May 1993

Further progress in robot juggling: the spatial two-juggle

Alfred A. Rizzi
University of Michigan

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Further progress in robot juggling: the spatial two-juggle

Abstract
We report on our recently achieved spatial two-juggle: the ability to bat two freely falling balls into stable periodic vertical trajectories with a single three degree of freedom robot arm using a real-time stereo camera system for sensory input. After a brief review of the previously reported one-juggle, we describe our initial approach to the two-juggle planning and control problem. We have developed a number of important refinements to our initial strategy in the course of getting the system to work, and these are reported in some detail. The paper concludes with a discussion of some data from typical two-juggle runs in the laboratory.

Comments

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

Further Progress in Robot Juggling: The Spatial Two-Juggle

A. A. Rizzi *and D. E. Koditschek †

Artificial Intelligence Laboratory
University of Michigan
Department of Electrical Engineering and Computer Science

Abstract

We report on our recently achieved spatial two-juggle: the ability to bat two freely falling balls into stable periodic vertical trajectories with a single three degree of freedom robot arm using a real-time stereo camera system for sensory input. After a brief review of the previously reported one-juggle, we describe our initial approach to the two-juggle planning and control problem. We have developed a number of important refinements to our initial strategy in the course of getting the system to work, and these are reported in some detail. The paper concludes with a discussion of some data from typical two-juggle runs in the laboratory.

1 Introduction

We have recently reached a long-targeted milestone in our laboratory: the ability to simultaneously juggle two freely falling balls with a single visually servoed robot arm. Our experiments in robot juggling began five years ago with the planar one-juggle — the ability of a one degree of freedom revolute bar to bat a single puck falling on a frictionless inclined plane into a stable periodic orbit with specified apex [3]. The planning and control strategy developed in that effort turned out to extend quite naturally to the planar two-juggle — simultaneously batting two pucks falling on the plane into two specified periodic orbits through properly scheduled impacts with the actuated revolute bar [2]. Last year we reported our success in generalizing the same strategy to achieve a spatial one-juggle — batting a ball falling freely in space into a specified vertical periodic orbit — with the system depicted in Figure 1. The same broad set of ideas continues to guide the present work. Yet several important refinements have been introduced in order to achieve this latest capability.

The juggling behaviors we explore are exemplars from a more general range of dynamically dexterous capabilities that autonomous robots will surely require in unstructured environments. Controlled collisions have been understood to play a key role in robotic manipulation since the beginnings of the field [5]. Indeed, suggestive experiments with planar "catching" based upon extensions of the algorithms reported here [2] offer some hint of how to handle the mechanics involved in the passage from our ballistic mode to the quasi-static mode of manipulation exemplified in the work of Mason and colleagues [7]. The connection to periodic dynamical tasks is even more clear. Preliminary analysis suggests that the same mechanism responsible for the vertical component of Raibert's hoppers accounts for the stability of our robots' behaviors as well [6]. Nonlinear oscillators have received significant attention as central pattern generators in biological locomotory systems [4], and our approach to "gait regulation" may offer a link to the mechanical sources of such oscillators.

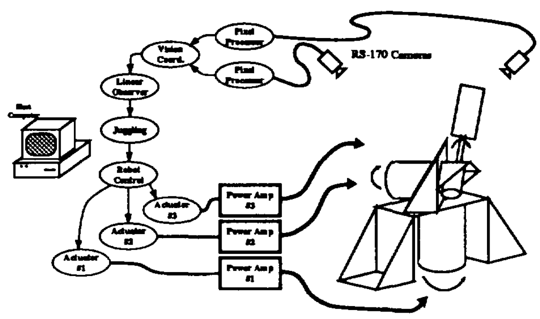

Figure 1: The Yale spatial juggling system.

2 Juggling Algorithms

This section describes how the juggling control methodology originally introduced for the planar system [3, 2] has been extended to the present apparatus, and in particular how the experimental work presented in

*Supported in part by IBM through a Manufacturing Graduate Fellowship and in part by the National Science Foundation under grant IRI-9123266.

†Supported by the National Science Foundation in part through a Presidential Young Investigator Award and in part under grant IRI-9123266.

1050-4729/93 $3.00 © 1993 IEEE

[11] has been continued to include implementation of a spatial two-juggle task. Skipping over the analysis of the prior planar work [1], we pass immediately to a "commented" presentation of the mirror law, a nonlinear function from the phase space of the body to the phase space of the robot that generates a reference trajectory as the ongoing time history of the ball's position and velocity is fed through it. This mirror law, is used in conjunction with a collection of analytic functions that intuitively implement our notions of "if-then-else" within a "geometric programming framework," to develop a juggling strategy for keeping two balls aloft simultaneously.

It must be emphasized that the functions we present here comprise at once a mathematical description of our algorithm and its actual implementation. Implementing "geometric programs" of this type amounts to merely placing the particular transformation law — in the present case, (3) or (7) — in the juggling block of the data flow path depicted in the left side of Figure 1. One immediate practical benefit of this arrangement is the availability of very powerful high level development environments in the form of commercial symbolic manipulation packages. In practice, we craft these functions in Mathematica on a SPARCstation and use the automatically generated C code on the target controller.

2.1 Review of the Working One-Juggle: The Mirror Law

A detailed development of the one-juggle control strategy can be found in [1, 10]. Briefly, the "mirror law," is a map $m : T\mathcal{B} \to \mathcal{Q}$ (from the phase space of a ball to the configuration space of the robot), that determines the robot's reference trajectory as $q_d(t) = m(w(t))$.

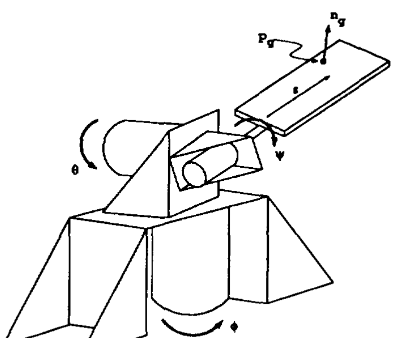

Figure 2: The Bühgler arm (left) and it's kinematics (right).

The function, m is defined as follows. Using (6) of [11], define the the joint space position of the ball

$$\begin{bmatrix} \phi_b \\ \theta_b \\ \psi_b \\ s_b \end{bmatrix} \triangleq p^{-1}(b), \tag{1}$$

where p^{-1} is the inverse kinematic map (including the paddle's length s that provides an effective fourth degree of freedom) for our machine, shown in Figure 2. We now seek to express formulaically a robot strategy that causes the paddle to respond to the motions of the ball in four ways:

(i) $q_{d1} = \phi_b$ causes the paddle to track under the ball at all times.

(ii) The paddle "mirrors" the vertical motion of the ball through the action of θ_b on q_{d2} as expressed by the original planar mirror law [3].

(iii) Radial motion of the ball causes the paddle to raise and lower, resulting in the normal being adjusted to correct for radial deviation in the ball position.

(iv) Lateral motion of the ball causes the paddle to roll, again adjusting the normal so as to correct for lateral position errors.

To this end, define the ball's *vertical energy* and *radial distance* as

$$\eta \triangleq \gamma b_z + \frac{1}{2}\dot{b}_z^2 \quad \text{and,} \quad \rho_b \triangleq \sin(\theta_b) s_b \tag{2}$$

respectively. The complete mirror law combines these two measures with a set point description ($\bar{\eta}$, $\bar{\rho}$, and $\bar{\phi}$) to form the function

$$m(w) \triangleq \begin{bmatrix} \overbrace{\phi_b}^{\text{(i)}} \\ \underbrace{-\frac{\pi}{2} - (\kappa_0 + \kappa_1(\eta - \bar{\eta}))\left(\theta_b + \frac{\pi}{2}\right)}_{\text{(ii)}} \quad + \\ \underbrace{\kappa_{00}(\rho_b - \bar{\rho}_b) + \kappa_{01}\dot{\rho}_b}_{\text{(iii)}} \\ \underbrace{\kappa_{10}(\phi_b - \bar{\phi}_b) + \kappa_{11}\dot{\phi}_b}_{\text{(iv)}} \end{bmatrix} \tag{3}$$

2.2 Planning the Two-Juggle: Phase Regulation and Urgency

A two-juggle task requires that the robot perform two simultaneous one-juggles with two independent balls separated in both space and time. Separation in space avoids ball-ball collisions (not currently part of the environmental model) and temporal separation (the two balls should not fall simultaneously) is necessary to ensure that the machine is capable of striking one ball and moving into position under the second, prior to the first falling to the floor. The juggling algorithm must be able to control the phase relationship between the two balls in addition to the new variables associated with the position and energy of the additional ball.

To accomplish this, we follow the ideas of [2] and introduce a new variable, ***ball phase***.

$$\epsilon(w) \triangleq -\frac{\dot{b}_3}{\sqrt{2\eta}}, \qquad (4)$$

which evaluates to 1 immediately prior to impact, -1 immediately after impact, and 0 at the apex of the ball's flight. This function measures the ball's progress through its repetitive sequence of fall-to-impact-to-rise-to-apex events in a manner that is independent of its total energy. A symmetric ***phase error*** can then be constructed based on the desired phase relationship between the two balls,

$$e_{ph}(w_0, w_1) = [\epsilon(w_0) - \epsilon(w_1)]^2 - 1. \qquad (5)$$

The one-juggle mirror law is then augmented by *phase error* to form

$$m_i(w_a, w_b) \triangleq m(w_q) + \begin{bmatrix} 0 \\ (\kappa_2 e_{ph}(w_a, w_b))\left(\theta_b + \frac{\pi}{2}\right) \\ 0 \end{bmatrix}. \qquad (6)$$

This new relationship between the ball's state and the robot configuration is essentially equivalent to (3), except that the expression for q_{d2} now includes a term based on e_{ph}. This term is responsible for maintaining "phase separation" between the two balls. Its overall effect causes the robot to strike a ball "harder" when it is following too closely behind the other ball. Similarly, it will strike a ball "more gently" should the other be too close behind it. Both of these behaviors result in increasing or decreasing, respectively, the ball's time of flight, thereby correcting the phase relationship, e_{ph}. Of course proper adjustment of the parameter κ_2 is crucial to overall system stability.

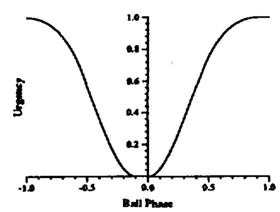

Figure 3: The mapping, σ, from ball phase to "urgency".

Individual mirror laws for the two balls, are then combined to form the overall two-juggle law by the use of a scalar valued analytic switch $s \in [0, 1]$,

$$\begin{aligned} m_{II}(w_0, w_1) &= s(w_0, w_1) m_0(w_0, w_1) + \\ &\quad (1 - s(w_0, w_1))\, m_1(w_1, w_0). \end{aligned} \qquad (7)$$

The function s encodes the mixture between the need to juggle ball 0 (follow m_0) or ball 1 (follow m_1), and is itself constructed from individual "urgency" functions (σ) for each ball by

$$s = \frac{1 - \sigma(w_1)}{2 - \sigma(w_0) - \sigma(w_1)}. \qquad (8)$$

Where the "urgencies" are produced by a map from the phase of a ball to the unit interval as show in Figure 3. The motivation for this implementation is as follows. ϵ varies smoothly from -1 immediately after impact, to 0 at the balls apex, to 1 immediately prior to impact. σ then describes the ***urgency*** of the ball (being near 1 when the ball is near impact, and 0 as it rises to its apex). Finally s combines these two ***urgencies*** by smoothly mapping the unit box onto $[0, 1]$ so $s = 0$ when $\sigma_1 = 1$, and $s = 1$ when $\sigma_0 = 1$.

3 Refinements in Implementation

The foregoing strategy represents a more or less straightforward generalization of the ideas developed in [2]. In order to achieve the spatial two-juggle in our laboratory, however, we have found in necessary to introduce a number of important refinements.

3.1 Smoothness of Reference Trajectories

We use a nonlinear inverse dynamics based robot controller [12] to track the reference trajectory output by the mirror law (7). Such controllers require that the reference trajectory be twice differentiable. Now (7) is an analytic function, and its input is the output of a mechanical system (the ball's position and velocity). However the velocity of the ball is necessarily discontinuous at the impact events. Consequently, the velocities we command the robot to attain are discontinuous as well. One might plausible hope that this might not be a problem in practice, as the discontinuity only occurs immediately after the machine has struck the ball (at a juncture when it does not matter what the arm does). Unfortunately the "settling time[1]" of our machine is on the order of 0.3 to 0.5 seconds. This is close enough to the expected temporal ball separation (time between impacts) at equilibrium (0.5 sec) to cause difficulties. The potential for failure during transients is much worse.

Under the scheme described above we observed that there were significant difficulties maintaining regulation of a one-juggle while introducing the second ball. On the rare occasions that we succeeded in releasing

[1]Of course since the dynamics of the robot are nonlinear there is no formally defined settling time for such a system. We use the term loosely to mean the time typically necessary for the robot and controller to recover from a transient in the reference trajectory.

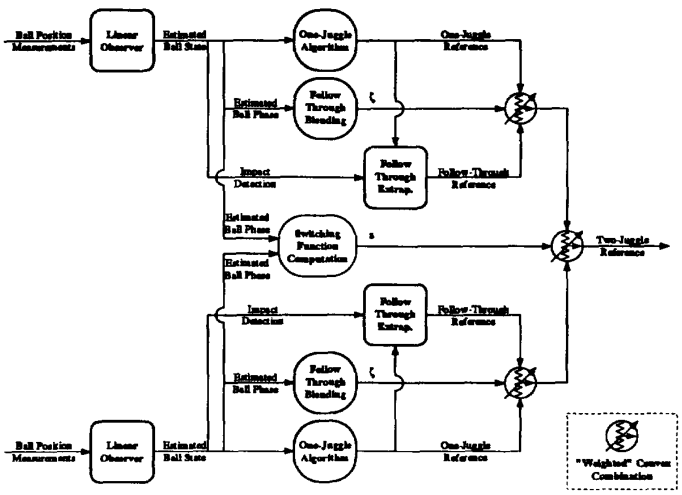

Figure 4: Block diagram of two-juggle algorithm with follow-through.

the second ball, failure would shortly follow due to wild hits, or simple misses. From the analysis of data from these failed attempts it became clear that one of the major causes was the inability of the robot to track the reference trajectories. In particular, the machine could not recover from a post-impact transient in time to reliably strike the next ball. The most natural solution was to correct the reference trajectory, and make it "trackable". Towards this end we have added what we have come to call a follow-through to the juggling algorithm.

3.2 Follow Through

The follow-through, whose implementation is shown in block diagram form in Figure 4, consists of two parts, a reference trajectory generator, and a mixing function. The follow through reference trajectory is simply the extrapolation of the commanded reference trajectory immediately prior to the impact. That is to say we capture the commanded state of the robot immediately before the impact, then integrate it forward in time with no acceleration to generate the follow-through reference. This trajectory is then mixed with the true juggling trajectory by taking a convex combination of the two trajectories based on the phase of the ball in question. More specifically the phase is passed through the function show in Figure 5 to produce ζ which is used to combine the two references as follows,

$$r = \zeta(w) r_{\text{follow}} + (1 - \zeta(w))\, r_{\text{juggle}}.$$

By properly choosing the function which relates ball phase to ζ it becomes possible to produce reference trajectories which remain twice differentiable across the impacts, yet still exactly track the trajectories given by (3) as the ball approaches impact. In particular for

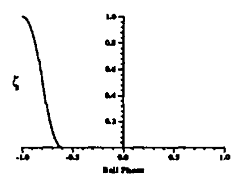

Figure 5: Splined follow-through mixing function, ζ.

these two properties to be met ζ must be 1 when the balls phase is at -1 (immediately after impact) and must fall to 0 before the phase reaches $+1$ (immediately before impact). We use, for ζ, a one parameter family of splined functions, consisting of two quadratic pieces which take ζ from 1 to 0 as the phases move from -1 to a chosen value and a constant piece which remains at 0 across the remainder of the domain.

3.3 Sensing Issues

The primary sensor for our spatial juggler is a stereo vision system[2]. As might be imagined, the task of keeping track of two flying balls is considerably harder than tracking a single one. The problems and our solutions are described at length in a companion paper [9] and are merely sketched here. Obviously there is the classic "tracking" problem of assigning measurements to tracks. Given of the real-time nature of the juggler we choose to produce ball locations from image data by recourse to "brute simple" computations of low-order moments (zeroth, first, and second) over small subwindows in the image (we currently use 1200

[2]We also use an audio sensor for precise temporal measurements of ball impacts.

pixels per window). Coupling the window placement of this tracker with a dynamical observer has proven extremely effective in correlating tracks to data, as the simple Newtonian model for the free falling ball is sufficiently accurate for reliable prediction. However the problem of real-time image interpretation becomes more complicated as the balls pass arbitrarily closely together in any particular image. The simplicity of our low-level image processing algorithm incurs significant risk of catastrophic failure, resulting either in the confusion of the two balls (both observer "tracking" one ball) or the simple loss of one or both. In order to avoid these pitfalls we have chosen to selectively ignore images where interpretation would become too computationally expensive to be undertaken in our real-time environment. The analysis of this decision algorithm comprises the main focus of [8].

4 Present Status

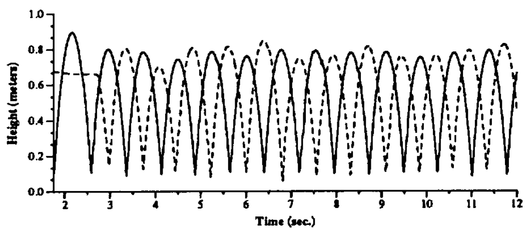

Figure 6: Two-Juggle ball trajectory: Height vs. Time.

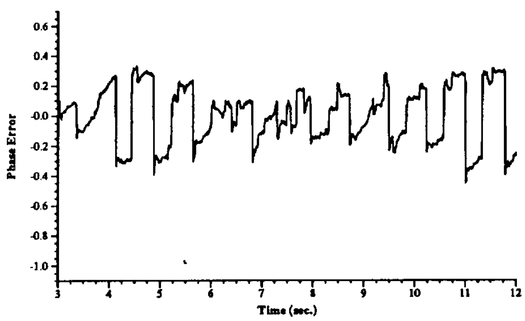

Figure 7: Two-Juggle ball trajectory: Phase Error vs. Time.

What follows here is a presentation of our most recent success at implementing the spatial two-juggle task. Figure 6 depicts the vertical position of two balls during a typical juggling run. As can be seen the machine succeeds in regulating the heights of the constituent one juggles to within roughly 10 cm of the target height. In this particular experiment the initial drop of the second ball was well timed (nearly perfectly out of phase with the original ball), and thus we see no significant effort on the part of the machine to correct the phase. This is depicted in the plot of phase error

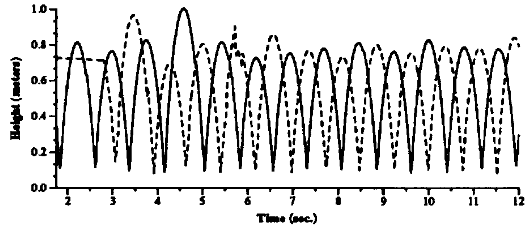

Figure 8: Two-Juggle ball trajectory: Height vs. Time.

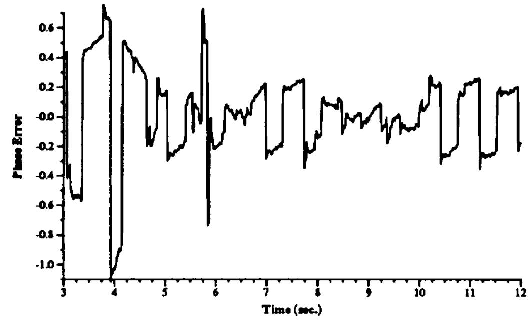

Figure 9: Two-Juggle ball trajectory: Phase Error vs. Time.

shown in Figure 7. In contrast Figures 8 and 9 show the same variables for a different initial condition, a rather poor drop. Here the effect of phase regulation is clearly visible in the trajectories of both balls near the 4 second mark. In order to improve phase separation the machine has temporarily sacrificed height regulation by gently striking the second ball (dashed line) and firmly striking the first ball (solid line) thus changing their times of flight and forcing the balls to be more out of phase with each other. Finally it is worth noting that at the 5.75 second mark of this experiment there is a large transient in the track of the second ball. This is due to an erroneous centroid measurement of the ball, and was recovered from without significant difficulty.

Figure 10 gives some feeling for the horizontal regulation performance of the system. The position of the balls is only controlled to an error of approximately 15 cm. This variation seems attributable to the various noise sources in the system, most notably surface irregularities on the paddle and controller error in the positioning of the roll axis of the robot. Fortunately, the proportional derivative terms in (3) are sufficiently stabilizing to result in acceptable performance.

Finally, Figure 11 and 12 allow the reader to assess the effect of adding follow-through to the performance of the robot controller. The former of these figures shows the generated reference trajectory (solid line) and the tracking performance for the shoulder joint of the robot (dashed line) without the use of the follow-through. The later figure plots the same variables for an identical run with the inclusion of the follow

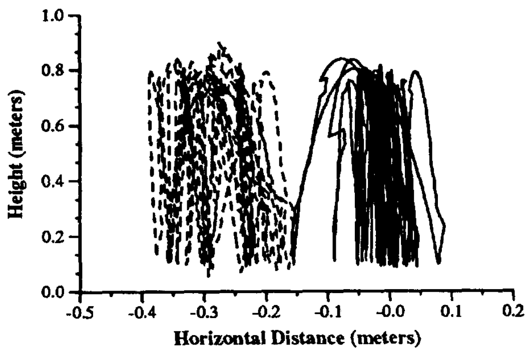

Figure 10: Projection of the ball trajectories onto the plane x-y=0 — facing the robot "head on".

through. As can be seen the sharp corner, which corresponds to a step in commanded velocity, disappears from the reference trajectory. Associated with this is a significant increase in tracking performance as can be seen from the greatly reduced errors immediately after the impact (which occur at the peaks in the graph).

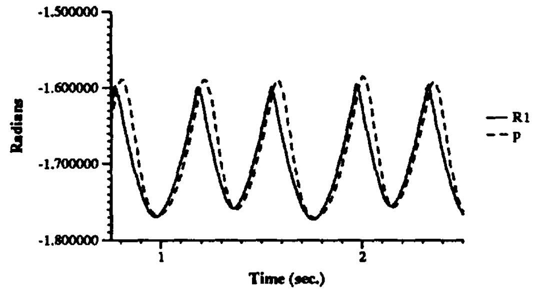

Figure 11: Tracking performance of the shoulder joint without a follow-through.

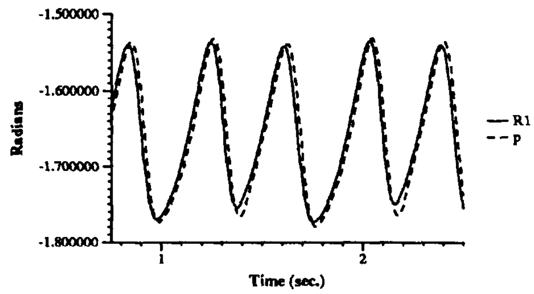

Figure 12: Tracking performance of the shoulder joint with a follow-through.

References

[1] M.Bühler, D. E. Koditschek, and P.J. Kindlmann. A Simple Juggling Robot: Theory and Experimentation. In V. Hayward and O. Khatib, editors, *Experimental Robotics I*, pages 35–73. Springer-Verlag, 1990.

[2] M. Bühler, D. E. Koditschek, and P. J. Kindlmann. Planning and control of a juggling robot. *International Journal of Robotics Research*, (to appear), 1992 .

[3] M. Bühler, D. E. Koditschek, and P.J. Kindlmann. A family of robot control strategies for intermittent dynamical environments. *IEEE Control Systems Magazine*, 10:16–22, Feb 1990.

[4] Avis H. Cohen, Philip J. Holmes, and Richard H. Rand. The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: A mathematical modle. *Journal of Mathematical Biology*, 13:345–369, 1982.

[5] R. C. Goertz. Mainipulators used for handeling radioactive materials. In E. M. Bennett, editor, *Human Factors in Technology*, chapter 27. McGraw-Hill, 1963.

[6] D. E. Koditschek and M. Bühler. Analysis of a simplified hopping robot. *International Journal of Robotics Research*, 10(6), Dec 1991 .

[7] Matthew T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, Fall 1986.

[8] A. A. Rizzi and D. E. Koditschek. Toward the control of attention in a dynamically dexterous robot. In Koichi Hashimoto, editor, *Visual Servoing — Automatic Control of Mechanical Systems with Visual Sensors*. World Scientific, 1992 (In preparation).

[9] A. A. Rizzi and D. E. Koditschek. Toward the control of attention in a dynamically dexterous robot. submitted for presentation at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, July 1993.

[10] Alfred Rizzi and Daniel E. Koditschek. Preliminary experiments in robot juggling. In *Proc. Int. Symp. on Experimental Robotics*, Toulouse, France, June 1991. MIT Press.

[11] Alfred A. Rizzi and D. E. Koditschek. Progress in spatial robot juggling. In *IEEE Int. Conf. Robt. Aut.*, pages 775–780, Nice, France, May 1992.

[12] Louis L. Whitcomb, Alfred Rizzi, and Daniel E. Koditschek. Comparative experiments with a new adaptive controller for robot arms. *IEEE Transactions on Robotics and Automation*, (to appear), 1992.