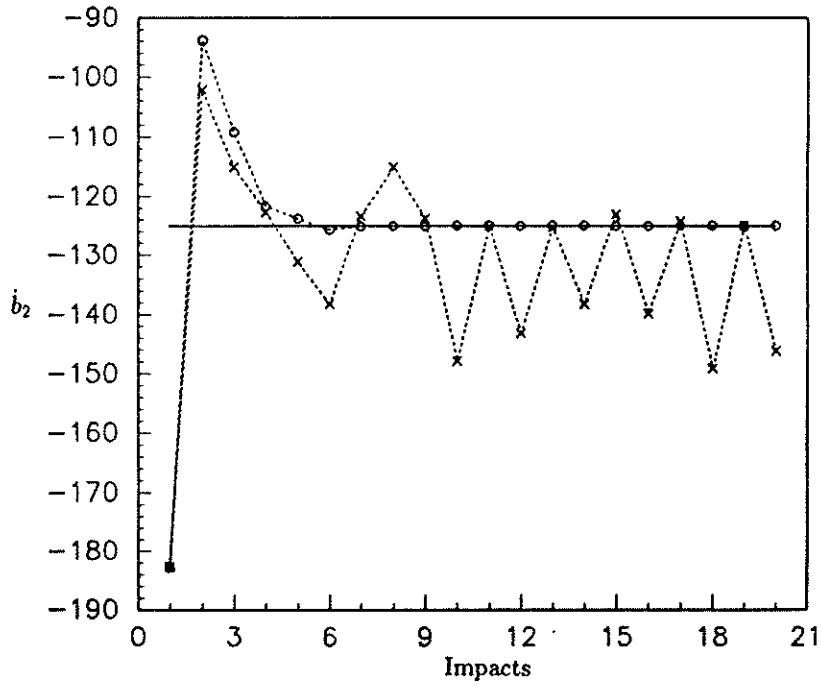
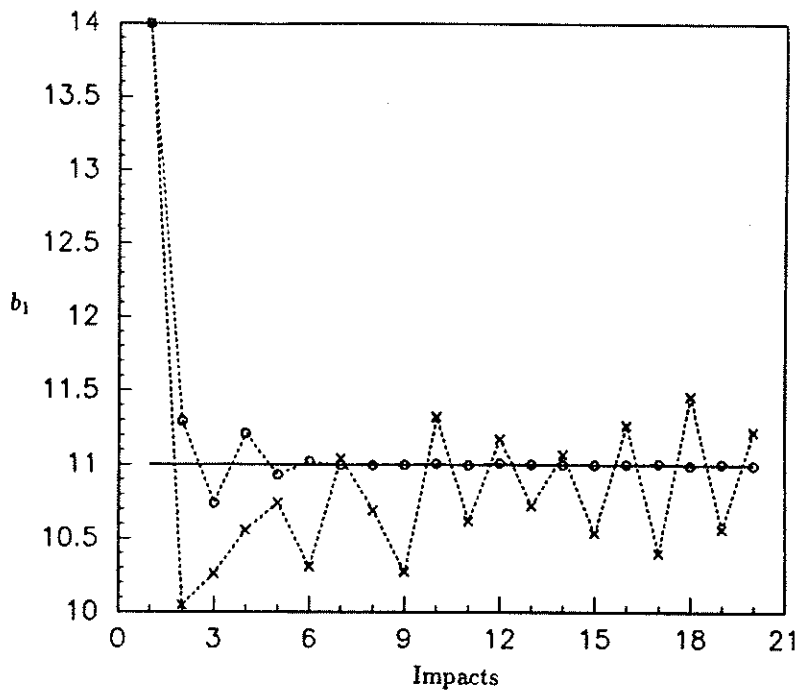


Figure 6: Fast local transient setting, Initial condition two.

Figure 7: Slow local transient setting, Initial condition two.



o Local Stability: Poles set to $[-0.1, 0, 0.1]$
 x Local Instability: Poles set to $[2.0, 0, 0.1]$

Figure 8: Simulation of a locally unstable, but globally stable system.

5 Analysis of the Mirror Algorithm: Controller Synthesis in Phase Space

In this section we provide a formal demonstration that the mirror algorithm succeeds in producing a correct vertical one-juggle: a locally asymptotically stable fixed point for any desired position on the task plane, \mathcal{T} . In Section 5.1 we demonstrate that the mirror law achieves a correct vertical one-juggle. In Section 5.2 we provide an example of how to “place poles” of the invariant submanifold restriction.

Before proceeding, some remarks about the usefulness of our analysis are called for. For the one degree of freedom system we can readily examine the nonlinear discrete impact map and derive the region of attraction [7]. However, for the planar system, the nonlinear analysis of the impact map is intractable. When confronted with such nonlinear systems, we first have to investigate the behavior of the linearized system around the fixed point in Section 5.1. This is an important first step and often provides a sufficiently accurate description of the system around a large enough region of the fixed point to be useful. In our case, even though we can command the poles of the linearized system using the results of the linear analysis, we cannot significantly affect the physically observable behavior of the system. This is because the influence of the algorithmic terms which affect the local behavior is so small that any changes are dwarfed by the error band of steady state performance in our specific implementation. Nonetheless, the linear analysis is valid and useful for investigating global behavior. For example, we started with a globally and locally stable one juggle algorithm. After detuning several of the gains k_i to destabilize the system, investigating the poles of the local linear system exhibited instable poles which confirms our expectation that the global behavior is reflected in the local linear characteristics.

5.1 Local Stabilization of an Invariant Submanifold

The local proof is divided in two parts; Section 5.1.1 establishes that the algorithm induces a three dimensional invariant submanifold of the effective closed loop environmental dynamics in Proposition 5.2. In Section 5.1.2 we provide relatively unrestrictive sufficient conditions under which the poles of the lower dimensional discrete control system — the restriction of (11) to this submanifold — may be placed by adjusting the mirror algorithm gains in Proposition 5.6. Finally, in Section 5.2 we choose gain settings of the mirror algorithm which meet the requirements of Proposition 5.6, and show how the pole locations of the data in Section 4.3 are obtained.

In Section 4.3 we introduced the mirror algorithm as an exercise in robot trajectory tracking where the reference trajectory is a function of the puck’s state. We now assume that the robot has achieved exact tracking via standard linear PD theory applied to (1), and interpret the consequences in terms of invariant surfaces in the puck-robot phase

space, $\mathcal{X} \triangleq \mathcal{W} \times \mathcal{R}$, introduced in Section 2.2.2. Of course, in practice, one obtains merely asymptotically exact tracking: we will suppose that the robot has “enough time” in between impacts to achieve a sufficiently close approximation to this asymptotic state at each successive impact.

5.1.1 The Induced Invariant Submanifold

We define a new surface, the “mirror surface,” $\mathcal{M} \subset \mathcal{X}$ to be the zero set, $\mathcal{M} \triangleq \tilde{\mu}^{-1}[0]$, of a new scalar valued function,

$$\tilde{\mu} : \mathcal{X} \rightarrow \mathbb{R} : (r, \dot{r}, w) \mapsto r - \mu(w) \quad (18)$$

where

$$\mu : \mathcal{W} \rightarrow \mathbb{R} : w \mapsto \kappa_1(w) \cdot \theta - \kappa_2(w)$$

θ is the “puck angle,” and κ_i are the gain functions detailed in (17). The robot implementation strategy is to control the robot’s state, $v = (r, \dot{r})$ so that the coupled robot-environment state of \mathcal{X} remains within the mirror set, \mathcal{M} , for all time. As described above, we now assume that this objective is attained. By studying the intersection of $\mathcal{M} \subset \mathcal{X}$ with the impact velocities $T_I \mathcal{C} \subset \mathcal{X}$, we may now determine the effective environmental control law that the robot realizes in so doing.

Lemma 5.1 *Suppose the puck-robot phase space trajectory, $x(t_j)$, remains in the mirror surface, $\mathcal{M} \subset \mathcal{X}$, defined in (18). Then the resulting impact schedule defines an environmental feedback law, $g : \mathcal{W} \rightarrow \mathcal{U}$ which realizes the fixed point condition of Proposition 3.1 for any $w^* \in \mathcal{T}$ if and only if the gain settings (17) satisfy*

$$\kappa_{10} = \frac{1 - \alpha}{1 + \alpha} - \gamma \frac{\kappa_{41} b_1^*}{\kappa_{42}^2 \dot{b}_2^*}$$

Proof: Assuming that the robot has achieved the mirror surface we have

$$(r, \dot{r}) = (\mu, \dot{\mu})$$

when $(b, r) \in \mathcal{I}$. Therefore the robot bar velocity at a point of impact is

$$u_2 = g_2(w) = \|b\|\dot{r} = \|b\|\dot{\mu} = \|b\| \cdot d\mu(w) \cdot n(w).$$

Evaluating this expression at $w^* = [b_1^*, 0, 0, \dot{b}_2^*]'$ with the above κ_{10} yields the desired robot control velocity at impact,

$$u_2^* = -\frac{1 - \alpha}{1 + \alpha} \dot{b}_2^*.$$

Using (8), the resulting velocity vector after impact is

$$\dot{b}^{*'} = \bar{C}\dot{b}^* + \bar{c}u_2^* = \begin{bmatrix} 0 \\ -\dot{b}_2^* \end{bmatrix}.$$

If the robot remains on \mathcal{M} , the next impact will occur again at w^* . Integrate the motion of the puck to compute the time of flight, as desired as,

$$u_1^* = -\frac{2}{\gamma}\dot{b}_2^*.$$

□

Proposition 5.2 *If the puck-robot phase trajectory, $x(t_j)$ remains in \mathcal{M} then the resulting impact sequence realizes an environmental feedback control law, $g : \mathcal{W} \rightarrow \mathcal{U}$ whose closed loop dynamics, f_g , in \mathcal{W} possesses an invariant co-dimension one set, $\mathcal{S} \subset \mathcal{W}$, that is*

$$f_g(\mathcal{S}) \subset \mathcal{S},$$

which is smooth in some open neighborhood of the fixed point, w^ .*

Proof: Observe that the intersection of the mirror surface, \mathcal{M} , with the tangents over the impact surface, $T_I\mathcal{C}$, is given by the cross product,

$$\mathcal{M} \cap T_I\mathcal{C} = \sigma^{-1}[0] \times \mathcal{R},$$

of the robot phase space, \mathcal{R} , with the zero set of the smooth map

$$\sigma : \mathcal{W} \rightarrow \mathbb{R} : w \mapsto \theta - \mu.$$

For if $x \in T_I\mathcal{C}$ then

$$x = ((b, r), (\dot{b}, \dot{r})) \text{ and } r = \theta(b),$$

while if $x \in \mathcal{M}$ then

$$x = ((b, r), (\dot{b}, \dot{r})) \text{ and } r = \mu(b, \dot{b}),$$

and it follows that $v = (r, \dot{r})$ is unconstrained, while $w = (b, \dot{b})$ satisfies

$$\theta(b) = \mu(b, \dot{b}).$$

The result now follows when we define $\mathcal{S} \triangleq \sigma^{-1}[0]$. For the differential,

$$\begin{aligned} d\sigma(w) &= d\theta - d\mu \\ &= \left[-\frac{b_2}{b_1^2 + b_2^2}, \frac{b_1}{b_1^2 + b_2^2}, 0, 0 \right] \end{aligned}$$

$$\begin{aligned}
& - \left[\frac{\kappa_1(w)b_2}{b_1^2 + b_2^2} + \kappa_{21}, \quad \kappa_{11}\gamma \operatorname{atan} \frac{b_2}{b_1} - \frac{\kappa_1(w)b_1}{b_1^2 + b_2^2}, \quad \kappa_{22}, \quad \kappa_{11}b_2 \operatorname{atan} \frac{b_2}{b_1} \right] \\
& - \frac{\kappa_{31} - 3 \cdot [(w - w^*)^T M(w - w^*)]^2}{[\kappa_{31} + [(w - w^*)^T M(w - w^*)]^2]^3} \cdot (w - w^*)^T M \\
& - \kappa_{41} \cdot \frac{\kappa_{42} - 3(b_2 - b_2^*)^2}{[\kappa_{42} + (b_2 - b_2^*)^2]^3} \cdot [0, 0, 0, 1]
\end{aligned}$$

is non-zero at any fixed point, $w^* \in \mathcal{T}$,

$$\begin{aligned}
d\sigma(w^*) &= d\theta(w^*) - d\mu(w^*) = \begin{bmatrix} 0, & -\frac{1}{b_1^*}, & 0, & 0 \end{bmatrix} - \begin{bmatrix} \kappa_{21}, & -\frac{\kappa_{10}}{b_1^*}, & \kappa_{22}, & \frac{\kappa_{41}}{\kappa_{42}^2} \end{bmatrix} \\
&= \begin{bmatrix} -\kappa_{21}, & \frac{2}{(1 + \alpha)b_1^*}, & -\kappa_{22}, & -\frac{\kappa_{41}}{\kappa_{42}^2} \end{bmatrix} \quad (19)
\end{aligned}$$

and by continuity, also non-zero in some open neighborhood, $\mathcal{O} \subset \mathcal{W}$. Thus $\mathcal{S} \cap \mathcal{O}$ is a smooth co-dimension one submanifold of \mathcal{W} according to the Implicit Function Theorem [16]. Moreover, under the assumption that the puck-robot phase remains in \mathcal{M} , all impact events must occur with puck phase $w \in \mathcal{S}$. Thus, whatever environmental control law, g is induced, it has the property that $w \in \mathcal{S}$ implies $f_g(w) \in \mathcal{S}$.

□

An immediate consequence of this result is that the puck dynamics may be studied in three dimensional space rather than $\mathcal{W} = \mathbb{R}^4$. For present purposes, since we are merely interested in asymptotic stability of the fixed point, it suffices to consider the linearized system on an open subset in \mathbb{R}^3 . Since w^* is a regular point of σ , it follows from the implicit function theorem that an open neighborhood of that point (in the induced topology of $\mathcal{S} \subset \mathcal{W}$) is parametrized by some smooth function, $p : \widehat{\mathcal{W}} \rightarrow \mathcal{S}$, where $\widehat{\mathcal{W}}$ is some open neighborhood around the origin of \mathbb{R}^3 . Let $p^{-1} : p(\widehat{\mathcal{W}}) \rightarrow \widehat{\mathcal{W}}$ denote the local coordinate chart corresponding to p , and $\hat{w}^* \triangleq p^{-1}(w^*)$ denote the local coordinate representation of the fixed point, w^* .

Corollary 5.3 *Assume that the puck-robot trajectory remains on the mirror surface. Define the arrays,*

$$P \triangleq [Dp](\hat{w}^*); \quad P^\dagger \triangleq [Dp^{-1}](w^*),$$

and recall from Proposition 3.2 that

$$A \triangleq [D_w f](w^*, u^*); \quad B \triangleq [D_u f](w^*, u^*); \quad K \triangleq [D_w g](w^*).$$

The induced discrete closed loop puck dynamics, f_g , on \mathcal{W} , achieves the vertical-one juggle if and only if the 3×3 array,

$$P^\dagger [A + BK] P,$$

has eigenvalues inside the unit circle.

Proof: According to the assumption, the puck trajectory w (discrete sequence of puck states at impact) remains in \mathcal{S} . Thus, a successful vertical one-juggle obtains if and only if the restriction of f_g to \mathcal{S} is locally asymptotically stable at w^* . The latter may be expressed in $\hat{\mathcal{W}}$ coordinates as

$$\hat{f}_g \triangleq p^{-1} \circ f_g \circ p,$$

hence, trajectories, w_j near w^* in \mathcal{S} are attracted to w^* if and only if

$$\left[D_{\hat{w}} \hat{f}_g \right] (\hat{w}^*)$$

has eigenvalues inside the unit circle. The desired result follows by applying the chain rule.

□

5.1.2 Pole Placement on the Invariant Submanifold

The linearized feedback gains at the fixed point, w^* , may now be obtained as follows. The second row of

$$\hat{K} = KP = \begin{bmatrix} \kappa_1^T \\ \kappa_2^T \end{bmatrix} P,$$

is immediately available.

Lemma 5.4 For $w^* \in \mathcal{T}$ the mirror algorithm induces the local linearized feedback gains

$$\hat{\kappa}_2 \triangleq P^T(w^*) \left[\begin{bmatrix} 0 & \dot{b}_2^* & 0 & -\gamma \\ 0 & 0 & 0 & 0 \\ b_1^* & 0 & 0 & 0 \\ 0 & b_1^* & 0 & 0 \end{bmatrix} d\mu(w^*)^T + \|b_1^*\| \cdot [D_w^2 \mu](w^*) \begin{bmatrix} 0 \\ \dot{b}_2^* \\ 0 \\ -\gamma \end{bmatrix} \right].$$

Proof: Recall that the velocity of the robot bar at point of impact with the puck is given by

$$u_2 = \|w_1\| \dot{r}$$

Since the impact must occur on \mathcal{S} , we have, $\dot{r} = \dot{\mu}$, hence,

$$u_2 = g_2(w) = \|w_1\| \cdot d\mu(w) \cdot n(w),$$

where n is the puck dynamical law (2). The result obtains from elementary calculus, since

$$k_2 \triangleq dg_2(w^*)^T$$

and thus

$$\begin{aligned}
\hat{k}_2 &= P^T k_2 \\
&= P^T \left(d\mu \cdot n \frac{1}{\|w_1^*\|} \begin{bmatrix} w_1^* \\ 0 \end{bmatrix} + \|w_1^*\| \cdot \begin{bmatrix} 0 & 0 \\ I & 0 \end{bmatrix} d\mu^T \right. \\
&\quad \left. + \|w_1^*\| \cdot [D_w^2 \mu](w^*) n \right) \\
&= P^T \left(\frac{1}{\|w_1^*\|} \begin{bmatrix} w_1^* w_2^{*T} & w_1^* a^T \\ w_1^{*T} w_1^* \cdot I & 0 \end{bmatrix} d\mu^T + \|w_1^*\| \cdot [D_w^2 \mu](w^*) \begin{bmatrix} w_2^* \\ a \end{bmatrix} \right).
\end{aligned}$$

Substituting

$$w^* = \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \begin{bmatrix} b_1^* \\ 0 \\ 0 \\ b_2^* \end{bmatrix}$$

yields the desired result.

□

Since $g_1(w)$ may be impossible to obtain in analytical closed form (depending upon the nature of μ), we must compute its jacobian implicitly.

Lemma 5.5 Denote the canonical basis of \mathbb{R}^2 by

$$\{e_1, e_2\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\},$$

and

$$s^T \triangleq d\sigma(w^*).$$

If

$$s^T B e_1 \neq 0,$$

then the mirror algorithm induces the local closed loop feedback gains in the restricted discrete dynamical system, \hat{f}_g , given by

$$\hat{K} = \begin{bmatrix} \hat{k}_1^T \\ \hat{k}_2^T \end{bmatrix} = -\frac{1}{s^T B e_1} \left(\begin{bmatrix} s^T A P \\ 0 \end{bmatrix} + \begin{bmatrix} s^T B e_2 \\ -s^T B e_1 \end{bmatrix} \hat{k}_2^T \right),$$

where \hat{k}_2 is given in Lemma 5.4.

Proof: All impact states w under the mirror law (17) will occur on \mathcal{S} which is parametrized by $w = p(\hat{w})$ and thus $\sigma \circ p \equiv 0$. As \mathcal{S} is an invariant submanifold of the closed loop environmental dynamics,

$$\sigma \circ f_g \circ p \equiv 0,$$

we have

$$\begin{aligned} 0 &= [D_w(\sigma \circ f_g \circ p)](\hat{w}^*) \\ &= [D_w\sigma](w^*)(A + BK)[D_{\hat{w}}p](\hat{w}^*) \\ &= s^T(A + Be_2k_2^T + Be_1k_1^T)P, \end{aligned}$$

Under the hypothesis, it follows that

$$\hat{k}_1^T = \kappa_1^T P = -\frac{1}{s^T B e_1} (s^T A P + s^T B e_2 \hat{k}_2^T),$$

and the result follows after substitution for \hat{k}_1 and rearrangement of terms in \hat{K} .

□

Proposition 5.6 *Let $\|w_1^*\| \neq 0$ and $s^T B e_1 \neq 0$. Define the arrays*

$$\bar{A} \triangleq \left(I - \frac{B e_1 s^T}{s^T B e_1} \right) A; \quad l \triangleq \begin{bmatrix} -\frac{s^T B e_2}{s^T B e_1} \\ 1 \end{bmatrix}.$$

If $(P^\dagger \bar{A} P, P^\dagger B l)$ is a completely controllable pair then the local transient behavior of \hat{f}_g may be arbitrarily determined by proper choice of the mirror algorithm gains, (17).

Proof: Using the results of Lemma 5.4 and Lemma 5.5 it suffices to demonstrate that \hat{k}_2 may be arbitrarily specified without changing s, P, P^\dagger , all of which also depend upon μ . But notice that the latter are determined by $\mu^* \triangleq \mu(w^*)$ and $d\mu^* \triangleq d\mu(w^*)$, while the former depends upon its hessian, $\bar{M} \triangleq [D_w^2 \mu](w^*)$, as well:

$$\hat{k}_2 = P^\dagger (h(\mu^*, d\mu^*) + \|w_1^*\| \cdot \bar{M} n(w^*)). \quad (20)$$

Taking the hessian of (17) we obtain

$$\bar{M} = M_1 + \frac{1}{k_{31}^2} M, \quad (21)$$

where $M_1 \triangleq -[D_w^2 k_1](w^*)$, $M_1^T = M_1$. Notice that M is an arbitrary symmetric matrix introduced in (17) — which does not appear in μ^* or $d\mu^*$. To achieve any desired value, \hat{k}_2^* , we need merely solve the linear equations

$$\frac{1}{\|w_1^*\|} (\hat{k}_2^* - P^\dagger h(\mu^*, d\mu^*)) = P^\dagger \bar{M} n(w^*), \quad (22)$$

for the symmetric matrix, $M \in \text{sym}(4)$. Denote the left hand side of this equation by the symbol \bar{h} , an arbitrary $\mathbb{R}^{3 \times 1}$ array. Now, using the kronecker product and stack notation [2], this system of equations is equivalent to

$$\bar{h} = [n(w^*)^T \otimes P^\dagger] \bar{M}^s,$$

where \bar{M}^s — an array in $\mathbb{R}^{16 \times 1}$ — is in the image of $\Pi_{\text{sym}(4)}$ — the linear isomorphism from \mathbb{R}^{10} to $\text{sym}(4)$ — the symmetric matrices of $\mathbb{R}^{4 \times 4}$. Thus, the equation may be rewritten as

$$\bar{h} = \left[n(w^*)^T \otimes P^\dagger \right] \Pi_{\text{sym}(4)} \tilde{m},$$

$\tilde{m} \in \mathbb{R}^{10}$. With $n(w^*) = [0 \ \hat{k}_2 \ 0 \ -\gamma]^T$ the expression $(n^T \otimes P^\dagger)$ has no rows in $\text{skew}(4)$. Thus the array

$$\left[n(w^*)^T \otimes P^\dagger \right] \Pi_{\text{sym}(4)}$$

has rank 3. For, $\text{skew}(4)$ — the set skew symmetric matrices on \mathbb{R}^4 — is the orthogonal complement of $\text{sym}(4)$ (both considered as linear subspaces of $\mathbb{R}^{4 \times 4}$). Thus, any \hat{k}_2 may be achieved by proper choice of M without changing s, P, P^\dagger .

According to Corollary 5.3, the local behavior of \hat{f}_g is determined by $P^\dagger AP + P^\dagger B \hat{K}$. Substituting for \hat{K} as given in Lemma 5.5 yields

$$P^\dagger AP + P^\dagger B \hat{K} = P^\dagger \bar{A} P + P^\dagger B l \hat{k}_2^T.$$

Thus, since \hat{k}_2 may be freely assigned, if $(P^\dagger \bar{A} P, P^\dagger B l)$ is completely controllable, it follows that the poles of the jacobian of \hat{f}_g may be placed arbitrarily.

□

5.2 Example: Choosing the Pole Locations for a Particular Run

In Section 4.3 we presented data for two different runs which we distinguished by the labels “slow” and “fast” local transient settings. In this section we use the theory developed above to show how to place poles of the local reduced dimensional discrete control system, and, in particular report the gain settings required to obtain the pole locations corresponding to those data.

5.2.1 Selection of Gains for Reduced Dimensional Controllability

In order to examine the controllability of our system according to Proposition 5.6 we must ensure that $(P^T \bar{A} P, P^T B l)$ is a completely controllable pair.

While the local analysis of section 5.1 gives us detailed instructions on how to choose the symmetric matrix M for arbitrary poleplacement, the rest of the gains — except κ_{10} — have to be determined experimentally. They were selected as follows:

$$\kappa_{11} = 3 \cdot 10^{-5}; \quad \kappa_{21} = 7 \cdot 10^{-3}; \quad \kappa_{22} = 5 \cdot 10^{-3}; \quad \kappa_{31} = 100; \quad \kappa_{41} = 0.01; \quad \kappa_{42} = 10. \quad (23)$$

With an inclination of the juggling plane of 30 degrees off the vertical, the effective gravitational constant is $\gamma = 334.36 \frac{in}{sec^2}$ and the dry friction coefficient is $\mu_{fric} = 0.16$. The desired one juggle was chosen to be

$$w^* = [11in, 0, 0, -125 \frac{in}{sec^2}]^T$$

The coefficient of restitution was experimentally found to be $\alpha = 0.7$. Now, from Lemma 5.1 we determine $\kappa_{10} = 0.1765$.

A and B are as given in Section 3.2,

$$A = \begin{bmatrix} 1 & -\frac{4\dot{b}_2^{*2}}{\gamma \dot{b}_1^*} & -\frac{2\dot{b}_2^*}{\gamma} & 0 \\ 0 & 1 & 0 & \frac{2}{\gamma} \alpha \dot{b}_2^* \\ 0 & \frac{2\dot{b}_2^*}{\dot{b}_1^*} & 1 & 0 \\ 0 & 0 & 0 & -\alpha \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 \\ \dot{b}_2^* & -\frac{2}{\gamma}(1+\alpha)\dot{b}_2^* \\ 0 & 0 \\ -\gamma & 1+\alpha \end{bmatrix}.$$

With w^* as given as above the first and with $s^T \equiv d\sigma(w^*)$ as given in (19), the second condition in Proposition 5.6 is satisfied.

In order to construct a matrix P notice that the condition

$$\sigma \circ p \equiv 0,$$

holds for all $\hat{w} = p^{-1}(w)$ by definition of the impact surface \mathcal{S} . Therefore, on \mathcal{S} , it must also be that

$$d\sigma Dp = s^T P = 0.$$

We now construct a matrix P, whose transpose has the kernel s^T , satisfying the previous equation:

$$P \triangleq [(J \otimes I)s; (L \otimes J)s; (K \otimes J)s] \cdot \frac{1}{\|s\|},$$

where

$$I \triangleq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad J \triangleq \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}; \quad K \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad L \triangleq \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

This particular choice of P has the additional property that $P^T P = I_{3 \times 3}$, thus we have

$$P^\dagger = P^T.$$

Finally we can compute the controllability matrix of the pair $(P^\dagger \bar{A} P, P^\dagger B)$. For the above settings of all parameters, variables and system constants its rank is 3 and its condition number is 0.001.

5.2.2 Selection of Gains for Reduced Dimensional Pole Placement

For the experimental and simulation data presented in Section 4.3 we selected two pole locations: a setting for slow local transients,

$$\hat{\Lambda}_{slow} \triangleq \{0.9, 0.9 \pm 0.1j\}$$

and a setting for fast local transients,

$$\hat{\Lambda}_{fast} \triangleq \{0, \pm 0.1j\}.$$

Given any desired pole locations for the linearized system, a standard numerical pole placement routine will now give us the desired \hat{k}_2^* . Now we use (20) to compute the necessary hessian \bar{M} and (21) to obtain the gain matrix M for (17).

By choosing

$$\bar{M} = \begin{bmatrix} 0 & \bar{m}_1 & 0 & 0 \\ \bar{m}_1 & \bar{m}_2 & 0 & 0 \\ 0 & 0 & 0 & \bar{m}_3 \\ 0 & 0 & \bar{m}_3 & \bar{m}_4 \end{bmatrix},$$

we can rewrite the right side of (22)

$$P^\dagger \bar{M} n(w^*) = P^\dagger \bar{M} \begin{bmatrix} w_2^* \\ a \end{bmatrix} = P^\dagger \begin{bmatrix} b_2^* & 0 & 0 & 0 \\ 0 & b_2^* & 0 & 0 \\ 0 & 0 & -\gamma & 0 \\ 0 & 0 & 0 & -\gamma \end{bmatrix} \begin{bmatrix} \bar{m}_1 \\ \bar{m}_2 \\ \bar{m}_3 \\ \bar{m}_4 \end{bmatrix} \triangleq Q \bar{m}.$$

As $\text{rank}(Q) = 3$, we can find a solution for \bar{m} :

$$\bar{m} = Q^T(QQ^T)^{-1} \frac{1}{\|w_1^*\|} (\hat{k}_2^* - P^\dagger h).$$

With \bar{M} available, the yet unassigned gain matrix M is determined by (21):

$$M_{slow} = \begin{bmatrix} 0 & 559.39 & 0 & 0 \\ 559.39 & 20.55 & 0 & 3.4 \\ 0 & 0 & 0 & 68.65 \\ 0 & 3.4 & 68.65 & 2.98 \end{bmatrix}, \quad M_{fast} = \begin{bmatrix} 0 & 25.95 & 0 & 0 \\ 25.95 & -15.55 & 0 & 3.4 \\ 0 & 0 & 0 & 1.2 \\ 0 & 3.4 & 1.2 & -1.32 \end{bmatrix}.$$

Now we have completely specified a "mirror algorithm" which is guaranteed to implement a vertical one-juggle according to our definition.

Conclusion

We have shown how the geometry of the contact configurations, \mathcal{I} , 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 under-constrained 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 [3]. 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] E. Aboaf, S. Drucker, and C. Atkeson. Task-level robot learning: juggling a tennis ball more accurately. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1290–1295, Scottsdale, AZ, May 1989.
- [2] R. Bellman. *Introduction to Matrix Analysis*. McGraw Hill, New York, 1965.
- [3] 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.
- [4] M. Bühler and D. E. Koditschek. A prelude to juggling. In *26th IEEE Conference on Decision and Control*, pages (paper available from authors — not in proceedings), Los Angeles, CA, Dec 1987.
- [5] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. A family of robot control strategies for intermittent dynamical environments. In *IEEE International Conference on Robotics and Automation*, pages 1296–1301, Scottsdale, AZ, May 1989.
- [6] 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*, pages 91–97, Tokyo, Japan, Oct 1988.
- [7] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. Planning and Control of Robotic Juggling Tasks. In H. Miura, editor, *Fifth International Symposium on Robotics Research*, page (to appear), MIT Press, Tokyo, Japan, 1989.
- [8] P. Collet and J. P. Eckmann. *Iterated Maps on the Interval as Dynamical Systems*. Birkhäuser, Boston, 1980.
- [9] D. E. Koditschek and M. Bühler. Analysis of a simplified hopping robot. *The International Journal of Robotics Research*, (to appear).
- [10] 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.
- [11] T. McGeer. *Passive Bipedal Running*. Technical Report IS-TR-89-02, Simon Fraser University, Centre for Systems Science, Apr 1989.
- [12] T. McGeer. Powered flight, child's play, silly wheels and walking machines. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1592–1597, May 1989.
- [13] T. McGeer. *Stability and Control of Two-Dimensional Biped Walking*. Technical Report IS-TR-88-01, Simon Fraser University, Centre for Systems Science, Sep 1988.
- [14] Marc H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.

- [15] J. F. Schaefer and R. H. Cannon Jr. On the control of unstable mechanical systems. In *IFAC*, pages 6c.1–6c.13, London, Jun 1966.
- [16] I. M. Singer and J. A. Thorpe. *Lecture Notes on Elementary Topology and Geometry*. Springer-Verlag, NY, 1967.
- [17] J. L. Synge and B. A. Griffith. *Principles of Mechanics*. McGraw Hill, London, 1959.
- [18] Yu Wang. *Dynamic Analysis and Simulation of Mechanical Systems with Intermittent Constraints*. PhD thesis, Carnegie-Mellon University, 1989.

