Dynamically Dexterous Robots
via Switched and Tuned Oscillators

D. E. Koditschek
Artificial Intelligence Laboratory and Controls Laboratory
EECS Department, College of Engineering
University of Michigan
1101 Beal Ave, Ann Arbor MI 48109-2110
kod@eecs.umich.edu

December 14, 1994

Summary

1 Introduction

This document proposes a renewal of funding to support a program of autonomous robotics research that seeks to encode abstract user goals in the language of dynamical systems theory. The theme of expressing one's goals in what is effectively "machine language" promotes the development of automatically generated robot controllers that achieve those goals in a verifiable manner. The focus is upon getting the robot to control the environment in the right way, although, not surprisingly, this requires careful attention to robot control and sensor architecture as well. The approach is founded on the premises arising from previously funded work in my laboratory that has resulted in several empirical and theoretical advances stemming from our systematic use of "practicable stability mechanisms" for nonlinear oscillators. The new emphasis in the proposed future work is placed upon developing a systems theory for combining existing behaviors to get more complex new ones via effective coupling techniques for these oscillators.

The bulk of the requested funds would be used to support two projects as the motivating focus for a constellation of theoretical questions discussed below. As in all of our past work, the theorizing would be yoked directly to three specific experimental problems as representative of the broader problem domain discussed in this document. Specifically, we propose to extend the previous work on sensor driven periodic batting to two new task settings. One addresses the problem of "stitching together" existing simpler behaviors and targets a very much abstracted physical setting. The other addresses the role of sensorless dynamical manipulation and addresses what some have come to embrace as a problem fundamental to the contemporary needs of flexible automation.

Sensor Driven Dynamical Pick and Place: use a flat paddle to acquire balls thrown into the workspace; balance or bat them as required through a clutter; loft them into an out of reach destination receptacle; resist unanticipated perturbations along the way;

Sensorless Dynamical Part Orientation: design the open loop motion of a three degree of freedom shaking table so that it brings a tray of randomly arranged identical rigid bodies all to rest in the same orientation.

The controllers required to carry out these tasks autonomously presently exceed the capability of any robot or programmable mechanism we are aware of (including, of course, our own) and thus present a suitable...
challenge to the design principles underlying dynamical dexterity that we have developed to date. At the same time, these settings are chosen to illustrate that dynamically dexterous capabilities have application both to short term problems of factory automation as well as to longer term autonomous robotics. Finally the two settings seem to represent ends opposite of an entire spectrum of systems coupling techniques we wish to explore: the orientation task presents the problem of parallel coupling; the pick and place task presents an application demanding sequential coupling.

1.1 Motivation and Overview

Machines play chess better than almost every human expert yet no machines have yet been built that can manipulate the pieces as well as the youngest human novice. Our past work, supported under the previous NSF grant (IRI-9123266), has resulted in an expanding family of working machines that exhibit superlative dexterity (arguably superior to that of a human) in a narrow domain as well as an immature but growing body of theory to explain how. The research proposed here focuses both on enlarging the domain of robot dexterity and on attempting to extract from the algorithms that confer it a primitive but very robust sort of computational intelligence.

1.1.1 Dynamical Dexterity

Task domains requiring dynamical dexterity present two key features. First, there are more (unactuated) degrees of freedom to be manipulated than there are robotic (actuated) degrees of freedom with which to manipulate. In consequence, as is intuitively clear, contact with the environment must be repeatedly made and broken, and, as seems less obvious but can be formally demonstrated, event driven robot strategies must have a hierarchical nature. Second, the environment's degrees of freedom are subject to independent dynamics. In consequence, since they can be only intermittently controlled, robot manipulation strategies stemming from the control theoretic traditions of plant inversion and trajectory tacking cannot generally be applied.

The motivation for considering environments possessed of more degrees of freedom than the robot seems compelling: in both structured (assembly operations) and unstructured (fold ing the laundry) settings we build general purpose machines for the express purpose of performing work on diverse objects. The motivation for focusing on dynamical environments is less familiar. Practically speaking, many manipulation tasks can be effectively accomplished by making and breaking contact far from the quasi-static regime and allowing the natural dynamics of the world to finish the job. Human manipulation almost always includes transitions to and from force closure grasps through fumbling, throwing and catching. Yet despite a modest swelling of the ranks in recent years, most researchers still shun such approaches to robot manipulation. Thus, one of the central aims of the work proposed here will be to move a little beyond our previous concentration on abstract laboratory demonstrations (juggling one or two balls with different kinematic chains and sensor systems) and find behaviors that point our robots more discernibly toward potential application settings. This slightly more utilitarian shift brings to the foreground our central fundamental concern: the development of a systems theory for building new and more complex robotic behaviors from suitable combinations of old ones.

1.1.2 Dynamical Systems Theory

Specifically, we propose to apply certain concepts and tools of dynamical systems theory to this problem. Looking back on our work accomplished, our notion of "autonomy" — the ability to recover from unanticipated setbacks — derives from considering the volume of the domain of attraction around a forward limit set. Our notion of "coordination" hinges on the interpretation of the contact set as a generalized Poincaré section that we suspect may be used to measure progress toward a goal (even if it is not necessarily a periodic motion as will occur when we move away from pure juggling) in the presence of the switching controllers required to

\footnote{This admittedly aging aphorism — it dates back at least five years [7, 7] — seems as true today as in the late 1980's when we coined it.}
achieve the tasks at hand. Our notion of "generalization" is inspired by the role of topological equivalence in formally classifying behavior and informs the controller tuning techniques we are attempting to introduce in the automatic generation of plans. Looking ahead to the new problems that we don't presently know how to solve, there seems to be increasing hope that certain ideas from the theory of invariant manifolds — the notion of a filtration [?] and the notion of normal hyperbolicity [?] — may offer a great deal of help in sorting out how to proceed.

The motivation for this reliance on the tools developed from the study of dynamical systems emerges most immediately from the emphasis on dynamical dexterity. Clearly, there are few other analytical tools to apply when objects are flying through the air and being batted at by the robot as happens in most of our laboratory work. But, as argued next, the benefits of focusing attention on the dynamics of the robot-environment interaction extend well beyond situations where the exchange of kinetic and potential energy is important. Even in the most generic quasi-static manipulation tasks, robot behaviors emerge over time in response to the machine's sensory experience of the environment. As with any phenomenon that evolves over time according to rules that affect a next situation in terms of measurements of the present situation, the evolution of such response patterns is properly viewed from the perspective of dynamical systems theory.

1.1.3 Feedback Systems Design

The tradeoffs between feedforward (predictive planning) and feedback (reactive planning) have been by now exhaustively debated both in the robotics literature and beyond [?, ?] to the point that there seems little worth in holding forth for one or the other in abstraction. Crudely speaking, feedforward achieves performance and feedback achieves safety: clearly, both are needed and may be applied at the various levels in the robot command hierarchy. Our view is that performance may always be added after a system is working safely but that the converse may be less true. Thus the work proposed here focuses almost exclusively on feedback — design of sensor policies that yield the state of an environment; design of actuation policies that achieve a next state as a function of the present. We explicitly use feedforward control techniques to gain performance where needed for our empirical work (for example, see the footnote on page ??) but, as roboticists, we remain convinced that the fundamentally difficult questions in robotics relate to the matter of how to guide the sense/plan/act loop as an integrated evolving whole — a dynamical system — rather than how to boost performance of some controller (a fundamental problem for control theorists) or some geometric computator: (a fundamental problem for computational geometers).

The notion of safety in question here relates to the predictability of the inevitably encountered error detection and recovery cycle. Our experience — both in the laboratory as well as through almost a decade of industrial consultation — suggests that failures in machine reliability frequently occur because of events which are not intrinsically unrecoverable but which violate dramatically our models and cannot be anticipated. Wire-wrapped boards send occasional spurious signals, balls fly off paddles in completely "wrong" directions, defective parts slide of the gripping tool in a novel fashion, all manner of temporary setbacks occur which "might have been made right with a little more thought." But there can never be sufficient thought. While control and recovery policies founded on human anticipation are clever, they intrinsically take an "optimistic" view — that any possible environmental state transitions have been included in the exception handler. In contrast, feedback policies take the most "pessimistic" view in providing a response to any possible state the environment could be in at any moment.

State driven machines, when they work, are indefatigable goal seekers. The question remains: what sort of goals can be encoded and how can the feedback designs be guaranteed to work? Once the robot is coupled to its environment via a feedback policy, the ultimate future of their states is determined by the limit set of the closed loop dynamical system they mutually define. Thus, to state the previous question a little more formally: what sorts of limit sets can we build with our feedback policies and what domain of attraction will obtain around the portion we care about that encodes the goal?
1.1.4 Tunable Oscillators and Their Interconnection

Addressing such questions has led us back to the problem of building oscillators — a notion of how to build robot systems that dates back at least to Wiener [?]. Throughout this proposal the term oscillator denotes a dynamical system with a properly contained compact forward limit set (that is, the future of every trajectory lies in a bounded region that is a strict subset of the state space). For example, global point attractors are trivial (period zero) oscillators. The term tunable denotes the property that the oscillator persists (despite likely bifurcations in the limit set topology) over large volumes in the parameter space that defines the dynamics.

Much of our past work in the laboratory and in theory has been devoted to finding and using stability mechanisms that give rise to oscillators in this sense. As will be discussed in Section ??, we have already identified and systematically deployed two such "practicable stability mechanisms" and hope to find at least one new one through the proposed forthcoming investigation of the sensorless parts orienting problem. Using these tools, we have been able to build simple oscillators whose goal set includes enough of the limit set to attract all but a set of measure zero of initial conditions. Moreover, we have been able to suggest how their tuning capabilities afford a means of generalization from one model task across families of tasks in the same domain. We now hope to find the means of hooking such oscillators together in the right way (again, using the appropriate state information) to achieve both parallel and sequential combinations that produce dramatically new and potentially useful behaviors.

Beyond the historical precedent, the term oscillator also points toward some newer considerations that have arisen in our work. The connection to nonholonomic control theory will be made in the next section. Our application of these ideas to the analysis of Raibert's hopping machines begins to make strong enough contact with the biological literature on Central Pattern Generators [?] and functional morphology [?] that we will shortly be preparing an independent proposal on the control and analysis of locomotion strategies in biobots.

1.1.5 Summary: Autonomy and Language

Task encoding: what it is; how to do it. Has given some suggestive behaviors.

Natural partition on parameter space — how to use it? Find a natural partition on the state space — symbols from signals — gives language — how to use it?

1.2 Review of Relevant Literature

The work proposed here addresses the following generic problem. Let the state of an environment be represented by some set of elements \( b \in B \) (typically, the positions and velocities of each degree of freedom) and let \( r \) denote the means by which a robot can change the state of the environment according to the rule\(^2\)

\[
U' = f(b,r),
\]

(1)

Suppose that a user's goals have been encoded in the form of a set of desired states, \( G \subset B \). We are interested in designing a feedback policy\(^3\)

\[
r = \Phi(b),
\]

(2)

resulting in a closed loop system, \( f(b, \Phi(b)) \) that brings as much of \( B \) as possible into the desired goal set, \( G \). More specifically, we are interested in reasoning about parametrized families, \( \{\Phi, \gamma \}_{\gamma \in \Gamma}, \{\tilde{\Phi}, \tilde{\gamma} \}_{\tilde{\gamma} \in \tilde{\Gamma}}, \ldots \), whose resulting limiting closed loop behavior, \( G, \gamma, \tilde{\gamma}, \ldots \), may be tuned and either coupled together in parallel (in

\(^2\)The primed variable, \( U' \) denotes either the state at next stage or the time derivative of the state, depending upon whether our model of the state's evolution is discrete or continuous in time — both sorts of models will typically be required at different levels of the hierarchical systems to be discussed in this proposal.

\(^3\)This abbreviated representation allows to some extent the more accurate notation to be developed in equations (4) and (5) and is introduced here for ease of exposition.
the case that there is a second robot, or more) or switched between in sequence (in the case that \( r \) remains the only actuated degrees of freedom) to achieve autonomously some more complex goal than any one of the families alone might achieve. While these switching and tuning procedures are presently to be developed by hand, the contention is that adaptive or learning techniques can replace the hand tuning, and finite state machines replace the hand generated switching rules.

In contrast, much work in robotics is concerned with developing plans,

\[
\begin{align*}
    r &= \Pi(t; b_f) \\
    b_f &= f(b_i, r) \\
    r' &= \Pi(t, b'_f)
\end{align*}
\]  

(3)

to bring \( b \) from a specified initial condition, \( b_f \) to a desired final condition. In artificial intelligence the tradition has been to write \( \Pi \) down in the form of "if-then-else" statements. In control theory, the tradition has been to write \( \Pi \) down in a form that effectively "inverts" (?) around a reference path from start to finish (which may itself "optimize" some additional criterion). Because they are written by humans, such plans (3) can result in very impressive performance when all is as modeled. But \( \Pi \) often is very sensitive to \( b_f \) (an open loop move-box-to-pallet procedure will fail badly if the box is not initially located as assumed) and relies very strongly upon the predictive model (1) (removing a bottom box from the pallet during the stacking operation will result typically in catastrophe). Of course most implemented robot systems surround (3) with periodic sensor derived "sanity checks" and include "exception handling." But no human programmer can anticipate all the varied ways in which the real world will depart from the response model (1). And assuming, as is typical, that anticipated errors are recovered by invoking a variant of (2) with the new view of the present environment, \( b'_f \), there is established an effective closed loop,

\[
\begin{align*}
    b'_f &= f(b_i, r) \\
    r' &= \Pi(t, b'_f)
\end{align*}
\]

a form of (2) whose steady state properties are almost never worked out and, moreover, rarely easy to ponder. Since we hope to study the reactions the world will have to our choice of actions, we prefer to start with (2).
plane, geometric reasoning for assembly was carefully addressed by Sanderson et al. [?] who derived sequences of assembly matchings from geometric parts descriptions (without considering the continuous part motions along the way). More recently, Wilson et al. [?] have studied the generation of motion sequences for disassembly from CAD/CAM descriptions. These approaches can yield motion plans for parts but leave unanswered the matter of how a manipulator may induce such motions without penetrating any of the rigid bodies being manipulated. The combined manipulator-part motion planning problem has been introduced in a very fine and seminal paper by Alami et al. [?] but seems not to have been pursued in recent years. In any case, geometric reasoning will at most give rise to a manipulator plan [?] with no consideration to introducing sensor “verification” — error detection and recovery procedures — along the way.

**Discrete Events and Real-Time Computation** The difficult question of how to reason about the interplay between sensing and recovery at the event level in robotics has been considerably stimulated by advent of the Ramadge-Wonham DES control paradigm [?]. In some of the most careful and convincing of such DES inspired papers, Lyons proposes a formalism for encoding and reasoning about the construction of action plans and their sensor-based execution [?, ?] but explicitly avoids the consideration of problems wherein geometric and force sensor reports must be used to estimate progress and thereby stimulate the appropriate event transitions. Moreover, while his methods are constructive enough to produce high level programs, the question of how to implement these programs at what would almost surely be a distributed computational environment remains unexamined. Such questions, in turn, seem to remain the isolated province of the real-time computation community [?]: even the most recent work in this field addressing robotics applications seems still concerned primarily with internal consistency (e.g., avoiding deadlock) rather than correctness relative to a specific planning/control algorithm or the relation of the computational processes to the physical events being controlled [?].

Our focus is on understanding how to deploy the robot’s behavioral repertoire (a combination of sequential transitions and parallel mixture) on the state of the environment at its least immediate level — the position, velocity and force data derived from sensor policy interpreting directly the robot’s sensor reports. The only other work we are aware of that attempts to study a specific class of discrete event systems arising from the transitions across a partition on the state space of a dynamical system is by Ramadge [?

The work proposed here rests on an extensive foundation of working real-time control practice, as discussed in Section ???. Our lab implementations do not (yet) enjoy any provable properties relative to the original formally developed algorithms.

**Error Detection and Recovery** Finally, the critical matter of how to integrate sensors into a system running in real time, how to extract useful information from the deluge of data they offer, and

**Error Detection and Recovery** Despite the dissimilarity in emphasis on quasi-static manipulation, the descendants of the “LMT” tradition have the most relevance to the work proposed here and will be discussed below in greater detail.

**Donald and Erdmann**

In this body of work, the closest comparison may be made to the EDR paradigm of Donald [?]. Despite the many dissimilarities (his is the quasi-static manipulation regime, his environment is uncertain but is not moveable, Unlike Donald’s problem framework we are interested in a In place of pre-computed pre-images of error detection and recovery point of view introduced to robotics by Lozano-Perez and colleagues [?] (hereafter, “LMT”) and students [?, ?].

**Sensorless Manipulation** Essentially, after a safe closed loop is turned on, we wish to think of it in the same terms as one of Mason’s funnels. Whether the sensing is done “actively” in the electronics or “passively” by the interaction of the mechanical surfaces seems incidental.

...they consider piecewise constant plans, Π
Reactive Planning  In a previous review I have contrasted three different approaches to dynamical dexterity based upon event-based automaton reactions to the environment state. Andersson's ping-pong player even more challenging task domain than our range of problems (games against opponent rather than merely against nature) but no theoretical insight. McGeer's walking robots good model for our approach to sensorless manipulation but we're interested in sensor/sensorless combinations and we seek a global point of view. Raibert's work is the most direct inspiration for what we're doing both in terms of encoding as well use of automaton.

Brooks' subsumption architecture seems most carefully thought out in relation to quasi-static environments. No documentation - the rest of this world documents pretty carefully. Major difference is that we embrace the question of how the reactive laws encode a goal with plan and how to manipulate them both practically and theoretically.

1.2.2 The Control Literature

Noholonomic Control Problems

Intelligent and Learning Control Literature  Switched and tuned feedbacks becoming important in intelligent control Noholonomic constraints necessitate switching - not even (the graph of a function is included in some continuous upper or lower envelope) semi-continuity Identify the appropriate function space — domain and range of required function — to promote the application of automatic programming or "learning" techniques [?, ?].

2 Results from Prior NSF Support

Report on the results of two recent grants — PYI and 9123266.

Key features: under-actuated environment with no dynamics; completely actuated environment with dynamics; under-actuated environment with dynamics

Task encoding: period zero oscillator; period one oscillator; more complex (hierarchy of games)

Over the last few years I have been using geometric and quasi-static motion planning and manipulation planning problems to think through a number of issues that arise in more complex form when the robot and the environment it must manipulate are governed by the exchange of kinetic and potential energy. In order to introduce the methods

Juggling seems an apt metaphor for the larger problem of keeping actuator and sensor attention focussed on the right aspects of the worlds multiply evolving features necessary to exert the desired influence.

As in the juggling problems we have previously worked on, the dynamical pick and place behavior will not be trivial to get working in the lab and incorporates the virtue of falling dramatically and unequivocally for as long as we do not get it right. Yet, at the same time, the robots we build to perform these tasks remain simple enough to permit a comprehensive systems level view of

Computational Architecture: As reported in [?], we have been able to advance some rules of thumb regarding the interrelation of algorithm design to process assignment to processor topology in message passing distributed real-time computational environments. We have as yet no proofs that our correct algorithms are correctly instantiated in software nor that the resulting asynchronous MIMD hardware will avoid deadlock. But we find it quite easy to teach newcomers to the lab and graduate students in the introductory robotics course how to develop rapid working prototype systems in this same style.

Sensor Architecture: As reported in [?]

where r is to be formed from a plan of action, a, that depends upon the robot's state, p, and its knowledge about the environment.
Let \( \hat{b} = s_p(b) \) denote the raw sensory data available to the robot when it is in state \( p \) and let \( \hat{b} \) denote the robot's best reconstruction of the true world state according to some sensor policy scheme
\[
\begin{align*}
\hat{c}' &= k(c, \hat{b}) \\
\hat{b} &= l_c(\hat{b}).
\end{align*}
\] (4)

**Problem:**

- Encode the user's goal for the state of the environment by the set \( G \in \mathcal{B} \).
- Design a robot policy,
\[
\begin{align*}
p' &= m(p, \hat{b}) \\
r &= a_p(\hat{b})
\end{align*}
\] (5)

so that the closed loop dynamical system formed by the coupled evolution of the environment, sensor, and robot policy,
\[
\frac{d}{dt}\begin{bmatrix} b \\ c \\ p \end{bmatrix} = \Xi(b, c, p)
\Xi := \begin{bmatrix}
f(b, a_p \circ l_c \circ s_p(b)) \\
q(c, s_p(b)) \\
m(p, l_c \circ s_p(b))
\end{bmatrix}
\] (6)

can be guaranteed to bring \( b \) into \( G \).

Assume that a robot has some limit on available torques and velocities so that its state space is a compact subset, \( \mathcal{P} \) of any possible velocity at a free placement. The total space, \( \mathcal{T} = B \times \mathcal{P} \) is the set of all non-penetrated positions and velocities that the robot and environment might be in. Contained within the set of ball states for a particular sensor state, \( c \in \mathcal{S} \) is the visible region, \( \mathcal{V}_e \subset \mathcal{B} \) within which the cameras can report two images \(^4\) and the velocity (rate of change of pixels on the image plane) is sufficiently small that the ball will not be lost. The *impact set*, \( \mathcal{I} \), consists of all tangents over the *contact set*, \( \mathcal{C} \) — those configurations in \( \pi_0 \mathcal{T} \) where the robot touches the environment \[?] — and hence is properly contained in the boundary, \( \partial \mathcal{T} \). The kinematics of contact can be complex in general \[?] but for the simple case of concern in this proposal — a paddle and ball — we have shown \[?] how to compute the inverse kinematics, \( k^{-1}_c : \pi_0 \mathcal{C} \rightarrow \pi_0 \mathcal{P} \), to get a contact for each reachable ball configuration.

(laid out in a recent tutorial \[?]?): robots are programmable machines that perform work in the physical sense of the word — exerting forces over motions; the computational representation of work takes the form of a dynamical system; goals may be effectively encoded in terms of the attracting set of a closed loop dynamical system that would result from coupling the

2.1 Switching Techniques for Coupling Period Zero Oscillators

When the excess environmental (unactuated) degrees of freedom relative to the robot (actuated) degrees of freedom have no independent dynamics then useful tasks may be encoded as period zero oscillators. In such cases, when the excess of unactuated freedom necessitates making and breaking contact, we have found it possible in simple settings to develop simple switching laws that couple together automatically the right controller in the appropriate order as to guarantee that the goal will be achieved.

We have shown how to interpret

2.1.1 A Stability Mechanism for Period Zero Oscillators

Consider the case of perfect sensing (\( s \) is the identity mapping and \( k \) is zero and \( l \) is the identity in (4)) and complete and continuous actuation wherein \( r \) is the vector of motor forces or torques applied directly to the positions and velocities of the bodies to be moved, \( b \), and (1) is either a first order (generalized

\(^4\)We have developed state a estimator to contend with the situation that one or the other or none of the cameras have the ball in their view.
damper model) or a second order (Lagrangian model) system depending upon whether we wish to describe a quasi-static (e.g., friction forces dominate inertial forces) or a truly dynamical (inertial forces dominate either because friction forces are relatively small or velocities are to be relatively large).

A (nondegenerate) dissipative Hamiltonian system has a particularly simple limit set: it consists of isolated equilibrium points. Its evolution toward the limit set is governed by a natural global measure of progress: the total energy or Hamiltonian serves as a Lyapunov function [?]. When the Hamiltonian has one unique minimum then we have a period zero oscillator.

Application: Purely Geometric Environments When the surrounding world doesn't move at all, for example, in the classic robot motion planning problem, then one may follow the suggestion of Khatib [?] and consider the use of potential functions for obstacle avoidance and goal seeking behavior. Our earlier work has shown that Khatib's ideas may be extended to the development of period zero oscillators through the introduction of navigation functions [?], a completely general solution to the motion planning problem in principle. For purposes of this proposal it will suffice to treat a navigation function, \( \varphi \), as a global Lyapunov function for the generalized damper dynamics

\[
\dot{r} = -(D\varphi)^T(r)
\]  

(7)

guaranteeing that the robot's configuration variable, \( r \), always remains among the free placements and eventually reaches the required destination, \( r^* \) if in fact this is possible (if the initial condition lies in the component of the free space connected to \( r^* \)). The specific of motion planning problems for which we have shown how to construct navigation functions is limited but includes (essentially) all planar problems involving radially symmetric robots [?].

Application: Completely Actuated Quasi-Static Environments Consider the situation depicted in Figure ?? where an arbitrary number, say \( N \), of radially symmetric robots move in a planar world. Although the configuration space of this problem — \( \mathcal{F} := \mathbb{R}^{2N} - \bigcup_{i,j \leq N} \mathcal{D}_{ij} \), where \( \mathcal{D}_{ij} \) denotes the placements of these \( 2N \) degrees of freedom where robots \( i \) and \( j \) are in contact or overlapping — does not fit the constructive theory developed to date [?] we have become convinced through simulation [?] and are very close to proving formally that a particular construction, call it \( \psi : \mathbb{R}^{2N} \rightarrow \mathbb{R} \), is in fact a navigation function on \( \mathcal{F} \). Again, this means that when the robots follow the appropriate projection of the negative gradient vector field

\[
\dot{r}_i = -(D_{r_i}\psi)^T(r_1, \ldots, r_i, \ldots r_N); \quad i = 1, \ldots, N,
\]  

(8)

eye move in a nicely choreographed ensemble toward their respective goal locations without ever bumping into each other.

2.1.2 Assembly Problems Necessitate Switching

Now maintain the idealized sensor assumption but consider instead the problem of incomplete actuation. For example, in the simplest quasi-static (purely kinematic) version of such problems, (1) takes a form where each of the environmental degrees of freedom is governed by the generalized damper model

\[
\dot{b}_i = c_i(b_i, r)\dot{r}.
\]  

(9)

This structure characterizes most of the classical nonholonomically constrained mechanical systems [?, ?] and, as we have shown in recent publications [?, ?], also describes the essential features of assembly problems. In all of these problems it turns out that the structure of the coupling (9) precludes the possibility of \( f \) in (1) covering any neighborhood of \( 0 \) in the target space. Hence, according to Brockett's condition and extensions [?], no continuous (nor even upper semi-continuous [?]) feedback, \( \Phi \) can stabilize an isolated equilibrium state.
This leads to a hybrid structure for (5) of the form

\[
\begin{align*}
p_0(k+1) & = L(p_0(k), P(p_1, b)) \\
p_1 & = p_2 \\
p_2 & = -g_0(p_1, b) \\
r_0 & = GRIP(p_0) \\
r_1 & = p_1
\end{align*}
\]

(10)

where \( P \) is a partition indicator function (piece-wise constant on each cell of some partition of \( B \times \mathcal{P} \)). The robot's controller switches between a finite number of possible algorithms (the force laws, \( g_i \), and "gripper modes", \( r_0 \)) based upon the "wisdom" of the "higher level automaton" \( p_0 \). In a number of simple but representative special cases, we have been able to show that the hybrid closed loop system (6) does indeed drive \( b \) toward the desired destination \( b^* \).

**Application: Exogenous Assembly** Consider the situation depicted in Figure ?? where none of the discs in Figure ?? is fixed but none can move unless gripped and dragged by a robot that we assume for this *exogenous* version of the assembly problem to be located outside the workplace. To place this situation within the framework of (9), denote the location of the unactuated \( i \)th piece as \( b_i \) and refer to the position of the unactuated robot as \( r \). In the two degree of freedom version of Figure ?? all the pieces inhabit one copy of \( \mathbb{R}^2 \) and the robot is isolated in another. In the one degree of freedom version of this problem, depicted in Figure ??, the pieces inhabit the line \( \mathbb{R} \) and the robot is isolated in a parallel copy. The joint configuration space for these problems is \( \mathbb{R}^{d(M+1)} \) (where \( d = 1 \) Figure ??, or \( 2 \) Figure ??) and the freespace is \( \mathcal{F} := \mathbb{R}^d \times \mathcal{F} \), the cross product of the robot's configurations and the free placements of the discs from the previous setting.

We have solved the one degree of freedom version of this problem Figure ?? [?] and are close to a proof of correctness for its generalization Figure ?? [?]. The solution approach extends the use of navigation functions as follows.

Suppose that the robot is presently gripping disk \( i \) so that we may temporarily think of \( \psi \) as a function of \( b_i \) alone (since it is the only currently mobile degree of freedom) that is parametrized by the (temporarily) fixed disc center locations, \( (\bar{b}_1, ..., \bar{b}_{i-1}, \bar{b}_{i+1}, ..., \bar{b}_M) \) — that is, we may define a measure of progress for piece \( i \) to be

\[\psi_i(b_i) := \psi(b_i = \bar{b}_i, ..., b_i, b_{i+1} = \bar{b}_{i+1}, ..., b_M = \bar{b}_M).\]

There are now \( M \) possible "motion force laws" corresponding to the \( g_i \) in (10)

\[g_i = - (D\psi)_i^T(r), \quad i = 1, ..., M\]

of which it makes sense to apply the \( k \)th when the robot is gripping body \( k \). This will result in the part moving toward its goal until possibly blocked by some other pieces (\( \psi \) is a navigation function on \( \mathbb{R}^{dM} \), but \( \psi_i \), defined on the \( i \)th coordinate slice will in general have numerous minima depending upon the disposition of the other pieces) — it is guaranteed that none of the pieces will collide. Supposing that the robot is free and we have decided that it should move next piece \( k \), it is simple to devise a law such as

\[g_{M+i} = -(r - \bar{b}_k) \quad i = 1, ..., M\]

that brings the robot into a position where it can grip the piece and move both according to force law \( g_k \).

The problem at hand is twofold. First, one must devise a partition scheme, \( P \), and automaton, \( p_0 \), to sit over it in such a fashion that the robot will be induced to mate with and then move each of the parts in turn, again and again, until the entire assembly is complete. Second, one must show formally that this will happen.

**Application: Endogenous Assembly**
2.1.3 Summary of Results and Open Questions

Non-Convex Smooth Identical Payoff Games: Let \( \varphi \) be a navigation function on a non-convex space. Show that the derived Gauss-Seidel descent game defines a discrete dynamical taking all but possibly a set of measure zero to the minimum of \( \varphi \).\(^5\)

Non-Convex Smooth Partially Competitive Games:

2.2 Tuning Techniques for Higher Period Oscillators

Suppose the environment is subject to independent dynamics that cannot be simultaneously cancelled at all times. Typically, this is either because the robot has too few (actuated) degrees of freedom or, because it has only a general purpose gripper, the available coupling mechanics do not permit direct cancellation. It seems intuitively clear that the robot will need to be constantly visiting and revisiting those constituent parts of the environment in order to keep them regulated. Indeed, in these situations we have found that goals must be encoded via more complex limit sets, \( G \). We have concentrated on situations wherein \( G \) takes the form of a limit cycle.

In the previous problems where the environment only moves when animated by the robot, it has seemed acceptable to ignore the problem of sensing since we may rely upon the robot's internal sensors when in contact, while those parts of the environment not presently in contact are modeled as staying where they were last placed.\(^6\) In contrast, we have found it impossible even to build a simple prototype dynamical problem domain without worrying about and building sensors equipped with state estimators.

In these cases it seems critical to include an account of the sensing policy (4).

2.2.1 Manipulation by Batting

Of all the work we have done in this area, the one and two body juggling results seem to have attracted the most attention, both in the robotics field as well as in the broader scientific and lay press [?]. Perhaps this is not surprising since the reliability with which our machines recover from near adversarial perturbations (bumping the ball while in flight, occluding one or another camera for a few frames, etc.) and persistently seek the targeted periodic behavior makes for very entertaining videos.\(^7\) Indeed, it is exactly our contention that these ideas can be extended to build a variety of useful dexterous machines that are similarly single-minded in their pursuit of the user's goal behavior and ability to surmount unanticipated obstacles along the way.

We have published technical accounts in archival form concerning the robotic infrastructure that makes these machines work in the laboratory: the real-time distributed computational architecture [?]; the jointspace level adaptive inverse dynamics controller [?]; and the real-time "active" stereo image processing architecture [?]. Thus, to save space, this discussion will concentrate on those aspects of the work that point most directly toward a systems theory for more building complex behaviors.

In these problems, the robot affects the environment through collisions that have the effect of resetting the environment's velocity component normal to their contact. For a particular environmental state, \( b \), denote the set of possible contacts the robot can make by \( C_b \subseteq P \). The environmental evolution law, (1), now takes the form of a collision rule defined over the contact set, \( C_b \).\(^8\)

Although batting seems to represent a very crude mode of interaction with a robot's environment its study poses three benefits. First focuses attention on make-break. Second characterizes situations where

\(^{5} \)It seems possible that this result may relatively easily fall out of recent work on asynchronous iteration by Kuznetsov and colleagues [?]

\(^{6} \)Acceptable does not mean desirable! The problem of sensor based navigation is terribly important. We simply have not yet gotten around to putting a sensor model into our period zero oscillator work.

\(^{7} \)Al Rizzi's (the student whose thesis concerned the spatial one and two juggle) submitted footage was a finalist for best entry in the 1993 IEEE ICRA video proceedings.

\(^{8} \)Formally, this is a map on the tangent bundle over the contact set as laid out, for example, in (??).
sculpted degrees of freedom are already being used for other purposes so have only general shapes left to manipulate with. Third, relationship to locomotion. We intend in the next round of research proposed below to introduce more diverse manipulation primitives as well.

2.2.2 Mirror Laws

One of the most interesting ideas to have come out of the lab was the notion of a mirror law, introduced in Bühler's thesis [?] and generalized in Rizzi's thesis [?]. Mirror laws entrain the robot's trajectory around that of the ball according to a user-selected "generalized reflection" of the ball's motion that the robot is forced to track.

Formally, a "Mirror law" [?] is a map assigning to each environment state a robot state,

\[ p = M(b) \]  \hspace{1cm} (11)

so that environmental motions, \( b(t) \) generate robot reference trajectories, \( M \circ b(t) \) that may be tracked according to by robot controllers (??) of the form

\[ \begin{align*}
\dot{p} &= \Gamma[M \circ b(t)] \\
\dot{r} &= a_p(b).
\end{align*} \hspace{1cm} (12)\]

Here, \( \Gamma \) is either an adaptive inverse dynamics controller [?] or a high gain controller, depending upon the kinematic complexity of the robot. If \( p \) tracks \( M \circ b(t) \). \hspace{1cm} (10) Contact occurs in the "mirror set,"

\[ M := \{(M(b), b) : [k \circ \pi_0 \circ M - \pi_0 | (b) = 0) \subset C \} \hspace{1cm} (13) \]

at which points the results of impact (1) are modeled by applying a coefficient of restitution law determined by the ball's velocity, \( \pi_1 b \) and that of the robot consequent upon the mirror's transformation of its velocity, \( a_p(b) = D_{\pi_0 p} k \cdot \pi_1 M(b) \).

The resulting closed loop dynamics (6) take the form of a discrete dynamical system arising from sampling the ball's continuous flight trajectory following the \( k \)th impact, \( F^k(b_k) \), and re-setting its velocity according to (1) at the \( (k+1) \)th impact which occurs at time, \( \tau_C \), the function relating \( t_{k+1} \) to \( b_k \) that solves the explicit equation

\[ (k \circ \pi_0 \circ M - \pi_0) \circ F^k(b_k) = 0. \hspace{1cm} (14) \]

Gedanken Problems: Point Mass Environment and Robot. We have posed and solved various versions of the one degree of freedom endogenous and exogenous assembly problems using batting in the presence of environmental dynamics. For example, for endogenous batting in one degree of freedom against gravity we have shown how to shape mirror laws \( M \) to yield closed loop systems governed by unimodal return map dynamics [?]. These dynamics are famous for their rich and universal pattern of period doubling bifurcations [?] so that we achieve a very rich repertoire in our parametrized family \( \Phi_u \).

As another example, when batting without gravity in one degree of freedom we have had to add extra dynamics in the robot controller (??) in order to achieve a discrete period zero oscillator that places the piece in a specified location from any initial condition [?].

Again, when batting against gravity in one degree of freedom we have generalized earlier results of Atkeson and Schaal [?] to give sufficient conditions for global asymptotic stability of a specified periodic orbit — a period one oscillator that cannot bifurcate.

---

9 We have developed a generalization of the standard Slodin-Horowitz [?,?] controllers that permits tracking around a nonlinear first order error system [?].

10 In principle, it ought to be possible to formalize and solve these controller design problems within the framework of optimality. This cannot be done directly: the traditional optimal tracking paradigm presumes knowledge of the entire future trajectory and robot initial conditions — neither of which is ever known in advance in our problems. The alternative — forcing our problems into an approximated framework accessible to a generalization of such methods seems to contradict the impulse toward "optimality." We prefer to tune directly the gains in our controllers rather than indirectly through adjustments in some ad hoc cost function.
How to get two degree of freedom point-mass behaviors when the robot has two actuated degrees of freedom?

Experimental Work: Point Mass Environment and Paddle Robot  The question now arises as to how these one degree of freedom oscillators may be coupled together to get more useful behaviors. It is for this purpose that we have built our series of experimental machines in the effort to apply the results of the previous discussion to physical problem settings.

The planar vertical one-juggle [?] arises from combining the period-one seeking vertical (1 dof) mirror law, \( M \), with a period-zero seeking horizontal (1 dof) potential law. This is generalized in the spatial vertical one-juggle [?] by combining the vertical mirror law with a period-zero seeking planar (2 dof) potential law. Although these constructions give rise to the dramatic laboratory demonstrations described above, and we have a pretty good idea of how they generalize absent obstacles in the workplace, it would not be accurate to say that we truly understand why. We have recently overcome a number of technical obstacles to write down for the first time the closed loop system (6) resulting from this control policy for the planar version of the problem [?]. We find that the "vertical axis" is indeed an invariant submanifold on which the restriction dynamics is exactly the unimodal map from the 1 dof problem (?). We also find that the horizontal dynamics looks like a discrete (non-uniformly sampled) version of the period-zero dynamics.

2.2.3 The Two-Juggle: "Stitching" Together Degrees of Freedom

The robot controller (5) is designed to track \( C^2 \) (twice continuously differentiable) reference trajectories and we have documented in the archival literature [?] performance comparable or superior to that of the best manipulator tracking algorithms reported to date. In the two-juggle, the controller's desire for smooth trajectories must be reconciled with the need to switch between tasks — the mirrors laws for the two distinct balls. To achieve this reconciliation we resort to a convex combination of the competing tasks’ trajectories,

\[
\Omega_\sigma(x,y) := \sigma x + (1-\sigma)y, \tag{15}
\]
governed by a function \( \sigma \) that weights the urgency of the respective tasks according to a phase function

\[
\tau(z,\dot{z}) := \frac{\dot{z}}{g \dot{z} + \dot{z}^2 / 2} \tag{16}
\]
that measures the portion of the flight phase remaining assuming a Gedanken robot applied to the vertical projection of each ball's flight.

2.2.4 Ongoing Work: Rigid Body Environment and Paddle Robot

2.2.5 Sensors and Their Added Dynamics

State estimator for mirror laws

Control of Attention  Ability to handle occlusions and out-of-frames represents the ability to begin to remove the "keep-you-eye-on-the-ball" weakness of the mirror based methods.

entrainment of Shannon Juggler

---

11 The adaptive inverse dynamics controller we have developed represents an example of our systematic use of feedforward in the control hierarchy. We incorporate in the feedforward loop not only inertial terms but distinct coulomb friction and static friction models to gain this performance. Because we work within such a powerful computational environment [?], this complicated forward loop is computed at almost the same update rates as the feedback loop (roughly .5 KHz vs. 1 KHz) and this surely gives us a significant boost in performance.

12 Please see the footnote on page 77 for an explanation of why we ignore the optimality framework for devising these reference trajectories.
Stability Mechanisms  Unimodal Maps; Shannon Juggler

Planar Juggler  analysis and experiments

2.2.6 Multiple Degrees of Freedom

can get cross product of two unimodals in (by coupling x to z in the one juggle) and pick up complex coordinated patterns as the bifurcation diagram goes up that are strongly attracting (they "feel" good).

2.2.7 Summary of Results and Open Questions

Domain of attraction must be expressed as some inverse projection, $\pi_B^{-1}[C]$, over the contact set since the controller will completely specify the robot's behavior in the

Global Palm: need to have better estimates for domain of attraction of palm: solve the global version of the ball and beam problem

Stitching Techniques: Need for stitching arises when coupling of behaviors is to be accomplished with fewer degrees of actuated freedom and in a dynamical environment.

More Complex Limit Sets: coupling bifurcating 1 DOF systems

Dynamical Obstacle:

3 Proposed Projects

We propose to pursue two distinct experimental problems that extend the scope of our previous work in different directions. The first presents a sensory driven task domain wherein several different dynamically dexterous behaviors must be "stitched" together to achieve a composite goal that cannot be accomplished by any one technique alone. The second presents a sensorless manipulation setting wherein two different oscillators must be coupled together in the right way to achieve the goal.

3.1 Theoretical Investigation

3.2 Dynamical Regrasping: Stitching New Behaviors From Old

3.2.1 Problem Motivation

A flexibly deployable robot of the future is equipped with a general purpose manipulator capable of handling many differently shaped and sized objects in unstructured environments: thus it is optimized for none. The robot is efficient and handles its objects in a dynamically dexterous fashion: guarded moves and quasi-static manipulations are reserved only for fragile loads — most objects are grabbed, batted or thrown. The robot is presented with objects that may not be freely acquired (e.g., they might need be knocked clear before they can be grasped) or may not be freely placed (e.g., they might need be lofted up onto a high shelf).

No human supervisor will be available to develop a strategy for teasing any particular object into a position where it can be grasped and maneuvered. Pre-programmed recovery procedures will not be feasible since the possible failure scenarios are far too numerous to list ahead of time and too varied to plan around in advance. Nevertheless, the robot is autonomous: it has a repertoire of dexterous maneuvers and is capable of applying these flexibly in accord with the exigencies of the present situation.
3.2.2 Problem Statement

Empirical Task Our three degree of freedom (nearly spherical kinematics) juggling robot is equipped with a flat paddle and a field rate (60 Hz) stereo camera system. As shown in Figure ??, its workspace will be cluttered with fixed obstacles — some suspended from the ceiling creating low “doors” and some protruding from the floor creating high “windows.” The suspended obstacles hang low enough that the paddle can only just pass through the doors when level with the ground. The protruding obstacles are high enough that the robot must raise its paddle to pass through.

The robot will be thrown a ball without warning into one of the free cells defined by the obstacles. The robot’s task will be to bring the ball into a bowl in one of the other cells. The bowl will be located out of the robot’s “reach” — either higher than the paddle’s level height, or outside the workspace radius — so that balls must be tossed in.

The ball cannot be tossed or batted through the “doors” because of the suspended obstacles: the robot must perform a “carry”. The ball must be lofted to get it through the “windows” since the protruding obstacles are so high that it will roll off the paddle if an attempt is made to “carry” it across: the robot must perform a “toss and catch.” When a ball is initially thrown it may have unacceptably high horizontal velocity or be heading for an obstacle and require a fast “rescue”; the robot must be able to distinguish situations where a ball can be saved from those where it would be too likely to damage itself in the effort to be worth trying. During the course of the manipulation, the work cell may be “invaded” by disturbances (we will poke at the ball with a stick as we presently do in the juggling work) that take the ball far off its course and off the track of the sequence of maneuvers previously planned.

After the robot has developed the ability to handle one ball reliably in the work cell, introduce a second. Keep both balls “safe” (off the ground — either balanced on the paddle or juggled in the air) while bringing them one by one to the desired receptacle.

Algorithmic Task Build a collection of tunable oscillators — a juggle, a “save, catch, toss, etc. — that arise from the robot’s response, to the environment’s state. For each oscillator in the collection, develop an estimator that returns the environment’s state from the available history of camera image pairs.

Given a fixed set of tunings, develop a strategy for turning on one of these oscillators and inhibiting the others in such a fashion that it can be guaranteed the robot will accomplish the overall task or “stall”

Theoretical Task Develop good estimates for the domain of attraction of each constituent oscillator. Develop a means of moving the tuning parameters around so that a

3.2.3 Present Capability

We have built on our success with the two juggle to the point where our robot can now accomplish a set of four constituents of the task presented above. We are now faced with the problem of tying these behaviors together to achieve the overall task.

These constituent behaviors have been documented in a recently completed paper [?]. A “palming” maneuver encoded as a one-juggle with vertical gains set to zero (11) provides the dynamical carry capability required to move the robot and ball through a doorway. A “tossing" behavior encoded in terms of a desired escape velocity provides the transition from continuous contact to batting required to move the ball through a window. A spatial “catch" derived from our earlier planar work [?] affords the transition from bat to contact modes. Thus we have developed a palette of dynamically dexterous capabilities that appears to be sufficient to achieve the task when mastered in the right combination. 13

\footnote{These capabilities present with varying degrees of success and theoretical understanding. The spatial juggle, the first behavior to have been achieved in our lab, functions arguably better than any of the others, and has a theoretically clearer basis as well. None of the other constituents is yet polished enough to warrant formal theoretical analysis at present but their representation (the manner in which they have been encoded) is clear and they are all functioning sufficiently well in isolation for us to proceed with the major problem of "stitching" them together.}
Why are we focusing on feedback — on robot strategies that emerge from laws of response to present circumstances? Because the alternative of open loop planning seems bound to encounter an unforeseen situation and thereby fail. It seems straightforward to devise a set of rules that would bring the ball from one specific initial condition to the final goal: for example, catch the flying ball, palm it through the door to the window, toss it through the window, catch it on the other side, then palm it to the bowl. It seems less straightforward but still possible to write a more complex program that might anticipate a great variety of initial conditions in which the ball might be found, then to match them with the appropriate sequence of behaviors that would bring the ball to the goal. It seems even more difficult to anticipate the various unintended new states that a perturbed ball might devolve toward and to develop exception handling methods that would bring it back on the planned sequence.

Instead, we prefer to rely on the deployment of feedback laws possessed of known (in some cases, conservative) domains of attraction toward a goal state.

3.2.4 Immediate Future Research Strategy

Need to explicitly write down the partition on state space induced by estimator-controller settings.

Juggle-Catch-Palm to Rest Point: The Obstacle Free Case. The domain of attraction of the most autonomous behavior — in our case, the juggle — defines the workcell. It is encoded by a total vertical energy (corresponding to desired height of the juggled ball) and a horizontal set point, thus its goal set $\mathcal{G}_{\text{juggle}}$ is a vertical line in $E$. The "palm" is encoded by a desired horizontal set point; its domain of attraction includes a large neighborhood of positions around that set point and can recover as well from very low relative velocities — it is a very skinny and long ellipse as depicted in Figure ???. The "catch" is encoded as a desired horizontal position at zero vertical velocity. Its domain of attraction seems to include all viewable vertical positions and velocities at the specified zero horizontal velocity.

Each collection of set points, $\{\zeta_1, ..., \zeta_n\}$ Now take $\Phi_{\zeta_1}$. When we arrange the setpoints for each of these behaviors so that the domain of attraction of one is contains the goal of another then we are guaranteed to be able to make the transition within the

Generalized Follow-Through On the control side, need to keep feeding twice differentiable signals to the robot. We have already used convex combination and this seems to be working.

On the observer side, need to distinguish between fast sloppy observers (good for start up and transitions across the state space partition) vs. slow accurate observers (good for lead in toward the state space transitions).

Time-of-Flight: Generalized Phase The two-juggle relies heavily upon a relative urgency function formed by monitoring the phase — the location in the duty cycle — of each ball [?, ?]. This is equivalent to measuring time to next collision, $\tau_C : B \rightarrow R$ as defined by (14). Since all the controllers introduced in Section 3.2.4 are defined using a mirror-like function (11) a generalized notion of phase obtains from solving (14) for each one to get a collection, $\{\tau_C^1, ..., \tau_C^n\}$.

When an obstacle, $\mathcal{O}$, is introduced into the environment, then assuming a boundary function, $\beta_0 : B \rightarrow R$, vanishing on the boundary of the manipulable states, $\pi_B^{-1}(\mathcal{C} \cap \mathcal{F}_0 \cap \mathcal{V})$. Denote the time of flight of a ball state with respect to $\mathcal{O}$ by $\tau_{\mathcal{O}}$ — the implicit function, that solves the equation

$$\beta_\mathcal{O} \circ F^t(b) = 0$$

for $t$ — the time to "impact." \footnote{The states that can be touched by the robot, do not involve penetration of the robot or the body, and are visible to the camera and travelling slowly enough to be estimated by the camera.}

\footnote{For our problem, $F(t)$ is quadratic and the boundary is simple so that $\tau_{\mathcal{O}}$ may be obtained directly in general, one would need to develop conservative estimates.}
"Safe" Juggle and Toss Manuevers: We propose to develop several new mirror laws for behaviors constituent to the pick and place problem. For any given obstacle, $O$, say that a robot policy $\Phi$ is *safe with respect to* $O$ if it achieves the following criteria

$$D_O(\Phi) := \{ b \in B : \tau_b < \tau_O \}$$

In other words, a safe controller for $O$ yields a *safe domain* of states, $D_O(\Phi)$, from which it can be assured that forthcoming contact will be made before the ball penetrates the obstacle, is invariant (the ball will remain safe after the next contact), and has its goal set within that invariant set.

Notice that for any obstacle and controller, a safe domain is invariant during flight, $F'(D_O(\Phi)) \subset D_O(\Phi)$, by its very definition. Thus, once $\Phi$ has been selected, only a violation of our models — e.g., a perturbation during flight or impact, a overly large observer error, etc. — can cause a transition out of $D_O(\Phi)$.

The *toss* is an "inverse dynamics" or "deadbeat" controller, $\Phi$, whose closed loop map (??) results in a fixed specified apex $b_\Phi = a \circ f_\Phi$. This is means of aiming toward a specific target point. We are particularly interested in developing a safe version of the toss for the windows and doors and out of reach portions of the ball's state space.

**The Automaton State as a Partially Ordered Set of Controllers** Say that controller $\xi_i$ prepares controller $\xi_j$, denoted $\Phi_i(\xi_i) \geq \Phi_j(\xi_j)$, if $G_i(\xi_i) \subset D_j(\xi_j)$. In this manner each point in the tuning space, $\Xi$, induces a partial order on the index set, $N$.

**Switching Logic** Each behavior has a cost function, $\varphi_i$ that takes unity on the boundary of its domain, $D_i$ and zero on its goal $G_i$. The closed loop, $f_i$, system guarantees that $\varphi_i$ never increases to unity and eventually decreases to zero when invoked on $D_i$.

A switch is an allowed transition from one controller to another.

An automaton pos:s a goal, $G$, and each behavior reacts with $G_i(G)$ (the projection of that goal onto its own state space), with a consequent $D_i(G)$. The automaton chooses that behavior with the smallest distance, $\text{DIST}(G, G_i(G))$ (this could be generalized to a "bidding price" a la Wellman if the two sets are incommensurable), such that $\varphi_i \leq 1$ and does not switch until $\varphi_i = 0$.

Suppose $b \notin G$ and $\varphi_k = 0$ and $\varphi_i > 1$ for every other $i$.

Show that if there is some behavior with $\varphi_1 \leq 0$ at start up then there will never occur a future time when this is true for no $i$.

We're studying task domains over which particular stability mechanisms are effective using notions like invariance, structural stability, etc. Would like to have a means of generating stability mechanisms...but it is not presently known how to do so in the dynamical systems literature.

**3.2.5 Specific Questions to Be Addressed**

**Task Completeness:** What set of lower level behaviors provides a set of constituents rich enough to be combined into a complete task?

**3.3 Parallel Vibratory Parts Feeding: Autonomous Manipulation from Sensorless Oscillators**

**3.3.1 Problem Motivation**

Mason and colleagues have pioneered the analysis and potential applications of sensorless manipulation in the robotics field [?]. Canny and Goldberg have enlarged this program in the effort to minimize sensing and automation complexity without unduly compromising its usefulness [?]. We seek to enlist properties of dynamical manipulation in this program of reduced sensory and actuator complexity. Specifically, we are
adapting suggestive work by Atkeson and Schaal on the "Shannon Juggler" to the sensorless manipulation paradigm [?] in the specific context of parts feeding for flexible automation.

Sensorless rejection techniques remain the contemporary part orientation strategy of choice. These involve shaking, dropping, or pushing parts to reorient them; rejecting those still wrongly oriented; then re-circulating the rejected parts and repeating the cycle. Custom tooling for such feeders often makes them expensive and inflexible. Moreover, rejection techniques are very inefficient: for example, recent work by Brost [?] indicates the typical part will be randomly re-oriented into its desired pose only twenty percent of the time.

Since rejection techniques are inefficient and remote sensing and orientation techniques are often slow or expensive we propose to investigate a potentially alternative strategy. Namely, we propose a flat level 3 DOF vibratory table as a viable means of orienting pre-singulated parts. We seek to design a table motion which will cause all the parts on the table to asymptotically approach a known state without using sensory feedback.

The vibrational strategy should work by bouncing parts gently on the table vertically, while inducing momentary horizontal forces at the contact points which cause a torque to be applied to the center of mass. One would hope that if the vibrations are adequately designed, after a short period of time the parts will all rotate to a stationary pre-determined orientation. The shaking may then be stopped and the parts land in this known orientation. Different part geometries and material properties ought to require different vibration strategies. But changing the type of part should only require changing the vibration strategy in software: the lack of orienting devices mounted on the table should allow a wide range of parts to be oriented without changing the hardware. 10

3.3.2 Problem Statement

3.3.3 Present Capability

In a recently submitted publication [?] we have "solved" the one degree of freedom version of this problem depicted in Figure ???. Namely, we have shown that a table motion mimicking the trajectory of a massive (relative to the part) perfectly elastic (lossless collisions and flight) bouncing ball will force a (one degree of freedom) part dropped with arbitrary initial conditions toward a periodic vertical motion whose apex is governed by that of the table — providing the coefficient of restitution describing the collision is sufficiently small. Atkeson and Schaal had already given conditions for local asymptotic stability [?]. Yet while this and related versions of the "bouncing ball" problem have generated a voluminous literature [?], I am not aware of any previous results indicating the possibility of global attraction — a necessary feature of any orientation device that must succeed with all parts it encounters.

In the same paper we have described the results of some preliminary simulations of a three degree of freedom version of the problem depicted in Figure ??.

Finally, we are nearing the completion of our conversion of the planar juggling apparatus [?] into a test bed for the one and three degree of freedom problems discussed above. We have received a gratifying response from our initial contacts with several potential industrial partners. It seems clear that if the three degree of freedom experiments yield promising results then we will be able to use donated or consigned industrial equipment to actuate the full three degree of freedom table required to study the true multiple part six degree of freedom application. I have submitted a separate proposal this summer under the CIS5 instrumentation program that includes in the budget a high speed (220 Hz) camera system — if that proposal is awarded then we will have the sensory apparatus to study the results of our three degree of freedom actuation strategies.

3.3.4 Immediate Future Research Strategy

3DOF Sinusoidal Experiments:

10The three degrees of table freedom are each translational; rotational motion is not appropriate because the parts near the edges of the table would experience forces different from those in the center.
1DOF Sinusoidal Actuation:

3DOF “Opportunistic” Actuation: In all our experiments to date — batting balls around with a manipulator — the environment can be modelled as one or more independent point mass degrees of freedom. Recently we have begun to think about the problem of batting rigid bodies around although there is as yet only simulation data for this problem.

match the table's motion to the rigid body integral curves to simplify time-of-flight calculations

3DOF Sinusoidal Actuation:

3.5 Specific Questions to be Addressed

Can couple the vertical period-one with the transverse period-zero potential law because no obstacles are present. Need to define dynamical obstacle properly and find a means of (conservatively) analyzing sampled dissipative systems or replacing with a new dynamical obstacle avoider. to make the toss work.

Previously thought that mathematics literature on Invariant Manifold theory would not be appropriate since requires "small" perturbations from decoupled dynamics in order to apply. Now find it's working in horizontal hopper analysis

Can we couple two unimodal maps and tune various gaits by moving up and down the two different bifurcation diagrams?

3.6 The Virtues of a Simple Problem Domain

The work proposed here presents a setting simple enough to afford study of integrated approaches in a working system.

We add the dynamical component to this mixture for the reasons stated above. This setting also provides the benefit of admitting sufficiently simple configuration spaces that the fundamental questions may be considered. The dynamical pick and place problem may have some closer correspondence to applications than our original juggling work, but its central interest lies in its presentation of a nontrivial problem in a very stripped down setting. Even were one to doubt entirely the utility of dynamically dexterous robotics per se, it is a (still somewhat unappreciated) fact that these regimes involve simpler experimental testbeds and analytical approaches than the corresponding quasi-static situations. Physical models of point collisions and flight are far easier to get right 17, require fewer degrees of actuated freedom, and simpler sensors than corresponding

A number of authors have argued that dynamical dexterity contains the very origins of intelligence [?]

Thus, the proposed research may be seen as the effort to derive within the framework of dyn. sys. theory... conferring computational intelligence means extracting the symbols from the signals... natural partitions on space of systems well understood...natural partition on the space of states not well understood...hybrid systems

---

17Here, getting it right means to first approximation. The true mechanics of contact and compression and the aerodynamics of spin are as complex and as difficult to model as anything in the quasi-static domain. The difference seems to be that the crude models have greater predictive relevance when there is less contact.