11-1991

Can Dumb Feedback Produce Intelligent Machines?

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

Follow this and additional works at: http://repository.upenn.edu/ese_papers
Part of the Electrical and Computer Engineering Commons, and the Systems Engineering Commons

Recommended Citation

An invited morning lecture delivered at the advanced software technology and mechatronics research institute of Kyoto.
Post-IROS International Forum on Advanced Robotics.
NOTE: At the time of publication, author Daniel Koditschek was affiliated with Yale University. Currently, he is a faculty member in the Department of Electrical and Systems Engineering at the University of Pennsylvania.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ese_papers/694
For more information, please contact libraryrepository@pobox.upenn.edu.
Can Dumb Feedback Produce Intelligent Machines?

Abstract
An intelligent machine is autonomous. An autonomous machine can operate successfully in a diversity of situations without resort to intervention by "higher level" processes, for example, humans. Physical machines are ultimately force or torque controlled dynamical systems: the specification of input torques, whether via syntactic prescriptions or feedback controllers, results in certain classes of vector fields. Control procedures whose resulting vector fields have globally attracting goal states may properly be said to evince autonomous behavior. In this light it makes sense to "program" robots using the language of dynamical systems via feedback. This talk will review various procedures developed within the Yale Robotics Lab that result in provably autonomous behavior according to the criterion developed above. These procedures are expressed in a rudimentary "geometric programming language" appropriate to the domain of tasks being undertaken and result in closed loop dynamical systems with global convergence properties (often, the strongest that the topology of the program can allow). A variety of simulation results and physical experimental studies will attest to the practicability of these methods.

For more information: Kod*Lab

Disciplines
Electrical and Computer Engineering | Engineering | Systems Engineering

Comments
An invited morning lecture delivered at the advanced software technology and mechatronics research institute of Kyoto.

Post-IROS International Forum on Advanced Robotics.

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

An Invited Morning Lecture Delivered at the
Advanced Software Technology and Mechatronics Research Institute of Kyoto
Post-IROS International Forum on Advanced Robotics
November 6, 1991
Kyoto, Japan

Daniel E. Koditschek ¹
Center for Systems Science
Yale University, Department of Electrical Engineering

October 23, 1991

¹This work was supported in part by the National Science Foundation under a Presidential Young Investigator Award.
Abstract

An intelligent machine is autonomous. An autonomous machine can operate successfully in a diversity of situations without resort to intervention by “higher level” processes, for example, humans. Physical machines are ultimately force or torque controlled dynamical systems: the specification of input torques, whether via syntactic prescriptions or feedback controllers, results in certain classes of vector fields. Control procedures whose resulting vector fields have globally attracting goal states may properly be said to evince autonomous behavior. In this light, it makes sense to “program” robots using the language of dynamical systems via feedback.

This talk will review various procedures developed within the Yale Robotics Lab that result in provably autonomous behavior according to the criterion developed above. These procedures are expressed in a rudimentary “geometric programming language” appropriate to the domain of tasks being undertaken and result in closed loop dynamical systems with global convergence properties (often, the strongest that the topology of the problem can allow). A variety of simulation results and physical experimental studies will attest to the practicability of these methods.
# Contents

1 Introduction ................................................. 3

2 Task Encoding: Toward a Scientific Paradigm for Robot Planning and Control .......................... 3
   2.1 Desiderata ........................................ 3
   2.2 Representation ...................................... 4
      2.2.1 The Robot .................................... 4
      2.2.2 The Environment ................................. 5
   2.3 The Task ........................................... 5

3 The Navigation Problem .................................... 8
   3.1 Representation ...................................... 8
   3.2 Navigation Functions ................................. 8
      3.2.1 A Practicable Global Stability Mechanism .... 9
      3.2.2 Existence .................................... 9
      3.2.3 Invariance ................................... 9

4 The Construction of Navigation Functions ................. 10
   4.1 A "Model" Problem .................................. 10
   4.2 A Class of Coordinate Transformations .......... 10
   4.3 Navigation Functions for Geometrically Complicated Spaces .................. 12
   4.4 Navigation Functions for Multiple Robots ....... 13

5 An "Assembly" Problem ...................................... 16
   5.1 Representation ..................................... 16
      5.1.1 The Necessity of a Hierarchical Representation 16
      5.1.2 The Contact Set as a Higher Level Task Representation 17
      5.1.3 The Environmental Control System Arising from a Juggling Task 18
   5.2 A New Practicable Stability Mechanism .......... 20
      5.2.1 Encoding a Simple Juggling Task ............. 20
      5.2.2 The Stability Properties of Unimodal Maps .... 21
      5.2.3 A Practicable Global Stability Mechanism .... 21
      5.2.4 Invariance ................................... 22

6 A Continuous Controller for Robot Juggling ............ 22
6.1 The One Degree of Freedom Case ........................................... 22
  6.1.1 Construction .......................................................... 23
  6.1.2 Analysis ............................................................... 23
  6.1.3 Experiments .......................................................... 24
6.2 Generalization to More Degrees of Freedom .............................. 24
  6.2.1 Construction .......................................................... 24
  6.2.2 Implementation ..................................................... 25
  6.2.3 Experiments .......................................................... 25
6.3 Generalization to Other Tasks .............................................. 25

7 Conclusion ................................................................. 32
1 Introduction

Programming machines to operate flexibly and autonomously in the physical world seems to require a sophisticated representation that encodes simultaneously the nature of a task, the nature of the environment within which the task is to be performed, and the nature of the robot's capabilities with respect to both. We seek a scientific methodology of robot task encoding that encompasses the desired behavioral goals and environmental conditions as well. The methodology must balance the need for flexible expression of abstract human goals against the necessity of eliciting a predictable response from the commanded machine. This talk focuses on two problems of robot planning and control as examples of how we might propose to say what we mean to a robot and to know what we have said.

More broadly, this talk reviews a program of research in robotics that seeks to encode abstract tasks in a form that simultaneously affords a control scheme for these torque actuated dynamical systems as well as a proof that the resulting closed loop behavior will correctly achieve the desired goals. Two different behaviors that require dexterity and might plausibly connote "intelligence" — navigating in a cluttered environment, and juggling a number of otherwise freely falling objects — are examined with regard to similarities in problem representation, method of solution, and causes of success. The central theme of the paper concerns the virtue of global stability mechanisms arising from appropriately chosen feedback controllers. Feedback laws that specify torque inputs in response to sensory readings are written in the "machine language" of physical systems and offer the promise of provably correct automatic synthesis of machine behavior from abstract goals. Such event driven procedures enhance robustness — that is, experimentally observable resistance to unmodeled changes in the robot or its environment — at the execution level. Global stability, if achieved, lends autonomy — that is, freedom from dependence upon some "higher level" of intelligence — at the planning level. Since the resulting closed loop systems necessarily incorporate a parametrized model of the desired behavior, this approach to command and control encourages the design of "canonical" procedures for "model" problems which may then be instantiated in particular settings by a change of coordinates.

Against this theme is set the lingering question: isn't feedback simply too "dumb" to do the job? Perhaps more precisely, one worries whether all the benefits of efficiency and predictability outlined above simply come at too high a cost with respect to ease of expression. Is it not better to use a "high level" language wherein at least we think we know what we meant? On a practical level, the central theme of this talk might be seen as offering by concrete example a feeling for just how rich an expressive medium the language of dynamical systems may be.

2 Task Encoding: Toward a Scientific Paradigm for Robot Planning and Control

Roughly speaking, control theory has traditionally postulated a plant and a reference trajectory, and addressed the problem of how to force the plant to "track" the desired trajectory. Speaking even more roughly, artificial intelligence has traditionally postulated some abstract (generally human) goal and a symbol processing machine, and addressed the question of how to represent the goal within that symbolic system, sometimes considering the problem of realizing that goal symbolically as well. There is an obvious gulf between the (traditionally static) symbol system and (the usually physical) dynamical plant: roughly speaking, one must ask "where did the reference trajectory come from?"

2.1 Desiderata

Arguably, then, the business of "intelligent control theory" should concern the generation of goal representations that admit an automatic synthesis of controllers whose effect upon the plant is provably
correct. Thus, the term "task encoding" is intended to encompass

1. a programming language based upon the dynamical geometry of the plant and its environment;

2. a synthesis procedure capable of generating controllers for the plant automatically from the programming language;

3. a proof that the plant together with the controller effects the correct behavior within its environment;

4. a generalization scheme for re-using, in "equivalent" situations, task specifications and/or the controllers that result.

The burden of this talk concerns the extent to which feedback can achieve these criteria. We are able to encode an entire family of reference trajectories in the much more parsimonious form of a vector field. Considered as a "plax," the vector field has mathematical properties that permit rather close formal reasoning about its qualitative features. Considered as a "controller," the vector field has a natural relationship to the intrinsic dynamics of physical machines.

The work my students and I have completed in the field of robotics represents a very tentative and incomplete effort toward a theory of task encoding for a greatly restricted class of plants and environments. We have as yet no formal language for any setting. In some areas our correctness proofs lag far behind our experimental observations, in others our proofs require such unrealistic assumptions that no fruitful experiments have yet been attempted. Our notions of generalization remain narrowly specialized. Nevertheless, the program of task encoding, however ambitious or even, possibly, wrong-headed, seems to represent one of the few clearly articulated statements of what it might mean to have a scientific program of research in "intelligent control."

2.2 Representation

One of the reasons that the program outlined above is difficult to put into practice is the general absence of models that are sufficiently accurate to represent the phenomena of interest but sufficiently simple to admit some hope of subsequent analysis. Nevertheless, it seems inescapable that the robotic problem may at some reasonably effective level of abstraction be cast in the following form. There is a configuration space, $Q = \mathcal{R} \times \mathcal{B}$, a compact connected manifold with boundary. It arises from various unactuated "bodies" with generalized coordinates $b \in \mathcal{B}$ subjected to the forces of manipulation imposed by various actuated "robots" with generalized coordinates $r \in \mathcal{R}$. Since the robots and bodies are physical systems, their motions are governed by a Lagrangian system,

$$M(r, b) \begin{bmatrix} \dot{r} \\ \dot{b} \end{bmatrix} + C(r, b, \dot{r}, \dot{b}) \begin{bmatrix} \dot{r} \\ \dot{b} \end{bmatrix} = \tau(r, b, \dot{r}, \dot{b}) + \begin{bmatrix} u \\ 0 \end{bmatrix}$$ (1)

where $M$ represents inertial coupling, $C$ represents the fictitious forces due accelerating frames, and $\tau$ represents the various external and self-generated forces to which these masses are subject. Here, $u$ denotes the forces imposed upon the system by actuators that we presume to be completely specified by the user — that is, $u$ represents the controlled input to the system. This second order equation of course might be re-expressed as a first order control system on the phase space, $TQ$, whereupon we obtain a controlled dynamical system.

We now sketch the origins of this model.

2.2.1 The Robot

Key to this point of view is the fact that any machine operating in the physical world is subject to both dynamical as well as geometric constraints. Kinematic chains impose a conceptually straightforward [21]
but mathematically complicated [48] geometry. Their Newtonian dynamics result in strongly nonlinear and surprisingly complex equations [4]. Yet since these constraints have been understood and satisfactorily modeled for a long time, well posed control problems for robot "plants" may be stated relatively easily. For purposes of the present paper, take the standard rigid body model of the "plant",

$$M(r)\ddot{r} + C(r, \dot{r})\dot{r} = u,$$

(2)

where $r$ is a vector of joint measurements, $u$ is a vector of control torques or forces exerted on each joint by the actuators, $M$ arises from the "inertial" properties of the links, $C$, from the coriolis and centripetal forces of motion (and the gravitational forces we assume to have been canceled by $u$). This accounts for the first component in the vector equation (1).

2.2.2 The Environment

In all but the most trivial instances, the robot's desired behavior involves interaction with an environment, $E$, that must itself possess geometric and dynamical properties. Moreover, in the context of particular tasks, various aspects of the robot's operation in the environment will give rise to a new set, $P$, that might be called the "planning set", within which particular goals may be formally represented. For example, in the case of robot motion planning, the environment's geometry together with the robot's kinematics and geometry give rise to a freespace [34] within which the robot's motions are constrained to lie. For peg-in-hole [56] and related problems of pushing [36, 35] additional structure must be added to the geometry of this space [8, 17] to account for frictional and jamming (contact degrees of freedom) reaction forces that the environment may impose upon the robot. This accounts for the appearance of the function, $\tau(r, q, \dot{q})$ in (1). The planning space might be taken as $TQ \times T^*Q$ in such situations.

In more general situations there will be degrees of freedom, $b$, that are not directly coupled to an independent actuator. When these arise from freely moving "bodies," defining the appropriate planning space may become rather subtle. Tasks such as playing ping-pong [2] or walking and running [41, 37] require explicit attention to the dynamics as well as the geometry of the environment. Thus, the environment, $E$, is the source of the entire second component in equation (1) which still inhabits the second tangent bundle, $TTQ$. The nature of the appropriate planning space for such problems will become clear in the sequel.

2.2.3 The Task

Finally, a robot operating in a specified environment might be assigned a variety of tasks. The specific task desired — an abstraction meaningful initially only to its human originator — must be encoded in terms that relate to the robot in its environment. Thus, within the context of the planning set there must be devised a formal representation of the desired behavior — the "encoding". In the two example task domains examined here the encoding takes the form of a set of goal states, $G \subset P$ to which it is desired to bring the robot-environment pair.

Ultimately, in order to achieve the task, some control, $u$, must be specified for the "plant" (1): it must ensure simultaneously that the robot both achieve the task as well as respect the environmental constraints. As the robot's (and, possibly, the environment's) state changes under the action of this controller, the planning space reflects these changes. Apparently, then, the solution to a robotic task imposes a dynamical system upon $P$. When, as here, the task at hand admits the representation in the form of a goal state(s), $G$, then a successful control scheme is one whose associated dynamics on $P$ brings as many initial states to $G$ as possible.
2.3 Intelligent Robots and Intelligent Designers

2.3.1 Intelligent Behavior Connotes Autonomy

Having concluded that the robot's execution of a task in a specified environment may be represented as a dynamical system on an appropriate space, and that the criterion of success is the achievement of some distinguished goal set in that space, one is in a position to assess the "intelligence" of the resulting behavior with respect to standard ideas from dynamical systems theory. This paper holds autonomy to be a primary design objective in the construction of intelligent machines. A machine equipped with an intelligent strategy ought be able to contend with the full spectrum of logically possible circumstances that arise in completion of its task.

Translated into the context of dynamical systems theory, autonomy implies global convergence. That is, an intelligent control strategy ought to be capable of bringing the robot-environment pair to the goal states, \( \mathcal{G} \), from any initial state represented in the planning space. More succinctly, \( \mathcal{G} \) ought to be an attracting set whose domain of attraction is \( \mathcal{P} \). Unfortunately, this is not always possible: Section ?? describes a situation where such global properties are topologically impossible. Instead, one might imagine a situation where "almost all" initial states are brought to the desired goal, and what is left over is very small. If the domain of attraction of a locally attracting set, \( \mathcal{G} \), includes all but a set of measure zero then say that \( \mathcal{G} \) is essentially globally asymptotically stable. Even where no topological obstructions are present, and even if one settles for essential global properties it is an unfortunate fact that estimating the domain of attraction of locally attracting sets is very difficult in practice.

For linear dynamical systems on a vector space, a local computation involving the eigenvalues of a matrix affords global conclusions. This is the archetypally "practicable" means of assuring global properties, and is almost by definition not to be found in general in the nonlinear case. The two examples presented in this paper, however represent instances where a global stability mechanism does enjoy such a practicable property: namely, a series of locally defined computations involving jacobians and their eigenvalues.

2.3.2 Intelligent Design Connotes Generalization

If global stability properties are a primary objective in the design of intelligent robot controllers, yet practicable instances of such mechanisms are rare, then an intelligent designer will seek to use and re-use existing instances again and again. A second theme of this paper is the utility of generalizing a specific controller design through a change of coordinates or "lifting" into higher dimension. The two behaviors reported below are achieved by recourse to two different stability mechanisms which share the unusual property of practicability in the sense developed above. In each case, the paper attempts to show how a canonical solution to a simple problem can be "deformed" into the solution to a seemingly complicated but essentially equivalent problem via the appropriate transformation of the problem space. The nature of the "lifts" will be noted in passing.

2.4 A Program of Robotics Research

It is not at all clear how to tell a robot to "fold the laundry" or "scramble the eggs" or "make the bed". For such tasks neither the environment, \( \mathcal{E} \), nor the appropriate planning space, \( \mathcal{P} \), nor the task encoding, \( \mathcal{G} \), seem very obvious. The program of research reviewed in this paper seeks to make progress toward the analysis and achievement of such confusing robotic tasks by a methodical investigation of more straightforward examples. For example, navigation tasks admit a relatively straightforward representation — a distinguished destination point within the free space. Juggling, hopping, walking, and other such rhythmic behaviors may be readily represented by the specification of a distinguished periodic trajectory in the robot's phase space [11, 37]. It should be immediately emphasized that the only thing straightforward
about these examples is the conceptual distinction between task, environment, and robot. Navigation has been shown to be fundamentally difficult [49], and juggling has not even been attempted until recently [10, 1]. Yet in these cases it has seemed sufficiently clear how to disentangle the constituent pieces of the problem definition that a careful look at how they interact in a successful implementation might provide more general insight into the problem of task encoding.
3 The Navigation Problem

Let a robot move in a cluttered but perfectly known workplace. There is a particular location of interest and it is desired that the robot move to this location from anywhere else in the workplace without colliding with the obstacles present. This "simple problem" takes its place in the representation introduced above as an instance of (1) where \( Q = R \) — that is, there are no moving bodies except the robot(s) so every degree of freedom is fully actuated, and the difficulty arises purely in consequence of the complexity of the configuration space, \( R \).

3.1 Representation

The constituent pieces of the problem seem readily apparent in this case. The robot model has already been introduced. The environment, \( E \), is simply the workplace — a subset of Euclidean 3-space remaining after the obstacles are removed. Contained within the robot's configuration space is the free space, \( F \) — the set of all robot placements which do not involve intersection with any of the "obstacles" cluttering the workplace. The appropriate planning set, for this problem is now clear: it is the phase space, \( P = TF \) formed over \( F \), that is, the union of all the robot's configuration space velocity vectors taken over each configuration in \( F \). For present purposes this may be modelled as a smooth manifold with boundary (but see [43] for the case of sharp corners). The task also seems straightforward to represent: a particular navigation problem results from the choice of one particular destination point in the interior of the freespace. The goal set, \( G \), is a singleton: the destination point at zero velocity. The problem is now to find a feedback controller under whose influence the robot's state will approach \( G \) from as large as set of initial configurations as possible while remaining in \( P \).

A few caveats are in order before proceeding. First, it is entirely likely that the robot's freespace is not connected — that is, there may be no collision free path from some legal configurations to the destination. In the more traditional version of this problem, the navigation problem includes the decision task of whether a particular initial configuration is in fact included in the same connected component of \( F \) as the destination. In the present formulation the robot must arrive (with probability one) at the goal if a path exists. Thus one can only conclude (with probability one) that no such path exists for a particular initial configuration only after the robot's motion under the controller ceases at some spurious location. Second, a constructive representation of the planning space, \( P \), may be very difficult to obtain in practice, even when \( E \) is perfectly known (which, of course, it might not be in the real world). Yet this work presumes that exact information concerning the boundary components of \( P \) is available.

3.2 Navigation Functions

Motivated by Lord Kelvin's assurance that dissipative mechanical systems end up at the local minima of the potential field, a great deal of interest in robotics has centered around the construction of artificial potential fields to encode navigation problems. Initiated by Khatib a decade ago [22], the idea of using artificial potential functions for robot task description and control was adopted or re-introduced independently by a number of researchers [38, 3, 40]. Since the interest in artificial potential functions originally emerged within the robotic control community, it is perhaps not surprising that little attention was paid to the algorithmic issues of global path planning in this literature. The question of whether the method could be used to guarantee the construction of a path between any two points in a path-connected space remained unexplored. Yet it is exactly this kind of global property that would lend autonomy from "higher level" intelligence to the controller.
3.2.1 A Practicable Global Stability Mechanism

In the present context, the utility of artificial potential functions for path planning rests upon the pos-
sibility of deducing global stability properties from local computations. Because the potential function
serves as a global Lyapunov function for its gradient vector field, it is easy to see that the minima of a
gradient system (that satisfies certain regularity conditions) will attract almost all trajectories [20, 27].
Of course, the condition for a minimum is a local one that may be constructively checked via calculus and
algebraic computation. Thus, if it can be assured that there is only one minimum and that it coincides
with the desired destination then a potential function serves as a global path planner on the freespace, \( \mathcal{F} \). Of course, the appropriate planning space is \( \mathcal{P} \), the space of legal configurations and all their possible
velocities. But a slight extension to Lord Kelvin's century old results on energy dissipation suffices to
make the same machinery work with a suitably designed controller for the robot on \( \mathcal{P} \) [27].

3.2.2 Existence

Gradually, there seems to have emerged a common awareness of several fundamental problems with the
potential function methodology First, researchers inevitably discovered through simulations or actual
implementations that progressive summation of additional obstacles often lead to spurious minima and
their accompanying local basins of attraction into which the robot would generally "stall out" long before
achieving the desired destination. Second, the infinite value of the artificial potentials required to prevent
trajectories of the ultimate mechanically controlled system from crashing through obstacle boundaries
obviously could not be achieved in the physical world and there were no clear guarantees as to when
the saturation torque levels of the robot's actuators would indeed suffice to prevent collisions. Thus, an
artificial potential function need satisfy a list of technical conditions in order to give rise to a bounded
torque feedback controller that guarantees convergence to the goal state, \( \mathcal{G} \), from almost every initial
configuration. This list comprises comprises the notion of a navigation function introduced to the
literature two years ago [42].

The question immediately arises whether such desirable features may be achieved in general. In fact,
the answer is affirmative: smooth navigation functions exist on any compact connected smooth manifold
with boundary [32]. Thus, in any problem involving motion of a mechanical system through a cluttered
space (with perfect information and no requirement of physical contact) if the problem may be solved
at all, we are guaranteed that it may be solved by a navigation function. There remains the engineering
problem of how to construct such functions.

3.2.3 Invariance

The importance of coordinate changes and their invariants is by now a well known theme in control theory.
Roughly speaking, these notions formalize the manner in which two apparently different problems are
actually the same. Their most familiar instance is undoubtedly encountered in the category of linear
maps on linear vector spaces whose invariants (under changes of basis) determine closed loop stability.
Of course, many other instances may be found in the control literature and, more recently, the utility of
coordinate changes in robotics applications has been proposed independently by Brockett [6] as well.

The relevant invariant in navigation problems is the topology of the underlying configuration space
[26]. In this regard, the significant virtue of the navigation function is that its desirable properties are
invariant under diffeomorphism [32]. Thus, instead of building a navigation function for each particular
problem, we are encouraged to devise "model problems", construct the appropriate model navigation
functions, and then "deform" them into the particular details of a specified problem.
4 The Construction of Navigation Functions

4.1 A "Model" Problem

A "Euclidean sphere world" is a compact connected subset of $\mathbb{E}^n$ whose boundary is the disjoint union of a finite number, say $M + 1$, of $(n - 1)$-spheres. We suppose that perfect information about this space has been furnished in the form of $M + 1$ center points $\{q_i\}_{i=0}^M$ and radii $\{\rho_i\}_{i=0}^M$ for each of the bounding spheres. There are two new ideas in our artificial potential function construction. First, we avoid spurious minima by multiplying the constituent functions together rather than summing them up. Namely, the "bad" set of obstacle boundaries to be avoided is encoded by the product function, $\beta : \mathcal{M} \to [0, \infty)$ is,

$$\beta \triangleq \prod_{i=0}^M \beta_i,$$

where

$$\beta_0 \triangleq \rho_0^2 - \|q\|^2, \quad \beta_j \triangleq \|q - q_j\|^2 - \rho_j^2, \quad j = 1, \ldots, M$$

are the outer boundary and inner obstacle functions, respectively. The good set, the desired destination, $q_d$ is represented by an ordinary Hook's Law potential, $\gamma \triangleq \|q - q_d\|^{2k}$, raised to an even power and the rough syntax "go to $\gamma = 0$ and do not go to $\beta = 0$" is encoded by the intuitively obvious product

$$\phi \triangleq \frac{\gamma}{\beta}.$$

Of course, $\phi$ is unacceptable since it is unbounded. The second new idea at work is to produce a bounded potential and gradient by a smooth "squashing" function,

$$\sigma(z) \triangleq \frac{z}{1 + z}.$$

Note that the composition

$$\sigma \circ \phi = \frac{\gamma}{\gamma + \beta}$$

is everywhere smooth and bounded, and attains its maximal height of unity only on the boundary components of the configuration space. For technical reasons we find it necessary to take the $k^{th}$ root of this ratio with the following result.

**Theorem 1 ( [32])** If the configuration space, $\mathcal{J}$, is a Euclidean sphere world then for any finite number of obstacles, and for any destination point in the interior of $\mathcal{J}$,

$$\phi = \sigma_d \circ \sigma \circ \phi = \left(\frac{\gamma^k}{\gamma^k + \beta}\right)^{\frac{1}{k}},$$

has no degenerate critical points and attains the its maximal value of unity on the boundary, $\partial \mathcal{J}$. Moreover, there exists a positive integer $N$ such that for every $k \geq N$, $\phi$ has one and only one minimum on $\mathcal{J}$.

The function, $N$, on which the theorem depends is given explicitly in [32].

4.2 A Class of Coordinate Transformations

A star shaped set is a diffeomorph of a Euclidean $n$-disk, $\mathbb{D}^n$ possessed of a distinguished interior center point from which all rays intersect its boundary in a unique point. A star world is a compact connected subset of $\mathbb{E}^n$ whose boundary is the disjoint union of a finite number of star shaped set boundaries. Now
Figure 1: planar sphere world with nine internal obstacles [46]. Contour lines denote the level curves of a navigation function constructed according to Theorem 1.

suppose the availability of an implicit representation for each boundary component: that is, let \( \beta_i \) be a smooth scalar valued function that is positive outside, negative inside, and vanishes on the boundary of the \( i^{th} \) obstacle. Assume, moreover, that a known center point location, \( q_j \), has been specified for each obstacle as well. Further geometric information required in the construction to follow is detailed in the chief reference for this work [44]. A suitable Euclidean sphere world model, \( M \), is explicitly constructed from this data. That is, one determines \((p_j, \rho_j)\), the center and radius of a model \( j^{th} \) sphere, according to the center and minimum "radius" (the minimal distance from \( q_j \) to the \( j^{th} \) obstacle) of the \( j^{th} \) star shaped obstacle.

A transformation, \( h : M \rightarrow F \), may now be constructed in terms of the given star world and the derived model sphere world geometrical parameters as follows. Denote the "\( j^{th} \) omitted product", \( \Pi_{j=0}^M \beta_j \) as \( \beta_j \). The "\( j^{th} \) analytic switch", \( \sigma_j \in C^\omega[F, \mathbb{R}] \),

\[
\sigma_j(q, \lambda) \triangleq \frac{x}{\sigma} \cdot \frac{\gamma \beta_j}{\beta_j} = \frac{\gamma \beta_j b + \lambda \beta_j}{\gamma \beta_j b + \lambda \beta_j},
\]

(where \( \lambda \) is a positive constant) attains the value one on the \( j^{th} \) boundary and the value zero on every other boundary component of \( F \). The "\( j^{th} \) star set deforming factor", \( \nu_j \in C^\omega[F, \mathbb{R}] \),

\[
\nu_j(q) \triangleq \rho_j \frac{1 + \beta_j(q)}{\|q - q_j\|},
\]

scales the ray starting at the center point of the \( j^{th} \) obstacle, \( q_j \), through its unique intersection with that obstacle's boundary in such a way that \( q \) is mapped to the corresponding point on the \( j^{th} \) model obstacle — a suitable sphere. The overall effect is that the complicated star shaped obstacle is is "deformed along the rays" originating at its center point onto the corresponding sphere in model space.

The star world transformation is now given as

\[
h_\lambda(q) \triangleq \sum_{j=0}^M \sigma_j(q, \lambda) \left[ \nu_j(q) \cdot (q - q_j) + p_j \right] + \sigma_d(q, \lambda) \left[ (q - q_d) + p_d \right],
\]

where \( \sigma_j \) is the \( j^{th} \) analytic switch, \( \sigma_d \) is defined by

\[
\sigma_d \triangleq 1 - \sum_{j=0}^M \sigma_j,
\]
and $\nu_j$ is the $j^{th}$ star set deforming factor. The "switches", make $h$ look like the $j^{th}$ deforming factor in the vicinity of the $j^{th}$ obstacle, and like the identity map away from all the obstacle boundaries. With some further geometric computation we are able to prove the following.

**Theorem 2** ([44]) *For any valid star world, $\mathcal{F}$, there exists a suitable model sphere world $\mathcal{M}$, and a positive constant $\Lambda$, such that if $\lambda \geq \Lambda$, then

$$h_\lambda : \mathcal{F} \rightarrow \mathcal{M},$$

is an analytic diffeomorphism.*

Thus, if $\varphi$ is a navigation function on $\mathcal{M}$, the construction of $h_\lambda$ automatically induces a navigation function on $\mathcal{F}$ via composition, $\tilde{\varphi} \triangleq \varphi \circ h_\lambda$.

![Figure 2: Planar star world with nine internal obstacles [46]. The contour lines are level curves of a navigation function induced by diffeomorphism according to Theorem ??, modified to take account of the "sharp corners" [55]. The model sphere world is depicted in the previous Figure 1.](image)

### 4.3 Navigation Functions for Geometrically Complicated Spaces

In a recent paper [43], we show how to extend significantly this class of coordinate transformations. Briefly, consider an obstacle $\mathcal{O}_i$ which is a union of several intersecting stars. The arrangement of the stars in $\mathcal{O}_i$ can be partially described by a graph. If the graph is a tree, and the geometric arrangement of the daughters to the parent stars in the tree satisfies certain other regularity assumptions [43] then say that the obstacle is a *tree of stars*; a *forest of trees of stars* is a freespace, $\mathcal{F}$, consisting of the disjoint union of a finite number of trees of stars. It can be shown that any deformed sphere world can be approximated arbitrarily closely by a suitable forest of trees of stars [46]. Such a forest, $\mathcal{F}$ has a *purged version*, $\tilde{\mathcal{F}}$, defined to be $\mathcal{F}$ with the leaves in trees consisting of more than one star filled-in and "reattached" to $\mathcal{F}$. Using the ideas presented above, we have shown how to define a change of coordinates from any forest, $\mathcal{F}$, to its purged version, $\tilde{\mathcal{F}}$, [43]. Successive purged versions of a forest result eventually in a star world. Thus by composing successively such "purging transformations", we change coordinates from the original forest of trees of stars to a star world on which a navigation function can be constructed as described above.

Figure 3 depicts a two-dimensional (the results, of course, work in arbitrary dimensions) forest of stars resembling a building floor plan. There are three internal tree-like obstacles, and the depth of the deepest tree is $d = 4$. According to the method described above, the purging transformation, $f_{\Lambda+1}$, is
applied \( d \) times, until a space whose obstacles are the roots of the original trees is obtained. This space is a star world: the the previously constructed star-world to sphere-world transformation \([44]\) may now be used to to obtain the corresponding model sphere world, \( \mathcal{M} \), on which the simple navigation function may be used.

![Figure 3: Planar forest of stars with three internal tree-like obstacles (bottom right), its "purged" versions, and its model sphere world (top left). \([43]\)](image)

### 4.4 Navigation Functions for Multiple Robots

In a recent set of papers \([55, 54]\) we have extended the navigation work to problems involving multiple moving bodies. Figure 4 provides an illustration of the idea. Each of \( n \) balls, which are free to move, is uniquely specified by its position \( b_i \in \mathbb{R}^2 \), radius \( \rho_i \in \mathbb{R}^+ \), and the composite vector of \( n \) balls is \( b \in \mathbb{R}^{2n} \). Label the desired ball positions \( d_i \in \mathbb{R}^2 \) and \( d \in \mathbb{R}^2 \) respectively. We necessarily assume both the initial and desired states are "legal" — no balls touch and all reside within the boundary. Figure 4 shows a typical 7 ball motion sequence from (random) initial configuration to "assembled" final configuration.

A function, \( \gamma \), is easily constructed according to the standard recipe for such geometrically simple environments, \( \mathcal{E} \), given in (3) where, in this case, the function \( \gamma(b) \) is given by

\[
\gamma(b) = \sum_{i=0}^{i=n} \gamma_i(b); \quad \gamma_i(b) = \frac{1}{2} ||b_i - d_i||^2
\]  

(6)
Figure 4: A Seven Ball Assembly Sequence: Balls indicate current configuration $b$, Lines indicate the ball paths. Frame (i) shows random initial configuration, (ii) and (iii) are intermediate configurations, and (iv) shows final (desired) configuration.

and $\beta(b)$ is given by

$$\beta(b) = \prod_{\substack{i=1 \atop i \neq j}}^{n} \beta_{b_i, b_j}(b)$$  \hspace{1cm} (7)

where the $\frac{\varphi}{2}(n - 1)$ functions

$$\beta_{b_i, b_j}(b) = \frac{1}{2} (||b_i - b_j||^2 - (\rho_i + \rho_j)^2)$$  \hspace{1cm} (8)

are always positive when the balls are not touching, and zero when ball $i$ and $j$ touch. According to our rather extensive simulation studies the performance of this construction is surprisingly good — for example $\varphi$ can be easily “tuned” so that the arclength of the typical path chosen approaches merely 1.25 that of the straight line Euclidean distance to the goal state in the configuration space (physically unrealizable, in general, because of collisions between bodies) [55]. We presume that a deformation theory similar to that discussed above for the topology of the sphere world will be forthcoming in this very different setting as well.

We surmise, but have not yet proven that this simple construction remains a valid navigation function in the radically different topology of the configuration space arising from Figure 4. We have shown [24] that this construction is indeed a navigation function for high enough values of $k$ in analogous when the dimension of the ambient space is one, for example, as depicted by the top line (a one degree of freedom environment) in Figure 5.
Figure 5: A One Degree of Freedom Dual Assembly Problem
5 An "Assembly" Problem

Assembly problems require that a robotic system with a few actuated degrees of freedom manipulate an environment with a greater number of unactuated degrees of freedom. For example, while the task depicted in Figure 4 has greater dimensional complexity than that of Figure 5, the latter introduces a heretofore unexplored problem. This, of course, is the implication that while the robot on its parallel axis is actuated by a force that is under our control, the bodies cannot be independently actuated unless the robot approaches them and engages them with its gripper. Such a set of hypotheses brings forth the full generality of (1).

One could introduce the same problem into the setting of Figure 4 by removing the assumption that the bodies are actuated, and hypothesizing some actuated machine inhabiting a parallel plane to that depicted, from which the passive bodies may be engaged when the robot is close enough. It would be still of greater interest to consider the situation wherein this actuated machine actually inhabits the same plane as all the other \(n-1\) balls. We shall consider a very special instance of this last problem now.

Since the dynamical coupling between degrees of freedom in this setting is a function of their relative configuration, the motion of such systems is subject to constraints that preclude smooth feedback stabilization. In other words, in contrast to the problem of purely geometric motion described above, task encoding for assembly defies a straightforward application of the planning and control paradigm outlined up until now. Nevertheless, passage to an abstracted "planning system" — a discrete dynamical controlled process that takes place on the contact set — affords a similar procedure.

5.1 Representation

There are many different models of robot-body coupling forces, \(\tau\) in (1). For purposes of this talk, we shall limit our consideration to a dynamical setting of the assembly problem assembly where these interactions connote in their simplest instance contact conditions between body and robot that are of extremely short duration and involve very large forces. In the limit, such impact conditions lead to the notion of impulses introduced to maintain inequality constraints in configuration [23]. The passage to impulses leads to a new planning set, \(\mathcal{P}\), that represents the state of the unactuated bodies just at the point of impact with the actuated robot.

5.1.1 The Necessity of a Hierarchical Representation

Transferring a system of bodies possessing a greater number of degrees of freedom than the number of actuated degrees of freedom available from one configuration to another is an assembly problem. Note that the possibility of solving such problems depends on the fact that the dynamical coupling between degrees of freedom is not fixed. More precisely, the allowable infinitesimal motions of the body system differ from configuration to configuration. Thus, since the motion of the bodies is subject to constraints that are not integrable, assembly problems give rise to nonholonomically constrained control systems. These seem to differ from the more commonly studied nonholonomic robotics problems involving steering wheels [39] or coupled vehicles [33] in that here the tangent bundle constraints are (at best) \(C^\infty\) rather than analytic. They seem to be very similar in most other respects, and there is good reason to hope that a principled treatment of one class should shed light on the other. One significant feature of such problems is that while the control systems that arise are completely controllable, Brockett's [7] stabilizability condition is not met [8], and there is no possibility of constructing a single smooth feedback controller to stabilize an isolated equilibrium state.

Contact forces, \(c(\delta, \delta')\), depend upon the relative configuration, \(\delta(b, r)\). Roughly speaking, they obtain when the robot and the body approach each other in close proximity, and we will suppose that \(\delta\) is a smooth function that vanishes on some portion of the contact set, \(\partial Q\). For example, if body 1 is actuated,
\( b_1 = r \) and all others are unactuated in Figure 4, then we might take
\[
\delta(r, b) = \prod_{j=2}^{n} \beta_{r, b_j}(r, b)
\]
from (7). Given some small "contact region", \( \epsilon > 0 \), we will require
\[
c(\delta, v) \begin{cases} 
> 0 & 0 < \delta < \epsilon, v < 0 \\
< 0 & -\epsilon < \delta < 0, v > 0 \\
= 0 & \text{otherwise}
\end{cases}
\]
according to the arguments of [23, §2.3.B.ii.c]. Clearly such a function cannot be analytic, but it is expedient to assume smoothness in order to avoid technical issues of existence and uniqueness in the sequel. \(^1\)

As matters stand, this crude model does not prohibit the possibility that sufficiently large and opposing velocities might give rise to trajectories whose relative configuration changes sign. Such trajectories would violate the physical principle that two bodies cannot occupy the same space at the same time. This principle may be enforced by letting \( c \) become unbounded as \( \delta \to 0 \), or, effectively the same, assuming that the initial kinetic energy of the joint system is limited (and that \( c \) does not add energy to the system) as a function of the maximal value of \( c \).

Suppose the contact forces can be turned on or off with a "gripping" function, \( u_2 \), and that the robot is completely actuated by a bounded force, \( u_1 \). Then the equations of motion (1) in the simple situation discussed here take the form
\[
\begin{align*}
\ddot{r} &= -u_2 c + u_1 \\
\dot{b} &= u_2 c
\end{align*}
\]
or, if \( \mathbf{x} \triangleq [r, \dot{r}, b, \dot{b}]^T \),
\[
\dot{x} = f(x); \quad f = \begin{bmatrix} x_2 \\
-u_2 c(x) + u_1 \\
x_4 \\
u_2 c(x)
\end{bmatrix}
\]
Notice that \( c \) vanishes when \( \delta > \epsilon \), thus the last entry of \( f \) is zero except in certain regions that are very close to the boundary of the configuration space. It follows that \( f \) cannot be locally surjective on any neighborhood of a point in the complement of this "collar" around the contact set, hence, that no equilibrium state therein may be stabilized by a continuous feedback law, \( u = g(x) \) [7]. In other words, there is no way of "shaping" \( f \) into a single vector field that would make an arbitrary interior point both attracting and stable. However, we may achieve the former at the expense of the latter by adopting a hierarchical control strategy that replaces with a more "abstracted" representation, the detailed continuous dynamics of contact [24].

5.1.2 The Contact Set as a Higher Level Task Representation

A unifying theme that has emerged from our previous work as a means of surmounting the difficulties described in the discussion above concerns the role of contacts
\[
\mathcal{C} \triangleq \{(r, b) \in \partial \mathcal{Q} : \delta(r, b) = 0\}
\]
in organizing the design and informing the analysis of hierarchical feedback controllers for autonomous manipulation for assembly. Roughly speaking, we are led to redefine a task in terms of the body

\(^1\)The function cannot even be differentiable at the origin, but it will prove necessary to avoid this regime anyway through the methods below.
configuration variables, and attempt to measure the progress toward the goal with respect to a discrete system whose time steps are punctuated by events at the contact set. The function of the continuous dynamical controllers, \( u \) in (1), is merely to achieve a continual return to the contact set in such a fashion that the next event is associated with this version of “progress.”

In previous work, we have encountered three concrete instances of how to define a consistent notion of an “event” with respect to the contact set. The first was contributed by Raibert [41], who realized that an easy way to achieve hopping or running would be to readjust the body’s energy level during stance phase (that is, on the contact set) to which a “return” is always assured by the presence of gravity [25]. The second obtains from our on-going studies of juggling that will form the chief source of examples in the sequel [14, 13, 47]. Here, the body’s energy level during impact (that is, on the contact set) is adjusted by choice of the robot’s motion preceding the impact. The return is again assured by the earth’s gravitational field. The last obtains from our initial work in assembly [24, 55], wherein a navigation function is used as the measure of progress, and the return to the contact set must be accomplished actively by the controller.

In the juggling and hopping studies, this formulation amounts to an appeal to \textit{return maps} on a Poincaré section and is quite conventional within the dynamical systems literature [18]. However, while a robot that brings a disorganized collection of bodies into an assembly must necessarily visit and revisit the contact set until the assembly is complete, no periodic phenomenon is natural to the problem. The recognition that the nonperiodic nature of an assembly task may still be amenable to a treatment resembling return maps on sections represents a considerable departure. It is a straightforward matter in discrete linear systems theory to build “deadbeat controllers” whose closed loops converge in finite time; one simply places all poles at the origin of the complex plane. The algebraic properties of nonlinear discrete systems with finite time convergence may be far less easily characterized, but the concept is the same.

Since the juggling work introduces a new practicable stability mechanism, we shall use this as the example of assembly in the rest of the paper.

5.1.3 The Environmental Control System Arising from a Juggling Task

In the following discussion, consider as a concrete instance of this problem the setting depicted in Figure 6.

![Figure 6: The Yale Spatial Juggler](image)

Figure 6: The Yale Spatial Juggler
The two properties of a body relevant to juggling are its flight dynamics (behavior while away from the paddle), and its impact dynamics (how it interacts with the paddle/robot). For simplicity, model the ball’s flight dynamics as a point mass under the influence of gravity. This gives rise to the flight model

$$\dot{w} = F(w); \quad w = \begin{bmatrix} \dot{b} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \end{bmatrix}; \quad F(w) = \begin{bmatrix} \dot{w}_2 \\ \ddot{w}_2 \end{bmatrix};$$

where $b \in \mathcal{B} = \mathbb{R}^2$, and $\ddot{a} = (0, 0, -\gamma)^T$ is the acceleration vector experienced by the ball due to gravity. A position-time-sampled measurement of this system will be described by the discrete dynamics,

$$w_{j+1} = F_s(w_j) \triangleq A_s w_j + a_s; \quad A_s \triangleq \begin{bmatrix} I & sI \\ 0 & I \end{bmatrix}; \quad a_s \triangleq \begin{bmatrix} \frac{1}{2} s^2 \ddot{a} \\ s \ddot{a} \end{bmatrix}; \quad C = [I, 0],$$

where $s$ denotes the sampling period, and $w_j \in TB$.

Suppose a ball with trajectory $b(t)$ collides with the paddle in robot configuration $r \in \mathcal{R}$ at some point, $p$ on the paddle which has a linear velocity $v$. We seek a description of how the ball’s phase, $(b, \dot{b}) \in TB$ is changed by the robot’s phase, $(r, \dot{r}) \in TR$ at an impact.

As in [14, 13, ?] we will assume that the components of the ball’s velocity tangent to the paddle at instant of contact are unchanged, while the normal component is governed by a simple impact law. To achieve the greatest simplification it proves useful to introduce a new force model. In the limit, as $|\gamma| \to \infty$, the time over which the contact forces are engaged becomes vanishingly small, and the integral of work done remain a finite value called the “impulse” [23, §2.5]. Energy is generally lost (the contact forces are not conservative) during the impact, and this leads to the notion of a coefficient of restitution [23, §4.2]. For some $c \in [0, 1]$ this impact model can be expressed as

$$(\dot{b}_n' - \dot{v}_n') = -c(\dot{b}_n - \dot{v}_n),$$

where $\dot{b}_n'$ and $\dot{v}_n'$ denote the normal components of the ball and paddle velocities immediately after impact, while $\dot{b}_n$ and $\dot{v}_n$ are the velocities prior to impact. Assuming that the paddle is much more massive than the ball (or that the robot has large torques at its disposal), we conclude that the velocity of the paddle will remain constant throughout the impact ($v' = v$). It follows that the coefficient of restitution law can now be re-written as

$$\dot{b}_n' = \dot{b}_n + (1 + c)(\dot{v}_n - \dot{b}_n).$$

and, hence,

$$\dot{b}' = b + (1 + c)n_n^T(v - b),$$

where $n$ denotes the unit normal vector to the paddle.

The robot kinematics relevant to the task of batting a body relates the machine’s configuration to the normal vector at a point on its paddle. Let $\tilde{p}$ represent (in homogeneous coordinates) a parametrization of the (finite) surface area expressed with respect to the gripper frame, $\mathcal{F}_g$. Associated with each point on the paddle’s surface, $p(o)$, is the unit normal, $\tilde{n}(o)$, again the homogeneous coordinate representation of the vector with respect to $\mathcal{F}_g$. The paddle’s “Gauss map” [52] is now parameterized as

$$N_{s} = [n(o), p(o)]$$

Denote $H(r)$ the robot’s forward kinematic map taking a configuration, $r \in \mathcal{R}$, to the homogeneous matrix representation of the gripper frame with respect to the base. The world frame representation of any paddle normal at a point is thus specified by the extended forward kinematic map,

$$G = [n(r, s), p(r, s)] = H(r)N(s)$$

At the instant of impact with a ball $b \in \mathcal{W}$, the robot’s configuration is of necessity in

$$\mathcal{R}_t \triangleq \{ r \in \mathcal{R} : (b, r) \in \mathcal{C} \},$$
the set of robot configurations capable of reaching a point \( b \). The robot's effect upon the ball is then determined by its velocity at this point,

\[ (r_b, \dot{r}_b) \in \mathcal{V}_b \triangleq \bigcup_{r \in \mathcal{R}_b} T_r \mathcal{R}. \]  

(17)

In particular the linear velocity of the hit point due the robot's motion may now be written explicitly as

\[ v = \sum_{i=1}^{\dim \mathcal{R}} \dot{r}_i D_r, H (r) \dot{p}(s) = D_r p \dot{r} = D_p \Pi_R \dot{r}; \quad \Pi_R \triangleq [D\pi_R]^{T} = \left[ \begin{array}{c|c} I_{\dim \mathcal{R} \times \dim \mathcal{R}} & 0_{\dim \mathcal{R} \times \dim \mathcal{R}} \end{array} \right]. \]  

(18)

Notice that the appearance of \( \Pi_R \) eliminates the velocity contribution due to the ball's motion tangential to the paddle surface. Combining (18), and (??) we may now rewrite the impact event (14) in terms of a "collision map" \( c : T\mathcal{C} \rightarrow T\mathcal{B} \), as

\[ \dot{b}' = \dot{b} + c(b, \dot{b}, r, \dot{r}) \]

\[ c(b, \dot{b}, r, \dot{r}) \triangleq -(1 + \alpha)n(r, s_c(b, r))n^{T}(r, s_c(b, r))(\dot{b} - D_p \Pi_R \dot{r}). \]  

(19)

Note that the vector direction attained by \( c \) is governed entirely by the choice of some configuration \( r_b \in \mathcal{R}_b \). This will serve the role of input in an abstracted model of impact events to be developed below (20).

Denote by

\[ \mathcal{V} \triangleq \bigcup_{b \in \mathcal{B}} T\mathcal{R}_b, \]

the robot's choices of impact normal velocity for each workspace location. Suppose that the robot strikes the ball in state \( w_j = (b_j, \dot{b}_j) \) at time \( s \) with a velocity at normal \( v_j = (r_j, \dot{r}_j) \in \mathcal{V} \) and allows the ball to fly freely until time \( s + t_j \). According to (19), composition with time of flight (11) yields the "environmental control system"

\[ w_{j+1} = f(w_j, v_j, t_j) \triangleq A_{t_j} w_j + A_{t_j} \left[ \begin{array}{c} t_j c(w_j, v_j) \\ c(w_j, v_j) \end{array} \right], \]  

(20)

that we will now be concerned with as a controlled system defined by the dynamics

\[ f : T\mathcal{B} \times \mathcal{V} \times IR \rightarrow T\mathcal{B}, \]

with control inputs \( u \in \mathcal{U} \triangleq \mathcal{V} \times IR \) (\( v_j \) and \( t_j \)).

### 5.2 A New Practicable Stability Mechanism

According to (20), the appropriate planning space for this problem is

\[ \mathcal{P} = \mathcal{W} \triangleq T\mathcal{C}/T\mathcal{R}, \]

the body phases at impact. There is still a dynamical system but it is now discrete.

#### 5.2.1 Encoding a Simple Juggling Task

Probably the simplest systematic behavior of this system imaginable (beyond the ball at rest on the paddle), is a periodic vertical motion of the ball. In particular, we want to be able to specify an arbitrary "apex" point, and from arbitrary initial conditions, force the ball to attain a periodic trajectory which
passes through that apex point. This corresponds exactly to the choice of a fixed point, \( w^* \), in (20), of the form
\[
\begin{bmatrix}
    b^* \\
    \dot{b}^*
\end{bmatrix}, \quad b^* \in \mathbb{R}^3; \quad \dot{b}^* = \begin{bmatrix} 0 \\ \nu \end{bmatrix}; \quad \nu \in \mathbb{R}^3,
\]
(21)
denoting a ball state-at-impact occurring at a specified location, with a velocity which implies purely vertical motion and whose magnitude is sufficient to bring it to a pre-specified height during free flight. Denote this four degree of freedom set of vertical one-juggles by the symbol \( \mathcal{J} \).

The question remains as to which tasks in \( \mathcal{J} \) can be achieved by the robot's actions. In particular we wish to determine which elements of \( \mathcal{J} \) can be made fixed points of (20). Analysis of the fixed point conditions imposes the requirements on that \( w^* \):
\[
\dot{b}^* = \frac{1}{2} \lambda \ddot{a}
\]
and for some \((r, \dot{r}) \in T\mathcal{R}\) and \( \lambda \in \mathbb{R}^+ \),
\[
p(r, s_b(b^*, r)) = b^* \quad \text{and} \quad c(b^*, \dot{b}^*, r, \dot{r}) = -\lambda \ddot{a}.
\]
Every element of \( \mathcal{J} \) satisfies (22), since this simply enforces that the task be a vertical one-juggle. For the Bühler Arm, the requirement (23) necessitates that \( n \) be aligned with \( \ddot{a} \) so as not to impart some horizontal velocity on the ball. From (??) it is clear that this will only be the case when \( r \in \mathcal{R}^* \), where
\[
\mathcal{R}^* \triangleq \{ r \in \mathcal{R} : \cos(r_3) \sin(r_2) = -1 \}.
\]
Thus, we can conclude that only those elements of \( \mathcal{J} \) satisfying the condition \( b^* \in p(\mathcal{R}^*) \) will be fixable. In particular, \( \mathcal{R}^* \) corresponds to the paddle being positioned parallel to the floor, and thus \( p(\mathcal{R}^*) \) is an annulus above the floor, as is intuitively expected.

This simple analysis now furnishes the means of tuning the spatial locus and height of the desired vertical one juggle. The fixed-point input satisfying these conditions, \( u^* \), is given by
\[
u^* = \begin{bmatrix}
    \frac{2}{\gamma} \| b^* \| \\
    \frac{\gamma}{\gamma + 1} D_p \dot{b}^*
\end{bmatrix}.
\]

### 5.2.2 The Stability Properties of Unimodal Maps

Say that an abstract feedback law for (20), \( g : \mathcal{V} \rightarrow \mathcal{V} \times \mathbb{R} \), is a verticle one-juggle strategy if it induces a closed loop system,
\[
f_g(w) = f(w, g(w)),
\]
for which \( w^* \in \mathcal{J} \) is asymptotically stable fixed point. For our original planar machine [9] it was shown that the linearized environmental control system (20) was controllable around every vertical one juggle task. However, experiments system revealed that the linearized perspective was inadequate: the domain of attraction resulting from locally stabilizing linear state feedback was smaller than the resolution of the robot's sensors [9]. Successful juggling was achieved by recourse to the "mirror algorithms" described below. Analytical results obtained to date suggest that these control algorithms owe their success to a new global stability mechanism quite different from the one explored in the previous section except in that it satisfies the critical criterion of "practicability" established in the introduction.

### 5.2.3 A Practicable Global Stability Mechanism

In contrast to the notion of energy dissipation that has been known for more than a century [51], the juggling behavior seems to arise through a stability mechanism that has been only recently recognized.
The principal results required here were stated a little more than a decade ago by Singer [50] and Guckenheimer [19]. They studied the dynamical systems arising from iterations of a special class of maps on the unit interval into itself. These $S$-unimodal maps increase strictly towards a unique maximum and strictly decrease over the remainder of the interval. Moreover, they have a negative Schwarzian Derivative [50].

Singer showed that $S$-unimodal maps can have at most one attracting periodic orbit [50]. Guckenheimer showed that the domain of attraction of such attracting orbits includes the entire unit interval with the possible exception of a zero measure set [19]. Thus, an asymptotically stable orbit of an $S$-unimodal map is essentially globally asymptotically stable. In other words, a local computation at a candidate fixed point suffices to demonstrate its global stability properties.

5.2.4 Invariance

Although the Singer-Guckenheimer theory is stated in terms of the apparently restrictive class of unit interval preserving maps, it extends to (at least) all their differentiable conjugates. Namely, say that $g$ is a smooth $S$-unimodal map if there is an $S$-unimodal map, $f$, to which $g$ is differentiably conjugate — i.e. there exists a smooth and smoothly invertible function, $h$ such that $g = h \circ f \circ h^{-1}$. It is straightforward to show that an attracting orbit of a smooth $S$-unimodal map is essentially globally asymptotically stable [16, 15].

Smooth $S$-unimodal maps form a sufficiently large family that this theory appears to have broad engineering applicability. For example, as described below, the line-juggler map falls within this class. Moreover, we have shown that simplified models of Raibert's hopping robots give rise to smooth $S$-unimodal maps as well [11]. An important caveat is that the Singer-Guckenheimer theory at present has only limited extensions to higher dimensional systems. Thus, in all cases where we would like to invoke these results, we have had to restrict attention to simplified one degree of freedom models of the systems in question. We presume that these form attracting invariant submanifolds in the more general case.

6 A Continuous Controller for Robot Juggling

This section introduces the mirror laws, sketches the relation to the Singer-Guckenheimer theory outlined above, and discusses certain generalizations.

6.1 The One Degree of Freedom Case

For ease of exposition it seems most convenient to introduce the discussion of the mirror algorithm to the simplified one-degree-of-freedom environment depicted in Figure 7.

![Figure 7: The Line-Juggler Model](image)

In any event, this is the model to which the Singer-Guckenheimer results are most directly applicable.
6.1.1 Construction

Suppose the robot tracks exactly the continuous "distorted mirror" trajectory of the puck,

\[ r = -\kappa_{10} \dot{b}, \]

where \( \kappa_{10} \) is a constant. In this case, impacts between the two do occur only when \( (r, b) = (0, 0) \) with robot velocity

\[ \dot{r} = -\kappa_{10} \dot{b}. \]  \hspace{1cm} (25)

For simplicity, assume that the desired impact position is always selected to be \( b^* = 0 \). Any other impact position can be achieved by shifting the coordinate frame for robot and puck to that position. Now solving the fixed point condition \( \dot{b}' = c(b^*, r(b^*)) = -\dot{b}^* \) for \( \kappa_{10} \) using (33) and (25), yields a choice of that constant, \( \kappa_{10} = (1 - \alpha)/(1 + \alpha) \) which ensures a return of the puck to the original height. Thus a properly tuned "distortion constant," \( \kappa_{10} \) will maintain a correct puck trajectory in its proper periodic course.

The ability to maintain the vertical one-juggle — fixed point condition — with such a simple mirror control law is an encouraging first step, but still impractical, as it is not stable. The second idea at work which will assure stability is borrowed from Marc Raibert [41], who also uses the total energy for controlling hopping robots. In the absence of friction, the desired steady state periodic puck trajectory is completely determined by its total vertical energy,

\[ \eta(w) = \frac{1}{2} \dot{b}^2 + \gamma b. \]

This suggests the addition to the the original mirror trajectory,

\[ r = -\kappa_1 \omega \dot{b}; \quad \kappa_1(\omega) \triangleq \kappa_{10} + \kappa_{11}[\eta(w^*) - \eta(w)], \] \hspace{1cm} (26)

of a term which "servos" around the desired steady state energy level. Thus, implementing a mirror algorithm is an exercise in robot trajectory tracking wherein the reference trajectory is a function of the puck's state.

6.1.2 Analysis

It is shown in [11] that the feedback law resulting from the strategy described above is The time of flight and the robot impact velocity is

\[ u_1 = \frac{2}{\gamma} \dot{b}^* \quad \text{and} \quad u_2 = \dot{r} = -\kappa_1 \dot{b}. \]

Substituting these robot control inputs in (23) yields the scalar map of puck impact velocities just before impact at the invariant impact position \( b^* = 0 \),

\[ f(b) = \dot{b} \left( 1 - \beta(\dot{b}^2 - b^*) \right), \] \hspace{1cm} (27)

where \( \beta = \kappa_{11} \cdot (1 + \alpha)/2 \).

It is not hard to show [11] that (27) satisfies the conditions of "S-unimodality" described above. A check of (27) reveals that the fixed point is locally stable when

\[ 0 < \beta < \frac{2/\zeta - 1}{\dot{b}^2}. \] \hspace{1cm} (28)

There immediately follows,

**Theorem 3** ([15]) *The mirror algorithm for the line-juggler results in a successful vertical one juggle which is essentially globally asymptotically stable as long as \( \beta \) satisfies the inequality (28).*
6.1.3 Experiments

We have shown a gratifying correspondence between theoretical predictions based upon the Singer-Guckenheimer results, simulation studies, and physical data [11]. Perhaps the most dramatic depiction of this correspondence is suggested by our bifurcation studies, for which there is no space in the present paper. This section nevertheless provides some feeling for the predictive power of the theory described above.

We have shown in the previous section that the local dynamical behavior is essentially global. The data in Figure 8 confirm that the transients can be predicted by recourse to local linear analysis of the scalar impact map. Evaluating the derivative of (??) at the fixed point (??) for the four gain settings $\kappa_{11} = 3/5/7/9 \cdot 10^{-5}$ shown in the figure, we predict locally an overdamped, critically damped, underdamped and an unstable response, respectively. This behavior is confirmed even from large initial conditions ("globally") on the juggling apparatus. When inspecting the transient for the last gain setting $\kappa_{11} = 9 \cdot 10^{-5}$ closely we see that it maintains a small oscillation, the predicted onset of instability. The fixed point in the presence of friction depends on the gain setting $\kappa_{11}$. Figure 8 confirms as well our ability to predict the steady state values for the vertical impact velocities with less than 3% error.

6.2 Generalization to More Degrees of Freedom

The mirror algorithm is readily generalized to situations involving higher degrees of freedom. For example, consider the system, pictured in Figure 6, consisting of a three degree of freedom ball and robot along with a real-time vision system.

6.2.1 Construction

The "mirror law," may again be expressed as a map $m : \mathcal{TB} \to \mathcal{R}$, so that the robot's reference trajectory is determined by

$$r(t) = m(u(t)).$$

Begin by using (??) to define the the "joint space position" of the ball

$$\begin{bmatrix} \theta_b \\ \phi_b \\ \psi_b \\ \epsilon_b \end{bmatrix} \triangleq p^{-1}(b). \quad (29)$$

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) $r_{d1} = \phi_b$ causes the paddle tracks under the ball at all times.

(ii) The paddle "mirrors" the vertical motion of the ball through the action of $\theta_b$ on $r_{d2}$ as expressed by the original one degree of freedom mirror law [14].

(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

$$\eta \triangleq \gamma \dot{\theta}_b + \frac{1}{2} \dot{\phi}_b^2$$

and,

$$\rho_b \triangleq \sin(\theta_b) s_b \quad (30)$$

24
respectively. The complete mirror law combines these two measures with a set point description ($\eta$, $\rho$, and $\phi$) to form the function

\[
r_d = m(w) \triangleq \begin{bmatrix}
\frac{\eta}{\phi_b} \\
-\frac{\pi}{2} - (\kappa_0 + \kappa_1 (\eta - \eta_0)) \left( \theta_b + \frac{\pi}{2} \right) + \kappa_{00} (\rho_b - \rho) + \kappa_{01} \dot{\rho} \\
\kappa_{10} (\dot{\phi}_b - \dot{\phi}) + \kappa_{11} \dot{\phi}_b
\end{bmatrix}
\]  

(21)

6.2.2 Implementation

For implementation, the on-line reference trajectory formed by passing the ball's state trajectory, $w(t')$, through this transformation must be passed to the robot tracking controller. The tracking controller resembles in practice an inverse dynamics type scheme that we have termed "ID" [53]. In geometric terms, it represents a means of making the surface

\[
\mathcal{M} \triangleq \{(v, w) \in T\mathcal{R} \times TB : \|v - T_w m F (w)\| = 0\}
\]

an attracting invariant submanifold by proper choice of feedback law [9]. In practice, these terms are computed symbolically from $m$ (31) and $F$ (??).

6.2.3 Experiments

We have succeeded in implementing the one-juggle task as defined in Section ?? on the Bühler arm. The overall performance of the constituent pieces of the system — vision module, juggling algorithm, and robot controller — have each been outstanding, allowing for performance that is gratifyingly similar to the impressive robustness and reliability of the planar juggling system. We typically record thousands of impacts (hours of juggling) before random system imperfections (electrical noise, paddle inconsistencies) result in failure. Figure 9 shows the three projections of the ball's trajectory for a typical run. As can be seen the system is capable of containing the ball within roughly 15cm of the target position above the floor. Figure 10 depicts the ball's height over time over a short segment of this same run, and demonstrates that the machine has also been capable of controlling the height of the flights to within 5cm of the desired height of 55cm. Interesting features shown in Figure 9 include:

(a) The introduction of the ball.

(b) A loss of visual data during the experiment. In this example the ball traveled outside the camera's field of view, but returned in roughly the same location allowing the vision system to recover.

It is worth noting that in the x-z and y-z projections there is evidently spurious acceleration of the ball in both the x and y directions. Examination of Figure 11 shows that the ball indeed seems to "curve" while in flight. Tracing this phenomenon through the system confirmed an earlier suspicion; our assumption that gravity is aligned with the axis of rotation of the base motor is indeed erroneous. Correction of this calibration error requires the addition of trivial workspace cues (a plum bob) to allow the direction of the gravitational force to be calibrated along with the remainder of the system. This correction is now in progress.

Analysis of our planar juggler revealed [11] that the strong stability and robustness properties of the machine could be attributed to the Singer-Guckenheimer unimodal return map [18, 25]. We have not yet undertaken an analogous formal study of the closed loop return map (24) induced by the spatial mirror.
law (31) but conjecture that the same stability mechanism will be found here too. In advance of this analysis, we note in passing our ability to generate the bifurcation to a period two orbit in (24) predicted by the Singer-Guckenheimer theory as depicted in Figure 12.

6.3 Generalization to Other Tasks

A two-juggle task is the requirement that the robot perform two simultaneous one-juggle tasks with two independent balls separated in both space and time. Separation in space avoids ball-ball collisions, which are not currently part of the environmental model, while temporal separation (meaning that 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, all before the first falls to the floor. Apparently there is an obstacle present in the phase space of the system which we are attempting to “avoid”. Thus the juggling algorithm must be able to control the phase relationship between the two balls in addition to the three new variables associated with the position and energy of the additional ball. Initially we have attempted to directly apply the algorithms used on the planar juggler [25] to the spatial system, with the hope that they would prove adequate.

In order to be capable of controlling the temporal separation of the balls we introduce a new variable, ball phase, which we will attempt to control. The phase of a ball is measured by

$$
\phi(w) \triangleq -\frac{b_0}{\sqrt{2\eta}},
$$

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,

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

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

$$
m_{II}(w_0, w_1) \triangleq \begin{bmatrix}
\frac{\phi_0}{\kappa_0(\phi_0 - \phi_1)} + \kappa_1\phi_0 & \kappa_2\phi_0
\end{bmatrix} + \kappa_0(\rho_0 - \rho_1) + \kappa_1\rho_0.
$$

This new relationship between the ball’s state and the robot configuration is essentially equivalent to (31), except that the expression for $a_{\phi}$ 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 ball be to 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 $\kappa_2$ is crucial to overall system stability.

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

$$
m_{II}(w_0, w_1) = s(w_0, w_1)m_0(w_0, w_1) + (1 - s(w_0, w_1)) m_1(w_1, w_0).
$$

The function $s$ encodes the mixture between the need to juggle ball 0 (follow $m_0$) or ball 1 (follow $m_1$). Two “urgency” functions are defined by

$$
s(w) = s_p(w)s_d(w),
$$
where

\[ \sigma_p(w) = \frac{1}{2} - \frac{1}{\pi} \text{ArcTan}[k_p(b_a - \zeta)] \]  
\[ \sigma_v(w) = \frac{1}{2} + \frac{1}{\pi} \text{ArcTan}[-k_v \dot{b}_3]. \]  

(37) \hfill (38)

The construction of \( s \) is then given by

\[ s = \frac{\sigma(w_0)}{\sigma(w_0) + \sigma(w_1)}. \]  

(39)

The motivation for this implementation follows directly from the previous work on the planar juggling system [25], and is based on the belief that \( \sigma(w_i) \) encodes a measure the amount of attention required by ball \( i \). By proper choice of the constants \( k_p, k_v, \) and \( \zeta, \sigma(w_i) \) can be designed to vary smoothly between 1 – when ball \( i \) needs to be hit – and 0 – immediately after it has been struck. The definition of \( s \) then smoothly combines these two “urgencies” causing the robot to track \( m_i \) when \( \sigma(w_i) \) approaches 1.

As in the one-juggle the \( q_d \) and its temporal derivatives are directly computed from \( F(w_0), F(w_1), \) and \( m_H \) and its jacobians. Experiments with this scheme are now in progress.
Figure 8: Line-Juggler Transients: Experimental Data
Figure 9: One-Juggle ball trajectory: (i) X-Y projection (ii) X-Z projection and (iii) Y-Z projection.

Figure 10: One-Juggle ball trajectory: Height vs. time.
Figure 11: One-Juggle ball trajectory: X and Y position vs. time.
Figure 12: An attracting period two orbit
7 Conclusion

This talk has presented examples of the task encoding program that underlies all of our intelligent controls work within the Yale Robotics Laboratory. This approach employs feedback controllers to shape the dynamics of the resulting closed loop robot-environment physical system. When the geometry of the feedback law has been properly adjusted, it is possible to realize autonomous behavior by virtue of certain practicable stability mechanisms. The strength of this approach has been verified by numerous simulation studies and physical experiments. The primary goal of the talk has been to suggest that programming via vector fields may not be so counter-intuitive as it first sounds.

One constructive technique, the navigation function represents the broadest scope of application we have yet found for any particular encoding. From the point of view of traditional control theory this merely represents an extension of proportional-derivative feedback techniques to settings wherein global convergence cannot obtain from linear methods because the state space is topologically distinct from a Euclidean vector space. Nevertheless, this simple notion seems to have considerable power in applications [29] beyond the control-theoretic reformulation of the famous "piano movers' problem," [49] treated here.

A second constructive technique, the mirror law, arises within the harder but somewhat narrower context of batting unactuated bodies into a periodic orbit. We have succeeded in encoding certain primitive dynamically dexterous tasks as throwing, catching, and juggling using a geometric formalism [14, 19], that seems to admit correctness proofs in simple cases [11, 12]. The stability mechanism underlying the success of these computational structures has been understood only recently within the "chaos" literature, and seems to generalize to successful hopping algorithms as well [25].

We have hinted how a combination of juggling and navigation ideas may extend the purview of this methodology to planning and control of automated assembly, now cast as the problem of controlling a nonholonomically constrained mechanical system [28, 30]. Recent thinking suggests that some combination of navigation functions [56] and juggling theory [31] may offer some help in solving such problems. This work is still very tentative.

In the next few years it will be essential to develop an adaptive version of these techniques in order to tolerate uncertainty in a priori knowledge of the environment. Moreover, some means of classifying the topological equivalence class of reasonably common free configuration spaces will be essential if the constructive nature of the approach is to be generalized further. Most broadly, the chief need is to remove enough a priori assumptions in the theory that reasonable physical experiments may begin.
Acknowledgement

It is a pleasure to acknowledge my students, Martin Bühler, Elon Rimon, Louis Whitcomb, and Al Rizzi who contributed many of the results reported here and most of the accompanying figures as well.

References


