Cambridge Scientific Abstracts
Computer Abstracts
EIC Intelligence
Engineering Index/Compendex
INSPEC Information Services

Printed in The Netherlands

0921-8890 / 92 / $05.00

Published eight times per year

Robotics and Autonomous Systems 9 (1992) 5–39
Elsevier

Task encoding: Toward a scientific paradigm for robot planning and control *

Daniel E. Koditschek

Dept. of Electrical Engineering, Center for Systems Science, Yale University, New Haven, CT 06520, USA

Abstract

Koditschek, D.E., Task encoding: Toward a scientific paradigm for robot planning and control, Robotics and Autonomous Systems, 9 (1992) 5–39.

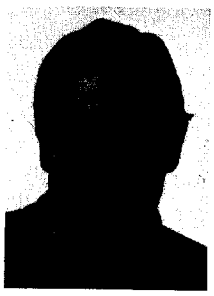
An autonomous machine requires a description of its designated task in a language that it 'understands'. The machine language of robots – physical mechanisms endowed with actuators and sensory devices for the purpose of performing work – is the language of dynamical systems. Since the diversity of robotic tasks is immense and our practical experience with computationally well endowed sensed and actuated mechanisms is still very slight, there seems little possibility of imposing at present a meaningful uniformity of description and command methodology on the practice of building these machines. However, a number of thematic desiderata can be articulated whose pursuit will arguably increase the ease of designing robotic algorithms that are predictably successful. This paper presents a summary of our approach to a number of robotic tasks that brings out by example the nature and utility of these desiderata.

Keywords: Robot planning; Robot control; Dynamical systems theory applied to robotics.

1. Introduction

This paper addresses the problem of how to say what we mean and how to mean what we say to a robot. Such matters lie at the intersection of AI and Control: Robotics might be viewed as a discipline concerned with intermediating between the traditional concerns of each. AI concerns itself with the representation of abstract goals and the nature of reasoning required to bring about their realization. Control engineers study classes of dynamical systems and develop strategies for producing input signals as a function of sensory readings that achieve specified output signals. Let a robot be conceived rather broadly as a mechanical system whose input signals take the form of commands to actuators – motors that deliver torque or force – and for which all goals of interest involve *work* in the physical sense – the application of specified forces over specified motions. This paper advances the view that roboticists ought to worry about how to take an abstract user goal – 'make scrambled eggs' or 'fold the laundry' or 'serve tea in the study' – and produce a control strategy resulting in the mechanical system achieving that goal.

More concretely, this paper 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 dy-



Daniel E. Koditschek received his BS in 1977 in Engineering and Applied Science and Ph.D. in 1983 in Electrical Engineering from Yale University. He was appointed Assistant Professor of Electrical Engineering at Yale in 1984 and is presently an Associate Professor in that Department. His interests include the application of dynamical systems theory to autonomous machine design, nonlinear control theory, and the application of computational theory and hardware to feedback control of physical processes.

* This work was supported in part by the National Science Foundation under grant DMC-8505160, and, in part by a Presidential Young Investigator Award.

namical systems as well as a proof that the resulting closed loop behavior will correctly achieve the desired goals. Such a method ought to include notions of:

Geometric language: Automated reasoning about the static properties of the physical environment has been a central concern of computational geometry for at least a decade. That the dynamics of physical settings admits an essentially geometric representation has been understood for several decades [1], but has not generally been exploited in engineering practice. One presumes that geometry should offer a natural language for expressing abstract goals relating to the physical world.

Synthesis: The step of generating a control strategy to achieve a specified abstract goal should be automatic. Since control strategies for mechanical systems can be succinctly rendered using mathematical relations between the continuous input and output variables, there is intuitive reason to hope that geometric task specification might be particularly well suited to automatic generation of control strategies.

Correctness: As the consequences of unanticipated failure in engineered systems become more serious, the importance of verifiably correct implementations increases. If they are not to pose a significant hazard to their human users, robots must be in large measure predictable. Given a model of the physical environment, a representation of the task and a control strategy, it seems essential to be able to furnish a formal proof that the resulting mechanical system will exhibit correct behavior. Although the validity of such proofs can be no greater than that of the models with respect to which they are argued, formal reasoning must reveal the underlying assumptions necessary to the desired conclusion. This should promote, at the very least, a more effective anticipation of failure modes.

Generalization: When are two apparently different tasks actually the same? How can a solution that works in simple settings with low degrees of freedom generate a paradigmatic approach to analogous problems involving greater complexity? The tradition of parsimony in natural science should extend to synthetic systems. In order to avoid repeating effort, one seeks a means of generalizing the passage from task specification to control algorithm along with the accompanying correctness proof.

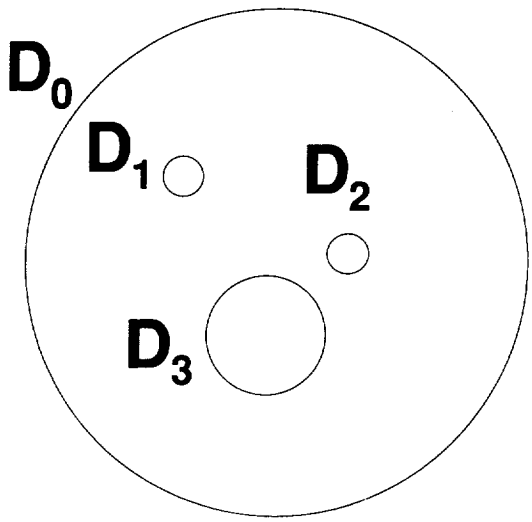


Fig. 1. A Euclidean sphere world [33].

Since there is as yet little formal insight into a general framework for defining and achieving these desiderata, it seems most profitable to proceed by example. This paper describes work concerning various robotic behaviors that might plausibly connote 'intelligence'. The focus of these examples is taken from the author's past work on navigation, juggling, and assembly. They can all be 'generated' by imposing slight variations on the simple geometric theme depicted in *Fig. 1* – a planar disk punctured by other disks. To say that the disparate task specifications and control solutions considered below may be generated from more or less the same setting belies the very different nature of each problem we will consider. Even a casual glance at the diversity of physical models, task specifications and controller constructions offered below will quickly convince the reader that the author can lay no claim to a unified paradigm of robotics. Indeed, one of the central points of the paper is to urge the importance of studying carefully the very diverse nature of specific robot tasks. At the same time, certain unifying themes that point back toward the desiderata set out above may be discerned.

Most importantly, the emphasis upon the dynamical nature of robot tasks and the utility of adopting a dynamical means of specification

seems critical. At a time when machines can play chess better than almost every human being on earth, there are still no machines capable of walking up the stairs as well as the average toddler. Although intelligence must surely involve more than simple dexterity, it does seem increasingly plausible that physical machine intelligence might be more likely to emerge from the exercise of mechanical competence than the reverse.

Given this emphasis upon translating goals into dynamical relations, a second theme concerns the utility of feedback controllers in achieving reliable and sophisticated behaviors. As is well known, feedback controllers, unlike open loop plans, are designed to work over large classes of initial configurations (tolerance to state uncertainty) and often succeed even when the underlying dynamics are imperfectly modeled (tolerance to parametric uncertainty). Further, this approach to planning encourages the design of 'canonical' procedures for 'model' problems which may then be instantiated in particular settings by a change of coordinates [46,47]. Moreover, the resulting compression of the more usual planning–execution hierarchy tends to force the designer's parametrization of the modeled environment into the 'low level' controller, thus suggesting a means of turning fuzzy questions of learning into clearly posed (albeit, difficult) problems in parameter adaptive control. Finally, such specifications make explicit the resulting (closed loop) dynamical system, and afford the application of well developed mathematical analysis when attempting proofs of correctness.

The reliance upon feedback mechanisms for control brings out the third central theme of the paper: the virtues of global stability mechanisms. They lend autonomy – that is, freedom from dependence upon some 'higher level' of intelligence – at the planning level. They enhance robustness – that is, experimentally observable resistance to changes in the robot or its environment – at the execution level. They encourage the design of 'canonical' procedures for 'model' problems which can then be instantiated in particular settings by a change of coordinates. In linear systems, local stability immediately implies global stability, but this is not at all true in general. As any student of nonlinear dynamical systems knows, global stability is a relatively rare phenomenon and, worse, there are no generally valid computable necessary conditions by which to detect its presence. Thus, to be useful in engineering applications – to be *practicable* – a global stability mechanism must admit some effectively computable test.

This latter idea connects, in turn, with an approach to the problem of generalization in task robot encoding. Since practicable global stability mechanisms are hard to achieve, it makes sense to use and re-use those that we have already found as much as possible. Generalization techniques are concerned with discovering the invariants of a task domain so that apparently different problems that share the same underlying features can be identified as such. Typically, the identification extends to the stability properties of the systems so that a solution for one problem is automatically applicable across its generalizations. Thus a lot of the sifting through examples in this paper is motivated by a desire to understand what transformations of what aspects of the robot and its environment reduce to the same class of problem. It must be stated at once that this understanding is presently rather rudimentary. For some time to come, the process of task encoding will likely remain rooted in the particulars of each task.

The paper proceeds as follows. An informal sketch of the similarities and differences in various robotic tasks arising from *Fig. 1* is presented in Section 2. Section 3 offers a solution methodology to range of problems involving navigation with no contact. Section 4 considers a few problems wherein the making and breaking of contact is an essential aspect of the task.

Evidently, it is presently not at all clear how to tell a robot to 'fold the laundry' or 'scramble the eggs' or 'make the bed'. 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. One hopes that carefully chosen miniature settings will hold the key to much more difficult situations.

2. Representation: The geometry of robotic tasks

This section is concerned with exploring the radical diversity of robotic tasks. We shall start with the very simple idealized setting depicted in

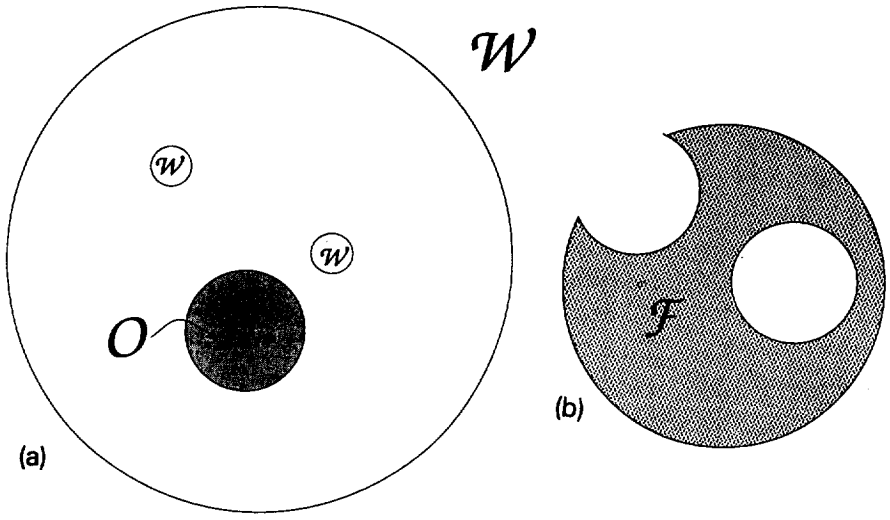


Fig. 2. (a) A disk object moving in a planar sphere world. (b) The associated configuration space.

conveniently represented by an $(n+1)\times(n+1)$ homogeneous matrix,

$$\boldsymbol{p}_i = \begin{bmatrix} P_i & p_i \\ 0^{\mathrm{T}} & 1 \end{bmatrix}. \tag{1}$$

Thus, the translation component of $\boldsymbol{p}_i$ is directly represented by the n-vector p_i. The special rotation group, $SO(n)$ admits a variety of representations in addition to the orthogonal matrix, R, in (1), the most useful for our purposes being its 'logarithm', r [17],

$$R = \exp\{Jr\}; \quad J = -J^{\mathrm{T}}; \quad r \in \mathbb{R}^n.$$

The logarithm is a multi-valued function – there are infinite choices of r for each R – hence this representation is not yet formally well defined. However, since all examples in the present paper take place on the plane, the formal discrepancy

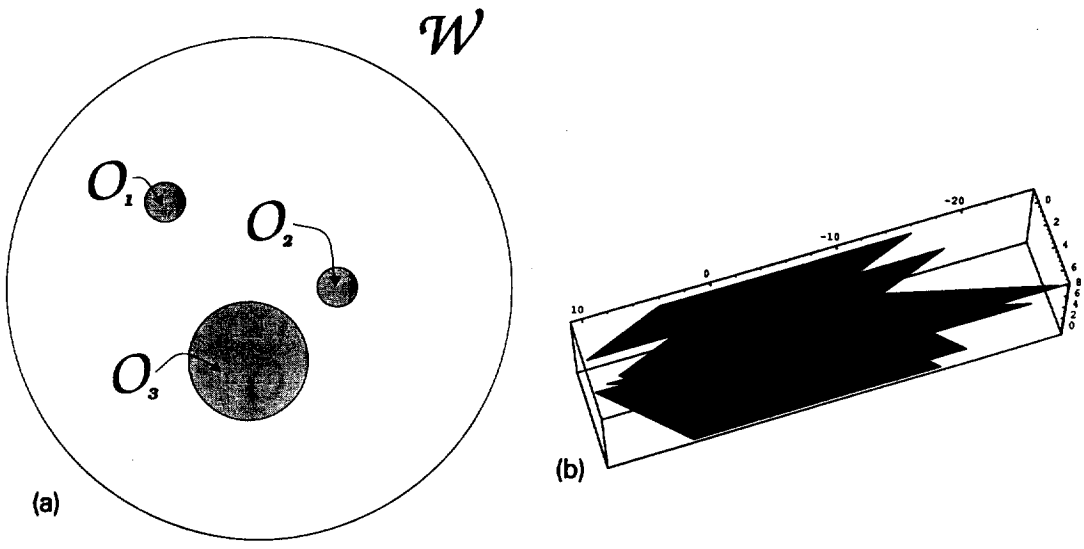


Fig. 3. (a) Three disk objects moving in a planar sphere world. (b) Schematic of the associated six dimensional freespace.

may be resolved by treating r as a real number modulo 2π – the angle of the associated planar rotation. This representation scheme forms the basis for our coordinatization of $\mathcal{C}$.

2.1.1. Freespace

The *free space*, $\mathcal{F}$ is the set of placements resulting in each object inside the workplace, $p_i\mathcal{O}_i \subset \mathcal{W}$, allowing intersections between objects only on their boundaries,

$$\mathcal{F} = \{(p_1, \ldots, p_M) \in \mathcal{P}: \; p_i\mathcal{O}_i \cap p_j\mathcal{O}_j \subset p_i\partial\mathcal{O}_i \cap p_j\partial\mathcal{O}_j \text{ and } p_i\mathcal{O}_i \cap \mathcal{W} \subset \partial\mathcal{W}\}.$$

Example 2.1.1.a. *A single object amidst simple obstacles*

In *Fig. 1*, take as object the third disk, $\mathcal{O} = D_3$ and locate a frame of reference at its center whose axes are aligned with the edges of the page. Let the other disks denote the workplace, as depicted in *Fig. 2a*. Since $\mathcal{O}$ is invariant under the planar group of rotations, $SO(2)$, it follows that $\mathcal{P} = SE(2)/SO(2) \approx \mathbb{R}^2$. In other words, a coordinate representation is simply given by the ordered pair $q = [p_1, p_2]^T$. A representation of the free space, $\mathcal{F}$, is obtained by thickening each component of $\mathcal{W}$ by the radius of $\mathcal{O}$ [36]. For example, if the radius of $\mathcal{O}$ is zero, that is, if the object is a single point, then clearly the free space coincides with the workplace, $\mathcal{F} \approx \mathcal{W}$. The free space corresponding to the present situation is depicted in *Fig. 2b*.

Example 2.1.1.b. *Multiple moving objects*

Reconsider the situation depicted in *Fig. 1*. Suppose that the workplace is formed simply by the outer disk, $\mathcal{W} = D_0$, and the rest of the three disks are taken to be objects, $\mathcal{O} \triangleq D_i$, as depicted in *Fig. 3a*. Each object being radially symmetric, the copy of $SE(2)$ that acts on it may be identified again with $\mathbb{R}^2$ and we have $\mathcal{P} \approx \mathbb{R}^6$.

Adopting the coordinate representation described above, let $p_i = [p_{i1}, p_{i2}]^T$ denote the coordinates of the center of $\mathcal{O}_i$ so that $q = [p_1^T, p_2^T, p_2^T]^T$. In order to complete the specification of $\mathcal{C}$ we must place functional constraints on the allowable configurations, q.

Further supposing that $\mathcal{O}_i$ has radius ρ_i, from the 'boundary component functions',

$$\beta_{ij}(p) \triangleq \begin{cases} \| p_i - p_i \|^2 - (\rho_i + \rho_i)^2 \\ \quad 0 < i, j, \leq M - 1, \\ -\| p_j \|^2 + \rho_0^2 \\ \quad i = 0; \; 0 < j \leq M - 1, \end{cases}$$

and, defining their product,

$$\beta(p) \triangleq \prod_{0 \leq i,j \leq 3} \beta_{ij}(p), \tag{2}$$

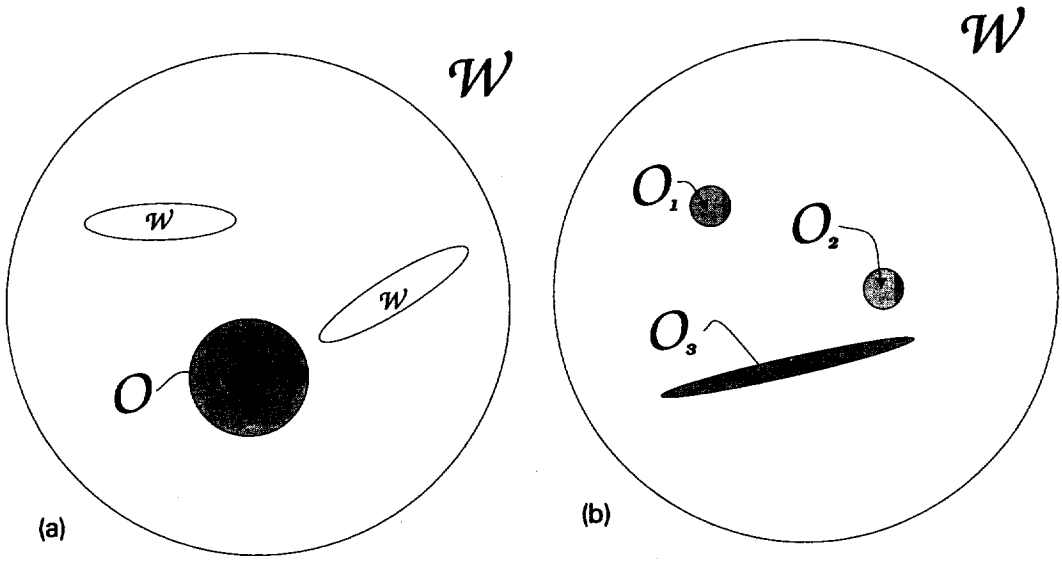


Fig. 4. (a) Changing the workplace geometry. (b) Changing the object geometry.

note that the freespace is given by

$$\mathscr{C} \triangleq \{p \in \mathbb{R}^6 : \beta(p) \geq 0\},$$

schematized in *Fig. 3b*.

Evidently, the expression (2) serves equally well to characterize the freespace of Example 2.1.1.a depicted in *Fig. 2* as it does that of Example 2.1.1.b in *Fig. 3* if one restricts the domain of β appropriately. Thus, the algebraic representation of the scene depicted in *Fig. 1* is incomplete without a further distinction between fixed workplace and potentially mobile objects. Not surprisingly, the possible variations in freespace become even greater when topologically equivalent variants of *Fig. 1* are considered. The workplace of *Fig. 4a* is a mere deformation of that of *Fig. 2a* and the object is identical in both scenes. However their respective freespaces differ not only in metric properties but in topological properties since a sketch of the freespace corresponding to *Fig. 4a* would reveal a planar region deformable to a simple disk which the region depicted in *Fig. 2b* clearly is not. The constrast between the scenes of *Fig. 3a* and *Fig. 4b* is even more extreme since the freespace corresponding to the latter can only be embedded in a space of dimension seven.

2.1.2. Kinematics

So far, all the constraints on the placements take the form of inequalities involving $\mathscr{P}$. But many robotic problems of interest involve *kinematic* constraints that may take the form of equalities and may involve the infinitesimal placements, $T\mathscr{P}$, as well. Since $\mathscr{P}$ is parallelizable, there is never a loss of generality in treating $T\mathscr{P} = \mathscr{P} \times \mathbb{R}^n$ as a cross product. In all of the examples of the present paper, the manifolds resulting from the imposition of constraints on $\mathscr{P}$ will also be parallelizable, and we will denote a tangent vector in the usual local coordinates as $(q, \dot{q}) \in T\mathscr{C}$.

When a constraint function is constant on each fiber, $T_p\mathscr{P}$, and it is used to specify a single level set, then it is called *holonomic*. Holonomic constraints, characterized by equalities always decrease the dimension of the configuration space relative to the freespace.

Example 2.1.2.a. *A simple planar kinematic chain*

Suppose, $\mathscr{O}_3$, the elongated ellipsoidal 'bar' of *Fig. 4b* is pinned in place at its center as depicted in *Fig. 5a*. Then this object is subject to a holonomic constraint whose functional form is given as $\gamma = 0$, where

$$\gamma(p_1, p_2, r_3, p_3) = p_3. \tag{3}$$

It follows that the configuration space has the coordinates $q = [p_1, p_2, r]^T \in \mathbb{R}^5$.

In robotics it has become fashionable to speak of all kinematics whose constraint functions are 'integrable' – that is, constant on each fiber, $T_p\mathscr{P}$, – as holonomic. But this flies in the face of traditional usage. Whenever the configuration space has a boundary, there are nonholonomic

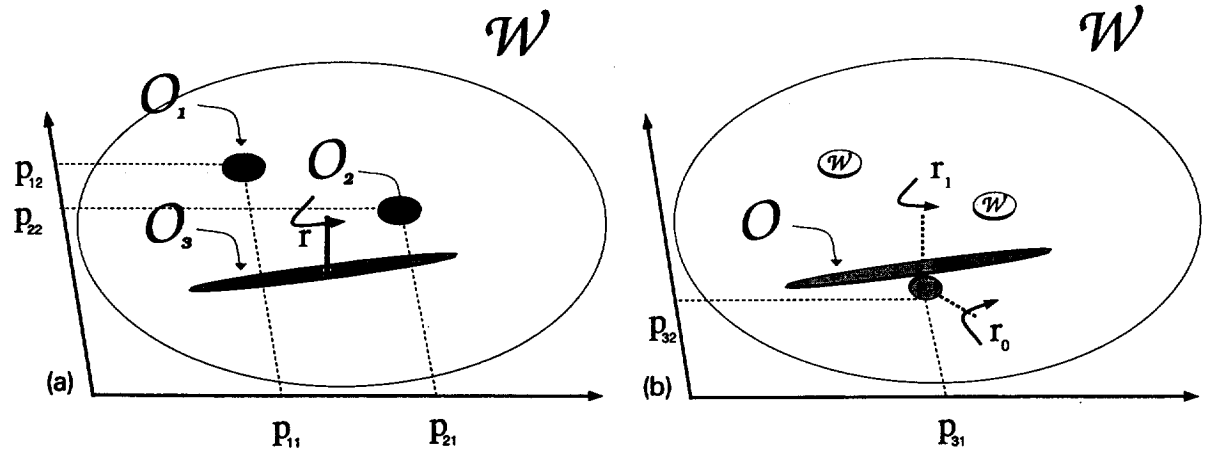


Fig. 5. (a) A one degree of freedom kinematic chain with two free disks. (b) A unicycle navigating amongst disk obstacles.

constraints in the classical sense since the constraint functions will take the form of inequalities [26]. For example, the situations depicted in *Fig. 2* and *Fig. 3* would be classically considered as involving nonholonomic constraints: they are defined in terms of an inequality involving the boundary function (2). And, strictly speaking, they imply tangent bundle constraints that are not constant over every fiber. For letting $\alpha(q, \dot{q}) \triangleq D\beta \cdot \dot{q}$, we require

$$\alpha \triangleq \begin{cases} \in \mathbb{R} & \beta(q) > 0, \\ \leq 0 & \beta(q) = 0. \end{cases} \tag{4}$$

When the task at hand involves simple obstacle avoidance then there is little lost or gained by adopting either usage (although it does seem inefficient for roboticists to depart from tradition unnecessarily). However, as will be seen below in Example 2.3.2.a or Example 2.3.1.c, once manipulation enters the picture, the nonholonomic nature of these situations plays a critical new role in the subsequent planning and control problems.

Both the traditional and robotics literatures seem to agree that when the constraint functions cannot be expressed with a function that is constant over a fiber – when the constraint is not integrable – then the constraint is nonholonomic.

Example 2.1.2.b. *The unicycle*

Imagine in *Fig. 4b* that the bar, $\mathscr{O}_3$, rides on a single wheel as depicted in *Fig. 5b*. In addition to rotating, the object is now free to move forward or backward, but only in the direction that it is pointing. Because the realizable translational velocities depend upon the rotational position, this machine is subject to nonholonomic constraints.

Denote the wheel angle orthogonal to the plane depicted in the figure by r_0, and the angle of the bar in the plane by r_1 so that $q = (p_{31}, p_{32}, r_0, r_1)$. The velocity constraints may be expressed by the function

$$\alpha(q) = \begin{bmatrix} \dot{p}_{31} \\ \dot{p}_{32} \end{bmatrix} - \begin{bmatrix} \dot{r}_0 \cos r_1 \\ \dot{r}_0 \sin r_1 \end{bmatrix} \tag{5}$$

and take the form $\alpha = 0$.

2.1.3. The contact set

The *contact set*, $\mathscr{C}$ contains all those placements at which the intersection over some of the placed objects (and the workplace boundary) is non-empty

$$\mathscr{C} \triangleq \{(p_1, \ldots, p_M) \in \mathscr{F} : p_i\mathscr{O}_i \cap p_j\mathscr{O}_j \neq \emptyset \text{ for some } i, j < M \text{ or } p_k\mathscr{O}_k \cap \delta\mathscr{W} \neq \emptyset \text{ for some } k < M\}.$$

In the simple settings considered in this paper the contact set is contained in the boundary – the closure minus the interior – of the freespace,

$$\mathscr{C} = \partial\mathscr{F} = \overline{\mathscr{F}} - \mathring{\mathscr{F}},$$

although the situation may be much more complicated in general [34].

Unlike the simple situation introduced in Example 2.1.1.a, where the contact set, is exactly depicted by the inner and outer curves of *Fig. 2b*, this boundary set has important additional structure. For $\mathscr{F}$ is a cell complex formed from the union of the free space interior $\mathscr{F} - \mathscr{C}$, an open pre-compact manifold comprising the cells of dimension $6M$, with various lower dimensional cells in $\mathscr{C}$ whose codimension with respect to $\mathscr{P}$ represents the physical notion of contact type – the number of degrees of freedom restricted by touching. These various cells will all be associated with a scalar valued function.

Example 2.1.1.b (continued).

Any placement wherein an object touches the boundary of $\mathscr{W}$ or another object is in the contact set. It will prove useful to further distinguish contacts involving simply one object or more than one in contact simultaneously: the number of objects touching each other or the boundary determines the relative loss of degrees of freedom of the placement – the dimension of the cell.

Algebraically, the contact set is comprised of those placements where β vanishes, $\mathscr{C} \triangleq \beta^{-1}[0]$. A '2-contact' set, $\mathscr{C}'$, may be similarly defined by the vanishing of

$$\bar{\beta} \triangleq \prod_{ijk} \beta_{kj|i}; \qquad \beta_{kj|i} \triangleq \left(\beta_{ij}^2 + \beta_{ik}^2\right).$$

Higher order contacts described by the zeroes of analogously defined functions, $\bar{\bar{\beta}}$, $\bar{\bar{\bar{\beta}}}, \ldots$, lead to situations where increasing numbers of bodies are simultaneously touching.

Example 2.1.2.a (continued).

To simplify the completion of β, the boundary function, and the higher order contact functions,

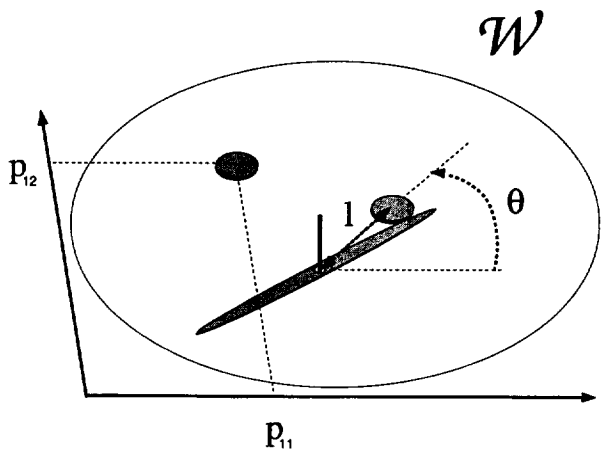


Fig. 6. The kinematics of rigid contact.

$\bar{\beta}$ in the more complicated geometric setting of *Fig. 5*, now assume that object, $\mathcal{O}_3$, the bar, has length λ_3 and no width. Assume, as well, that its pivot point has been placed at the center of the disk, D_0, that defines the workplace, and that this disk also has radius λ_3. Finally, assume that the other two disks, $\mathcal{O}_1$, $\mathcal{O}_2$ have zero radius, $\rho_1 = \rho_2 = 0$.

It will prove helpful in computing these functions to parametrize the holonomically constrained motion of the surface of $\mathcal{O}_3$ via the *forward kinematic map*, g, and its inverse,

$$g(r, l) = \begin{bmatrix} l\cos r \\ l \sin r \end{bmatrix};$$

$$\begin{bmatrix} \theta \\ \rho \end{bmatrix} = g^{-1}(x, y) = \begin{bmatrix} \arctan\, y/x \\ \sqrt{x^2+y^2} \end{bmatrix}, \tag{6}$$

that relate the location of points on the surface of $\mathcal{O}_3$ to its configuration space variable and length out from the pivot as depicted in *Fig. 6*. Thus, $\mathcal{O}_i$ contacts $\mathcal{O}_3$ if and only if

$$\beta_{i3} = [r_3 - \theta(p_i)]^2 \tag{7}$$

vanishes.

The functions β_{ij}, $0 < i, j < 3$ being as in Example 2.1.1.b, the remaining steps follow in exactly the same manner.

2.2. The geometry of motion and forces

Physical machines are built for the purpose of performing *work* in the traditional sense of applying forces over a motion. Having examined the manner in which motions may be constrained, it remains to advance models of how motions may be produced. According to Newton, motions are produced by forces, thus this section offers essentially a consideration of certain force models. Force models are required for various 'synthetic' reasons (as a means of effecting motion constraints) as well as the modeling of 'natural forces' and 'control inputs'. Forces may be modeled in terms of the virtual work they are capable of effecting. That is, given an infinitesimal motion at some placement, $v_p \in T_p\mathcal{P}$, a force at p, F_p is a vector that acts on v_p to produce a scalar, hence, $F_p \in T_p^*\mathcal{P}$. When using configuration space coordinates we shall try to emphasize the dual nature of forces by writing them as row vectors.

2.2.1. Forces of inertia

Since $SE(3)$ posseses a natural bi-invariant Riemannian metric [6], the set of placements $\mathcal{P}$, and, hence, the configuration space $\mathcal{C}$ admits the structure of a Riemannian manifold on each cell of the complex. There is another Riemannian metric on $\mathcal{P}$ arising from the 'dynamical parameters' – the mass distribution over the objects, $\mathcal{O}_i$ – that is inherited by $\mathcal{C}$ as well. We will denote this metric in the local coordinate system defining $\mathcal{C}$ as the positive definite symmetric matrix valued function $M(q)$.

In the Lagrangian formulation of Newtonian dynamics, motion in the absence of external forces arises from the principle of least action. The kinetic energy, a positive definite quadratic form over the velocities that varies smoothly in configuration,

$$\kappa(q, \dot{q}) = \dot{q}^{\mathrm{T}} M(q) \dot{q},$$

may be interpreted as a Riemannian metric, in which sense, the motions resulting from the principle of least action are geodesics [1]. In local coordinates, the motions obtain as solutions to the differential equations arising from the Lagrange-Euler operator,

$$\mathcal{L} \triangleq D_t D_{\dot{q}} - D_q, \tag{8}$$

applied to the kinetic energy, with the result

$$0 = \mathcal{L}\kappa^{\mathrm{T}} = M\ddot{q} + C(q, \dot{q})\dot{q}.$$

In all the examples of this paper, since the holonomic constraints are very simple, the Riemannian metric will have zero curvature so that C

will evaluate to zero. The more general situation is discussed in [16] or [30].

Example 2.2.1.a. *Simple holonomic constraints yield zero Riemannian curvature*

Assuming a uniform distribution of mass throughout $\mathscr{B}$ in *Fig. 2*, the inertia tensor is a constant multiple of the identity matrix,

$$M = \mu \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where μ is the mass. The total energy is given by $\kappa = \mu(p_1^2 + p_2^2)$, hence the motions of the system result from (8)

$$0 = \mathscr{L}\kappa = \ddot{q}^{\mathrm{T}} M.$$

Similarly, in *Fig. 3*, recalling that $q = (p_1, p_2, p_3) \in \mathbb{R}^6$, and denoting the mass of $\mathscr{B}_i$ as μ_i, the kinetic energy of this system is now specified by

$$M = \begin{bmatrix} \mu_1 I_{2\times 2} & & 0 \\ & \mu_2 I_{2\times 2} & \\ 0 & & \mu_3 I_{2\times 2} \end{bmatrix}.$$

Even in *Fig. 5a*, where the holonomic constraints that yield $q = (p_1, p_2, r) \in \mathbb{R}^5$ require the forward kinematic map (6), it is not hard to see that

$$M = \begin{bmatrix} \mu_1 I_{2\times 2} & & 0 \\ & \mu_2 I_{2\times 2} & \\ 0 & & \mu_3 \end{bmatrix},$$

where μ_3 now denotes the moment of inertia of the bar.

In all these cases the Lagrangian dynamics take the same linear time invariant form given above.

The inertial forces cannot maintain the configuration space boundaries, or enforce any of the other constraints on motion explored previously. To remain consistent with the Newtonian dogma that motions can only change in the presence of forces, it becomes necessary to introduce 'reactive' force models that can.

2.2.2. Forces of constraint

Potential forces arise as exact differential one-forms associated with a potential energy – a scalar valued function of configuration, $v : \mathscr{C} \to \mathbb{R}$. Recall that in the presence of potential forces, the equations of motion result from an application of the Euler-Lagrange operator (8) to the difference between kinetic and potential energy, $0 = \mathscr{L}(\kappa - v)$.

Holonomic constraints may be implicitly imposed by use of kinematic maps that parametrize the remaining degrees of freedom as in Example 2.1.2.a, above, or through the use of potential forces that achieve infinite magnitudes away from the constraint manifold [4]. The nonholonomic constraints of contact imposed by the boundary of the freespace (4) may be introduced via potential forces as well. Such forces must achieve infinite magnitude at the boundary. Depending upon the task at hand, they may be assumed to vanish elsewhere or not.

In the context of manipulation, some model of compliance at contact seems essential. A compliance model acts in the neighborhood of the contact set: it presents infinite forces at the freespace boundary and zero forces away from the neighborhood. Thus, to define a compliance potential, it is useful to introduce a class of *smooth but not analytic* scalar 'interpolation' functions,[2] that vanish identically outside some small interval near zero, $\epsilon > 0$, and take a specified value, say K, at zero,

$$\xi_K(\delta) \triangleq \begin{cases} 0 & \delta \geq \epsilon, \\ K & \delta = 0. \end{cases} \tag{9}$$

Example 2.2.2.a. *Compliant contacts*

Suppose that the workplace boundary and the surface of each disk in Example 2.1.1.b has been coated with a stiff elastic material whose thickness extends an ϵ-neighborhood beyond the nominal so that the freespace now consists of those configurations at which $\beta(q) \geq \epsilon$. One imagines that the material introduces a spring-like opposing force in response to compression whose magnitude becomes infinite as the material is compressed to the true boundary. This may be modeled using a smooth interpolating function (9)

[2] While most of the concerns of this paper can be limited to a consideration of smooth (differentiable) functions, an important class of problems involves the seemingly picky added distinction of analyticity. Analytic functions admit a global Taylor series representation and, for example, can never be constant on an open set unless they are everywhere constant.

that vanishes for values greater than ϵ and approaches ∞ near zero. That is, the spring potential,

$$v_\beta(q) \triangleq \sum_{0 \le i,j \le M-1} \xi_\infty \circ \beta_{ij}(q), \tag{10}$$

has a gradient, $-D_q v_\beta$, that enforces the nonholonomic constraint, $\beta(q) > 0$, as follows. Applying the variational operator (8) yields

$$0 = \mathscr{L}(\kappa - v_\beta) = M\ddot{q}^{\mathrm{T}} + D_q v_\beta.$$

Trajectories of this system describe geodesics away from the boundary, and, 'bounce back' under the influence of the gradient force upon intersecting the thickened collar region.

If manipulation is not of interest then the picky details involved in introducing a realistic spring model – for example, the exact functional form of ξ_∞ above – may not be worth the conceptual gain. By passing to a discontinuous model of velocity transition, much of these modeling details may be ignored. In the limit, as the force magnitude becomes infinite, the time over which these reactive contact forces are engaged becomes vanishingly small, and the integral of work done remains a finite value called the 'impulse' [26, §2.5]. Energy is generally lost (the contact forces are not conservative since there is viscous friction) during the impact. Such impulse models give rise to a relation between relative velocities before and after contact governed, for point masses, by a 'coefficient of restitution' [26, §4.2],

$$\left(\dot{p}_{i\mathrm{F}} - \dot{p}_{j\mathrm{F}}\right)_{\mathrm{n}} = -\alpha\left(\dot{p}_{i\mathrm{I}} - \dot{p}_{j\mathrm{I}}\right)_{\mathrm{n}}, \tag{11}$$

where the subscripts I, F, denote initial and final velocities respectively, and the subscript, n, denotes the relative normal component. Generally speaking, one knows the initial velocities and desires to predict the final velocities. This requires an extra constraint, and we will find it most useful to assume that one of the bodies in the collision is so much more massive than the other that its velocity is unchanged, $\dot{p}_{j\mathrm{I}} = \dot{p}_{j\mathrm{F}}$, yielding.

Example 2.2.2.b. *Impulsive contact*: *Batting with a rotary bar*

To apply (11), let p_3 denote the point of contact of the bar with the other body, and let $\dot{p}_3$ denote its linear velocity normal to the bar. Assume in *Fig. 5a* that the bar, $\mathscr{B}_3$, is much more massive than the two bodies $\mathscr{B}_1$, $\mathscr{B}_2$, $\mu_3 \gg \mu_i$, $j = 1, 2$. In this case, when $i = 3$ in (11) we may assume that $\dot{p}_{3\mathrm{F}} = \dot{p}_{3\mathrm{I}}$, yielding

$$\dot{p}_{j\mathrm{Fn}} = \dot{p}_{j\mathrm{In}} + (1+\alpha)\left(\dot{p}_3 - \dot{p}_{j\mathrm{I}}\right)_{\mathrm{n}},$$

or,

$$\dot{p}_{j\mathrm{F}} = \dot{p}_{j\mathrm{I}} + (1+\alpha) nn^{\mathrm{T}}\left(\dot{p}_3 - \dot{p}_{j\mathrm{I}}\right),$$

where n is the normal vector at the point of contact. Adopting the simplified kinematic model underlying, (6), a contact $\beta_{j3}(p_j, r) = 0$ implies that the zero radius object, $\mathscr{B}_j$, touches the bar, $\mathscr{B}_3$ at the point $g(r, \| p_j \|) = p_j$. The unit normal of this contact

$$n = \frac{J_{p_j}}{\| p_j \|}; \qquad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

also specifies the effective linear velocity of the contact point on the bar,

$$\dot{p}_3 = \dot{r}n$$

and it now follows that

$$\dot{p}_{i\mathrm{F}} = \dot{p}_{i\mathrm{I}} + (1+\alpha) J \begin{bmatrix} \cos r_3 \\ \sin r_3 \end{bmatrix} \left(\dot{r}_3 - \frac{\mathrm{d}}{\mathrm{d}t}\theta(p_j) \right)$$
$$\triangleq c(q, \dot{q}). \tag{12}$$

In contrast, a force model of the nonholonomic constraints resulting from dependence between infinitesimals,

$$A(q)\dot{q} = 0,$$

may be obtained by adjoining to them the dynamical system

$$\mathscr{L}(\kappa - v) - \lambda^{\mathrm{T}} A = 0,$$

according to the method of undetermined Lagrange multipliers, λ [21,41]. Under the reasonable assumption that A has full rank for all $q \in \mathscr{Q}$, we may solve explicitly for the multipliers, λ.

Example 2.1.2.b (continued).

Let the inertia tensor be as given in Example 2.2.1.a. For $q = (p, r) = (p_1, p_2, r_0, r_1)$ note from (5) that

$$A(q) = [A_1, A_2](q); \quad A_1 = I_{2\times 2};$$

$$A_2 = \begin{bmatrix} 0, & -\cos r_1 \\ 0 & -\sin r_1 \end{bmatrix},$$

hence the method of undetermined multipliers yields

$$\begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}\begin{bmatrix} \ddot{p} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} \lambda \\ A_2^{\mathrm{T}}\lambda \end{bmatrix}, \qquad \dot{p} = -A_2\dot{r}.$$

Differentiating the second equation,

$$\ddot{p} = -A_2\ddot{r} - \dot{A}_2\dot{r},$$

and substituting $M_1\ddot{p}$ for λ in the second component of the first equation yields

$$(M_2 + A_2^{\mathrm{T}}M_1A_2)\ddot{r} = -A_2^{\mathrm{T}}M_1\dot{A}_2\dot{r}.$$

Since $M_1 = \mu_1 I_{2\times 2}$,

$$A_2^{\mathrm{T}}\dot{A}_2 \equiv 0 \quad \text{and} \quad A_2^{\mathrm{T}}A_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

the equations of motion are now specified by

$$\ddot{r} = 0,$$

$$\dot{p} = \mathrm{c}(r_1)\dot{r}_o;\ \mathrm{c}(r_1) = \begin{bmatrix} \cos r_1 \\ \sin r_1 \end{bmatrix}.$$

2.2.3. Forces imposed by the environment

In addition to inertial and constraint forces, a variety of other 'external' forces may act upon the placed objects,

$$\mathscr{L}(\kappa - v) - \lambda^{\mathrm{T}}A = F_{\mathrm{ext}},$$

and thereby change the nature of the task and hand. A simple example of such an external force is gravity.

Example 2.2.3.a. *Force of gravity*

Suppose that the plane of *Fig. 5a* is inclined nearly vertically. Denote the acceleration of the earth's gravitational field by v_0. Then the potential function due to gravity may be written as

$$v(p_1, p_2, r_3) = v_0(p_{12} + p_{22} + \rho_3 \cos\theta_3),$$

and the resulting force law takes the form

$$F_{\mathrm{grav}} = D_p v = -v_0[0, 1, 0, 1, \rho_3 \sin\theta_3]^{\mathrm{T}}.$$

A more difficult issue is posed by the need to consider friction, $F_{\mathrm{fric}} = G(q, \dot{q})$. The simple viscous damping model for which G is linear in generalized velocity and piece-wise constant in generalized position (the coefficients might change discontinuously over different cells in the contact set) is fine for tasks that occur at relatively high velocities. However, all tasks involving manipulation encounter the much more complicated Coulomb and stiction forces. These contact forces have been the subject of intense scrutiny by mechanicians for many years. One of the most useful treatments from the point of view of robotic manipulation has been given in Donald's thoughtful thesis [19]. For the present purposes, it will suffice to model Coulomb friction as a simple step function of velocity. Thus, our friction model will be

$$F_{\mathrm{fric}} = -G_1\dot{q} - G_0\ \mathrm{sgn}\ (\dot{q}), \tag{13}$$

where it is understood that sgn $(\cdot)$ is applied to each entry of the vector argument.

One important consequence arising from a consideration of friction is the *generalized damper* dynamical model. The Lagrangian model presented thus far represents the application of Newton's laws governing force and change of momentum to the generalized coordinates. The new model, a pre-Gallilean version of physics, associates forces with a change of position,

$$\dot{q} = M^{-1}(q) - F_{\mathrm{ext}}, \tag{14}$$

rather than a change of velocity. Its utility arises in situations where the inertial forces are negligible in comparison to those arising from friction. [3]

Example 2.2.3.b. *The effects of Coulomb friction*

In *Fig. 5a*, suppose, as in Example 2.2.2.b that the bar is much more massive than the other two objects. Suppose, moreover, that it pivots on some nearly frictionless bearing, while, in contrast, $\mathscr{O}_1$ and $\mathscr{O}_2$ must slide over a rough surface

[3] This model was introduced to the robotics literature by Whitney [58] and has been nearly universally adopted within the assembly and fine motion planning literature [35]. In Whitney's setting the justification for this model is the presumption that some very tight local velocity control loop that operates at each freedom. In such a case one assumes that the time constants associated with the acceleration terms of the Lagrangian vector field are sufficiently fast relative to the relevant bandwidth of the task that it may be treated as an algebraic equation, and forces are associated with velocities rather than accelerations (14). In the present setting where some of the objects will not even be actuated as suggested by the input influence matrix, B, introduced below, this assumption can have little merit.

with pronounced Coulomb friction, so that the friction matrices (13) might now look like

$$G_1 = \begin{bmatrix} 0 & & 0 \\ & 0 & \\ 0 & & g_1 \end{bmatrix}; \qquad G_0 = \begin{bmatrix} g_0 & & 0 \\ & g_0 & \\ 0 & & 0 \end{bmatrix};$$

$$g_1 \ll g_0.$$

Further assume that $g_0 \gg \mu_1, \mu_2$, so that objects $\mathcal{O}_1$, $\mathcal{O}_2$, when not in the contact set, effectively experience no motion – the opposing force, g_0/μ_i, $i = 1, 2$ is simply too great. In the portion of the contact set $p_1\mathcal{O}_1 \cap p_2\mathcal{O}_2 \neq \emptyset$ the situation is not very different since $g_0 \gg \mu_1 + \mu_2$. However, when $p_3\mathcal{O}_3 \cap p_i\mathcal{O}_i \neq \emptyset$, $i = 1, 2$ then, assuming $g_0/\mu_3 \ll 1$, motion is indeed possible. This interesting portion of the contact set is given by

$$\mathcal{C}_3 \triangleq \{(p_1, p_2, r) \in \mathbb{R}^5 : \beta_{13}\beta_{23} = 0\},$$

where β_{i3} are derived in (7).

The coefficient of restitution argument employed in Example 2.2.2.b is no longer valid since $\mathcal{O}_1$, $\mathcal{O}_2$ can only be in motion when in contact with $\mathcal{O}_1$. It follows that these two objects must match the velocity of the normal surface of $\mathcal{O}_3$ at their point of contact. Assuming that the coupling due to contact vanishes beyond some ϵ-thickened collar away from $\mathcal{C}_3$ as in (10), use the interpolation function (9) to define the coupling strength

$$\mathrm{c}_j(r, p) \triangleq \xi_1 \circ \beta_{j3}; \qquad j = 1, 2,$$

and note that the effective velocity constraints may now be written in the form

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \dot{r} \begin{bmatrix} \mathrm{c}_1 r \cdot J_{p_1} \\ \mathrm{c}_2 r \cdot J_{p_2} \end{bmatrix}.$$

This situation is depicted in *Fig. 7*. Notice that the model of Coulomb forces we have employed here results in the imposition of two constraints on motion. One is holonomic, and, using the second component of the inverse kinematics (6), asserts that $\dot{\rho}(p_j) = 0$. This implies that we may take as our new configuration space variable $q = (\theta_1, \theta_2, r) \in \mathbb{R}^3$, coordinatizing T^3 – the three-torus. Of course, the nature of the freespace depends upon the initial placements of $\mathcal{O}_1$ and $\mathcal{O}_2$: if their radial extent with respect to the pivot point of $\mathcal{O}_3$ overlaps as in *Fig. 7* then the free configurations are as depicted in *Fig. 3b*. The

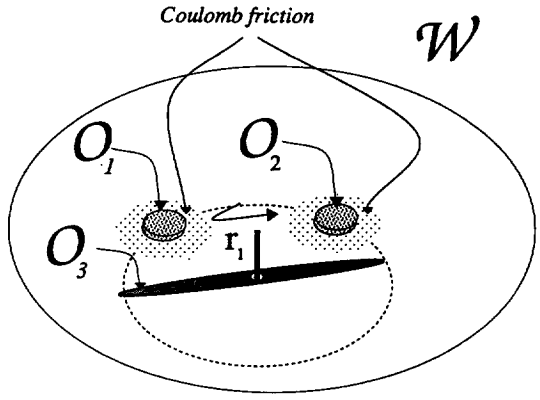


Fig. 7. A rotary arm must approach its workpieces in order to move them.

second constraints is nonholonomic and takes the form $A(q)\dot{q} = 0$ where

$$A(q) = [A_1, A_2](q); \qquad A_1 = I_{2\times 2};$$

$$A_2 = -\mathrm{c}(q) \triangleq \begin{bmatrix} \mathrm{c}_1(r, \theta_1) \\ \mathrm{c}_2(r, \theta_2) \end{bmatrix}. \tag{15}$$

This may be verified after recalling from (6) the relation $\dot{\theta}(p_j) = p_j^{\mathrm{T}} J \dot{p}_j / \| p_j \|$.

Following the continued discussion of Example 2.1.2.b, we must now compute

$$M_2 + A_2^{\mathrm{T}} M_1 A_2 = \mu_3\left(1 + (\mu_1/\mu_3)\mathrm{c}_1^2 + (\mu_2/\mu_3)\mathrm{c}_2^2\right) \approx \mu_3,$$

$$A_2^{\mathrm{T}} M_1 \dot{A}_2 = \mathrm{c}^{\mathrm{T}} \mu_1 \mathrm{c}_1 \cdot \dot{\mathrm{c}}_1 + \mu_2 \mathrm{c}_2 \cdot \dot{\mathrm{c}}_2.$$

Notice that both expressions vanish outside the ϵ-neighborhood of $\mathcal{C}_3$. The first expression is dominated by μ_3 on all configurations (but the second may not be, depending upon the speed of approach to the contact set). We now have

$$\ddot{r} = \frac{1}{\mu_3}\left(\mu_1 \mathrm{c}_1 \cdot \dot{\mathrm{c}}_1 + \mu_2 \mathrm{c}_2 \cdot \dot{\mathrm{c}}_2\right), \tag{16}$$

$$\dot{\theta} = \dot{r}c(r).$$

2.3. The geometry of tasks

Obviously, the problem representations considered above are incomplete since we have explored the constraints on the placements of objects but not the possibility of actively placing them. Ultimately, in order for a machine to take any physical action, it must be capable of exerting forces upon its environment, that is, a selected

profile of external forces, must be applied to the equations of motion. Thus, whatever the language used to express a robot's plan, its implementation must result in a controller. And the effect of adopting that plan will be determined by the manner in which the dynamics integrates the resulting controller. In this section we will finish the sketch of task diversity by first adding an input model to the previously explored examples and then seeking to translate particular goals into the language of limit sets.

2.3.1. Actuation: Robot and environment

The external forces that figure most importantly in robotics are those directly under our control resulting from actuators we have placed on certain objects. We will assume that these actuators provide some set of s 'ideal' sources of force that are capable of delivering instantaneously any vector $u \in \mathscr{U}$ where $\mathscr{U}$ is a bounded and open subset of $\mathbb{R}^s$. We will further assume that in the local coordinates these forces can be applied to the objects in the form

$$\mathscr{L}(\kappa - v) + \lambda^{\mathrm{T}} A = F_{\mathrm{ext}}(q, \dot{q}) + F_{\mathrm{act}}(q, \dot{q}, u), \tag{17}$$

where for all $(q, \dot{q}) \in T\mathscr{C}$, $F_{\mathrm{act}}(q, \dot{q}, \cdot)$ is assumed to be injective with respect to the second argument, so that none of the degrees of actuation freedom is ever 'wasted'. On the other hand, it is only in the exceptional case that F_{act} is surjective – that is, only rarely will all objects be fully actuated.

Example 2.3.1.a. *Fully actuated objects*

Suppose $\mathscr{O}$ in *Fig. 2a* is endowed with two jet engines that impart bounded but arbitrary forces in orthogonal directions. Then in the coordinate system $q = (p_1, p_2)$ we have $\mathscr{U} = \mathbb{R}^2$

$$F_{\mathrm{act}} = u = [u_1, u_2].$$

For $x = (q, \dot{q})$ we have

$$\dot{x} = \begin{bmatrix} x_2 \\ -M^{-1}u \end{bmatrix}, \tag{18}$$

a set of two decoupled double integrators. Similarly, in *Fig. 3a*, consider the situation that each object is endowed with two jet engines. Then in the coordinate system $q = (p_1, p_2, p_3)$ we have $\mathscr{U} = \mathbb{R}^6$

$$F_{\mathrm{act}} = u = [u_1, u_2, u_3].$$

For $x = (q, \dot{q})$ we have

$$\dot{x} = \begin{bmatrix} x_2 \\ M^{-1}(-Dv_\beta - u) \end{bmatrix}. \tag{19}$$

Evidently, the same collection of objects may give rise to radically different task settings depending upon the actuation structure, F_{act}.

Example 2.3.1.b. *Two degree of freedom dual assembly*

Instead, in *Fig. 3a*, consider the situation that only $\mathscr{O}_3$ is endowed with actuators, $F_{\mathrm{act}} = [0, 0, u]$. Let $x = (x_1, x_2) \triangleq (p_3, \dot{p}_3)$ and $y = (y_1, y_2)$; $y_1 \triangleq (p_1, p_2)$; $y_2 \triangleq \dot{y}_1$. Then the system has the dynamics

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{1}{\mu_3}(-D_{x_1} v_\beta(y_1, x_1) - u) \end{bmatrix},$$
$$\dot{y} = \begin{bmatrix} y_2 \\ \begin{bmatrix} 1/\mu_1 & 0 \\ 0 & 1/\mu_2 \end{bmatrix}(-D_{y_1} v_\beta(y_1, x_1)) \end{bmatrix}. \tag{20}$$

Similarly, suppose in *Fig. 5a* that there is a direct drive motor placed at the pivot joint of the segment, $\mathscr{O}_3$. Using the coordinates introduced in Example 2.1.2.a, let $x = (r, \dot{r})$ and y be as in the previous paragraph. Then the system has dynamics given by (20) with the exception that v_β is replaced by $v_\beta + v_{\mathrm{grav}}$.

In general, we will refer to the actuated degrees of freedom in the objects as the 'robot', x, and the remaining unactuated degrees of freedom, y, together with the immobile workplace as the 'environment'.

Example 2.1.2.b (continued).

Suppose that in addition to a direct drive motor at the center of the line segment in *Fig. 5b* capable of changing the orientation, there is an actuator that exerts torque along the vehicle's wheel measured by r_0. Then in the coordinate system $q = (p_{31}, p_{32}, r_0, r_1)$ we have

$$F_{\mathrm{act}} = [0, 0, u_0, u_1]^{\mathrm{T}}.$$

The robot is the steering wheel and drive wheel $x = (x_1, x_2)$; $x_1 = (r_0, r_1)$; $x_2 = \dot{x}_1$; the environment consists of the translational degrees of freedom of $\mathscr{O}$ $y = (p_{31}, p_{32})$, together with the workplace, $\mathscr{W}$. We have

$$\dot{x} = \begin{bmatrix} x_2 \\ \dfrac{1}{\mu_3} u \end{bmatrix}, \qquad (21)$$
$$\dot{y} = \mathrm{c}(x_{12}) x_{21}.$$

Just as apparently similar geometries can give rise to very different dynamical equations, it is curious that surprisingly different settings can yield similar dynamics.

Example 2.3.1.c. *One degree of freedom dual assembly*

Suppose in Example 2.2.3.b that the rotating bar is raised above the plane of the other two objects and endowed with some 'perfect gripper' that is capable of holding them fast (e.g., it contains an electromagnet and the objects are metal) with a force u_0. Let the pivot joint be actuated by a source of torque, r. The effect of u_0 is to mediate the nonholonomic constraints which are as written in (15) with the exception that $A_2 = u_0 \mathrm{c}$. Thus, substituting in (16), if

$$r \triangleq u_1 - \mu_1 \frac{\mathrm{d}}{\mathrm{d}t}(u_0 \mathrm{c}_1) + \mu_2 \frac{\mathrm{d}}{\mathrm{d}t}(u_0 \mathrm{c}_2),$$

letting $x = (r, \dot{r})$ and $y = (\theta_1, \theta_2)$ yields

$$\dot{x} = \begin{bmatrix} x_2 \\ \dfrac{1}{\mu_3} u_1 \end{bmatrix}, \qquad (22)$$
$$\dot{y} = \mathrm{c}(x, y) x_2 u_0.$$

2.3.2. Task encoding

Plans are written in languages. The history of robot languages over the last two decades reveals an overwhelming concern for syntactical systems that (to the extent they have any formal properties at all) admit the model of a Turing machine. This is very odd, since one would think on the face of things that robot languages ought to bear some relation to the properties of actuated mechanisms. Hopefully, the preceding discussion has made clear that the fact that any formal language taking as a model the physical behavior of actuated mechanisms will necessarily involve propositions concerning the behavior of dynamical systems on manifolds. A case can be made, then, for seriously re-thinking the design of robot languages.

While a general program for undertaking this re-design seems quite daunting at present, there is at least one specific aspect of the endeavor whose implications seem clear. To the extent that a 'goal' represents the desired end result of some sequence of robot actions and environmental responses, it seems most appropriate to express these in terms of the notion from dynamical systems theory of limit sets [24,42]. Namely, if $\mathscr{G} \subset \mathscr{E}$ is the desired goal set, then we would like the input (17) to result in a dynamical system whose limit sets is $\mathscr{G}$.

Example 2.3.1.a (continued).

The single robot or multiple robots navigating amidst the obstacles presented by the workplace, $\mathscr{W}$, have the very obvious goal of arriving at some destination, $d \in \mathscr{O}$ while never colliding with the obstacles. It is necessary that a navigation plan be furnished for every configuration in the freespace, but the velocities along the way are of little concern. Thus, it suffices to build a dynamical system on $T\mathscr{O}$ whose motions converge to $(d, 0)$ while remaining in any subset of $\mathscr{E}$ that contains a neighborhood of the 'zero velocity set', $\mathscr{O} \times 0$.

Take for place space coordinates $x = (p, \dot{p})$. Then the robot in its environment is specified by the dynamical system

$$\dot{x} = \begin{bmatrix} x_2 \\ -M^{-1} u \end{bmatrix}, \qquad (23)$$

where we require $x \in \mathscr{E} \triangleq \{(p, \dot{p}) \in \mathbb{R}^4 : \beta(p) + \dot{p}^{\mathrm{T}} \dot{p} \leq 0\}$.

The task specification for the Unicycle Example 2.1.2.b and the One Degree of Freedom Assembly Example 2.3.1.c takes exactly the same form as the 'navigation' problem for Example 2.3.1.a formulated above. Of course, the nature of the solutions will be quite difficult because of the very different tangent bundle constraints that operate – this will be seen in the sequel. In the case of the Two Degree of Freedom Assembly Example 2.3.1.b the situation appears to be rather different since the 'robot' is incapable of imparting motion to the unactuated bodies in the envi-

ronment without some form of collision. If the generalized damper dynamical model is adopted then this situation fits within the traditional quasi-static manipulation context that has been explored quite extensively in the literature. This domain of tasks includes such problems of insertion [58,35] or pushing [38,37] wherein consideration of the full cell complex is necessary [9]. Moreover, additional structure must be added to account for frictional and jamming (contact degrees of freedom) reaction forces that the environment may impose upon the robot [20,19]. Since this task domain has been carefully studied in the literature, it seems unnecessary to add more here.

However, if, in problem settings like those of Example 2.3.1.b, a dynamical rather than quasistatic model is introduced, the task changes dramatically. It is fair to say that such problems have received almost no attention in the literature. We will consider in the sequel a small piece of this setting, anticipated in Example 2.2.2.b where the Lagrangian dynamics is adopted on $\mathscr{T}$ but a simpler impulse model is placed over the contact set instead of the continuous compliance model of Example 2.2.2.a.

The impulse model of contact seems a reasonable starting place for tasks like playing ping-pong [2] or walking and running [44,39] or juggling [13] and catching [28]. These all require explicit attention to the dynamics of the task.

Example 2.3.2.a. *A juggling problem*

Suppose within the setting of *Fig. 5a* we impose the batting model of Example 2.2.2.b. The dynamics of the unactuated objects, away from the contact set (that is, where Dv_β vanishes), may be simplified from (20) as

$$\dot{y} = \begin{bmatrix} y_2 \\ M_2\tilde{a} \end{bmatrix};$$

$$\tilde{a} = Dv_{\text{grav}} \begin{bmatrix} 0 \\ I_{4\times 4} \end{bmatrix} = -v_0[0, 1, 0, 2].$$

A time sampled version of the flight of one object under these dynamics is given as

$$y_{j+1} = F^s(y_j) \triangleq A_s y_j + a_s;$$

$$p_j = Cw_j$$

$$A_s \triangleq \begin{bmatrix} I & sI \\ 0 & I \end{bmatrix}; \qquad a_s \triangleq \begin{bmatrix} \frac{1}{2}s^2\tilde{a} \\ s\tilde{a} \end{bmatrix}; \qquad C = [I, 0]. \tag{24}$$

Suppose that the robot strikes the ball in state $y_j = (p_j, \dot{p}_j)$ at time s with a velocity at normal v_j. Taking the composition of the previous sampled system with the collision model (12), yields the 'environmental control system'

$$y_{j+1} = f(y_j, v_j, t_j) \triangleq A_{t_j} y_j + a_{t_j} + \begin{bmatrix} t_j \mathrm{c}(y_j, v_j) \\ \mathrm{c}(y_j, v_j) \end{bmatrix}, \tag{25}$$

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

$$f: \mathscr{Y} \times \mathscr{V} \times \mathbb{R} \to \mathscr{Y},$$

with control inputs $u \in \mathscr{U} \triangleq \mathscr{V} \times \mathbb{R}$ (v_j and t_j).

Probably the simplest systematic behavior of this environment imaginable (after the rest position), is a periodic vertical motion of the puck in its plane. Specifically, we want to be able to specify an arbitrary 'apex' point together with a vertical position, and from arbitrary initial puck conditions, force the puck to attain a periodic trajectory which impacts at zero vertical position and passes through that apex point. We call this task the *vertical one-juggle*. Such tasks are exactly represented by the choice of a desired fixed point, y^*.

When assumptions about the rigidity of the world are abandoned ('fold the laundry') then the spaces introduced above have infinite dimension, but analogous structure [1]. Such problems far exceed the scope of this paper.

3. Navigation problems

Consider the class of problems represented by Example 2.1.1.a or Example 2.1.1.b. A robot – a fully actuated object or set of objects – moves 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. The environment, $\mathscr{E}$, is some compact subset of $T\mathscr{Q}$ that includes a neighborhood of the zero velocity configuration space, $\mathscr{Q} \times 0$. The goal set, $\mathscr{G}$, 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 $\mathscr{G}$ from

as large a set of initial configurations as possible while remaining in $\mathscr{E}$.

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 $\mathscr{F}$ as the destination. In the present formulation the robot must arrive (with probability one) at the goal if a path exists. Thus one can 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 environment, $\mathscr{E}$, may be very difficult to obtain in practice, even when $\mathscr{W}$ and $\mathscr{O}$ are perfectly known (which, of course, neither might be in the real world). Yet this work presumes that exact information concerning the boundary components of $\mathscr{E}$ is available.

3.1. 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 [25], the idea of using artificial potential functions for robot task description and control was adopted or re-introduced independently by a number of researchers [40,3,43]. 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.

3.1.1. A practicable global stability mechanism

In the present context, the utility of artificial potential functions for path planning rests upon the possibility 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 [24,31]. 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, $\mathscr{F}$. Of course, the appropriate planning space is $T\mathscr{P}$, 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 (17) on $T\mathscr{P}$ [32].

3.1.2. 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, $\mathscr{G}$, from almost every initial configuration. This list comprises the notion of a *navigation function* introduced to the literature two years ago [45].

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 [33]. 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.

3.1.3. *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 [7] as well.

The relevant invariant in navigation problems is the topology of the underlying configuration space [29]. In this regard, the significant virtue of the navigation function is that its desirable properties are invariant under diffeomorphism [33]. 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.

3.2. *The construction of navigation functions*

This section presents the highlights of our constructive work to date in actually building navigation functions for concrete problems. Roughly speaking, progress to date is as follows. We have essentially solved all instances the situations depicted in *Fig. 2a* and *Fig. 4a* where there is a single object with trivial geometry and an arbitrarily complicated environment, $\mathscr{W}$ [49]. We have made some progress in the situation depicted in *Fig. 3a* or *Fig. 8* where there are multiple objects with trivial geometry [57,27]. We only begun to scratch the surface of problems such as the one in *Fig. 4b* where there are objects with nontrivial geometry [48].

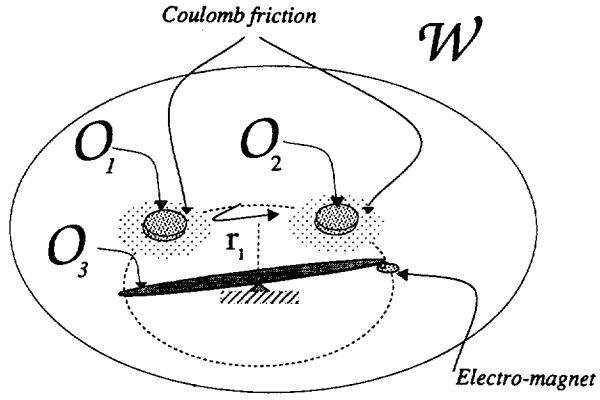


Fig. 8. A rotary arm must approach its workpieces in order to move them.

3.2.1. *A 'model' problem*

A 'Euclidean sphere world' is a compact connected subset of E^n whose boundary is the disjoint union of a finite number, say $M+1$, of $(n-1)$ spheres as depicted in *Fig. 1*. We suppose that perfect information about his 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, $\beta : \mathscr{M} \to [0, \infty)$ as in (2). 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

$$\hat{\varphi} \triangleq \frac{\gamma}{\beta}.$$

Of course, $\hat{\varphi}$ 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,

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

Note that the composition

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

is everywhere smooth and bounded, and attains its maximal height of unity only on the boundary

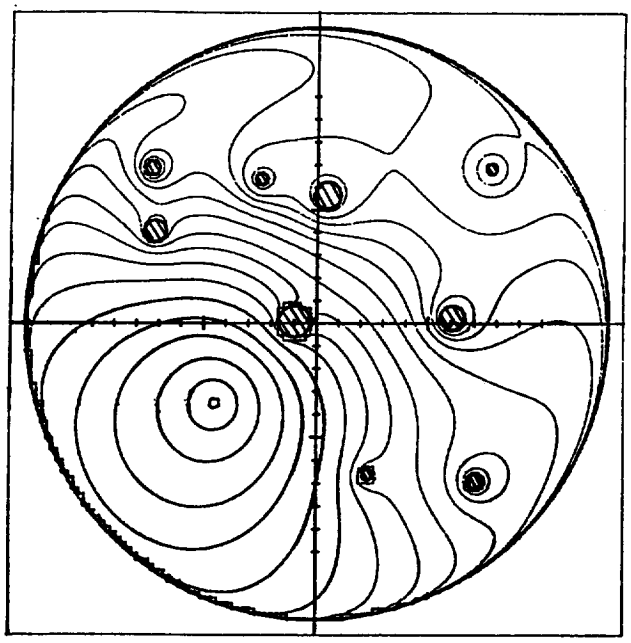


Fig. 9. Planar sphere world with nine internal obstacles [50]. Contour lines denote the level curves of a navigation function constructed according to Theorem 1.

components of the configuration space. For technical reasons we find it necessary to take the kth root of this ratio with the following result.

Theorem 1 ([33]). *If the configuration space is a Euclidean sphere world then for any finite number of obstacles, and for any destination point in the interior,*

$$\varphi = \sigma_d \circ \sigma \circ \hat{\varphi} = \left(\frac{\gamma^k}{\gamma^k + \beta} \right)^{1/k}, \tag{26}$$

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

That is, φ is a navigation function. The function, N, on which the theorem depends is given explicitly in [33].

3.2.2. A class of coordinate transformations

Of course, as seen in the previous section, a freespace with the simple geometry depicted in *Fig. 1* is quite rare even when both $\mathcal{O}$ and $\mathcal{W}$ are described by euclidean disks.

A *star shaped set* is a diffeomorph of a Euclidean n-disk, 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 E^n whose boundary is the disjoint union of a finite number of star shaped set boundaries. Now suppose the availability of an implicit representation for each boundary component: that is, let β_i be a smooth scalar value function that is positive outside, negative inside, and vanishes on the boundary of the ith 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 [47]. A suitable Euclidean sphere world model, $\mathcal{M}$, is explicitly constructed from this data. That is, one determines (p_j, ρ_j), the center and radius to the jth obstacle) of the jth star shaped obstacle.

A transformation, $h: \mathcal{M} \to \mathcal{F}$, may now be constructed in terms of the given start world and the derived model sphere world geometrical parameters as follows. Denote the 'jth omitted product', $\prod_{j=0}^{M} \beta_j$ as $\bar{\beta}_j$. The 'jth analytic switch', $\sigma_j \in C^\omega[\mathcal{F}, \mathbb{R}]$,

$$\sigma_j(q, \lambda) \triangleq \frac{x}{x+\lambda} \circ \frac{\gamma d\bar{\beta}_j}{\bar{\beta}_j} = \frac{\gamma d\bar{\beta}_j b}{\gamma d\bar{\beta}_j b + \lambda \bar{\beta}_j},$$

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

$$\nu_j(q) \triangleq \rho_j \frac{1 + \bar{\beta}_j(q)}{\| q - q_j \|},$$

scales the ray starting at the center point of the jth 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 jth model obstacle – a suitable sphere. The overall effect is that the complicated star shaped obstacle 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

$$h_\lambda(q) \triangleq \sum_{j=0}^{M} \sigma_j(q, \lambda)\left[\nu_j(q) \cdot (q - q_j) + p_j\right] + \sigma_d(q, \lambda)\left[(q - q_d) + p_d\right], \tag{27}$$

where σ_j is the jth analytic switch, σ_d is defined by

$$\sigma_d \triangleq 1 - \sum_{j=0}^{M} \sigma_j, \tag{28}$$

and ν_j is the jth star set deforming factor. The 'switches', make h look like the jth deforming factor in the vicinity of the jth obstacle, and like the identity map away from all the obstacles boundaries. With some further geometric computation we are able to prove the following.

Theorem 2 ([47]). *For any valid star world, $\mathscr{F}$, there exists a suitable model sphere world $\mathscr{M}$, and a positive constant Λ, such that if $\lambda \geq \Lambda$, then*

$h_\lambda : \mathscr{F} \to \mathscr{M}$,

is an analytic diffeomorphism.

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

3.2.3. Navigation functions for geometrically complicated spaces

In general, the freespace will not be a star world, as *Fig. 2b* is not. In a recent paper [49], we show how to extend the class of coordinate transformations developed above to a class of spaces that includes almost all instances of freespace arising from the placements of a spherically symmetric object in a planar world. Briefly, consider an obstacle, a disconnected subset of the workplace, $\mathscr{W}_i \subset \mathscr{W}$ which is a union of several intersecting stars. The arrangement of the stars in $\mathscr{W}_i$ 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 [46] then say that the obstacle is a *free of stars; a forest of trees of stars* is a freespace, $\mathscr{F}$, 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 [50]. Such a forest, $\mathscr{F}$ has a *purged version*, $\hat{\mathscr{F}}$, defined to be $\mathscr{F}$ with the leaves in trees consisting of more than one star filled-in and 'reattached' to $\mathscr{F}$. Using the ideas presented above, we have shown how to define a change of coordinates from any forest, $\mathscr{F}$, to its purged version, $\hat{\mathscr{F}}$, [46]. Successively 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. *Fig. 1* 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_{Λ_i}, 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 previously constructed star-world to sphere-world transformation [47] may now be used to obtain the corresponding model sphere world, M. Thus the sequence of transformations is

$$\mathscr{F} \xrightarrow{f_{\Lambda_1}} \mathscr{F}_1 \xrightarrow{f_{\Lambda_2}} \mathscr{F}_2 \xrightarrow{f_{\Lambda_3}} \mathscr{F}_3 \xrightarrow{f_{\Lambda_4}} \mathscr{F}_4 \xrightarrow{h} \mathscr{M}.$$

Each 'intermediate' purged version is depicted in the figure. In each space, the level lines of the navigation function as it is 'pulled back' via composition is shown as well. The destination point is chosen arbitrarily at the origin, and the level lines

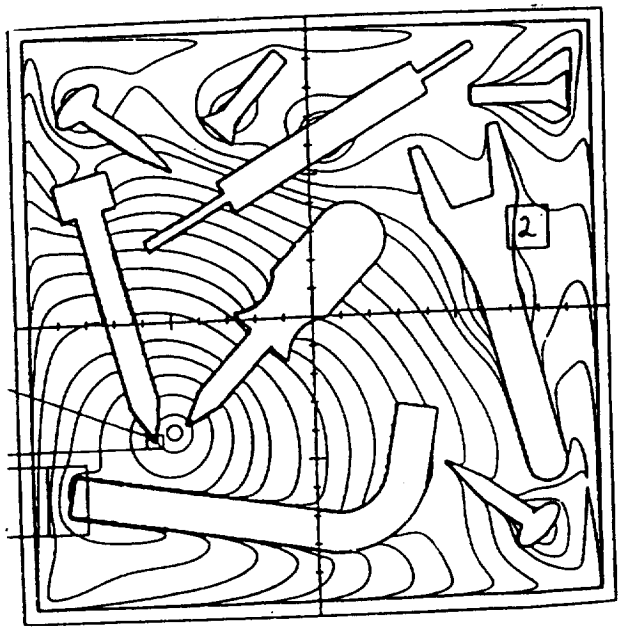


Fig. 10. Planar star world with nine internal obstacles [50]. The contour lines are level curves of a navigation function induced by diffeomorphism according to Theorem 2, modified to take account of the 'sharp corners' [49]. The model sphere world is depicted in the previous *Fig. 9*.

vary between zero (at the destination point), and one (on all the boundary components).

3.3. Navigation functions for multiple robots

In a recent set of papers [57,56] we have extended the navigation work to problems involving multiple moving bodies as in Example 2.1.1.b. *Fig. 12* 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

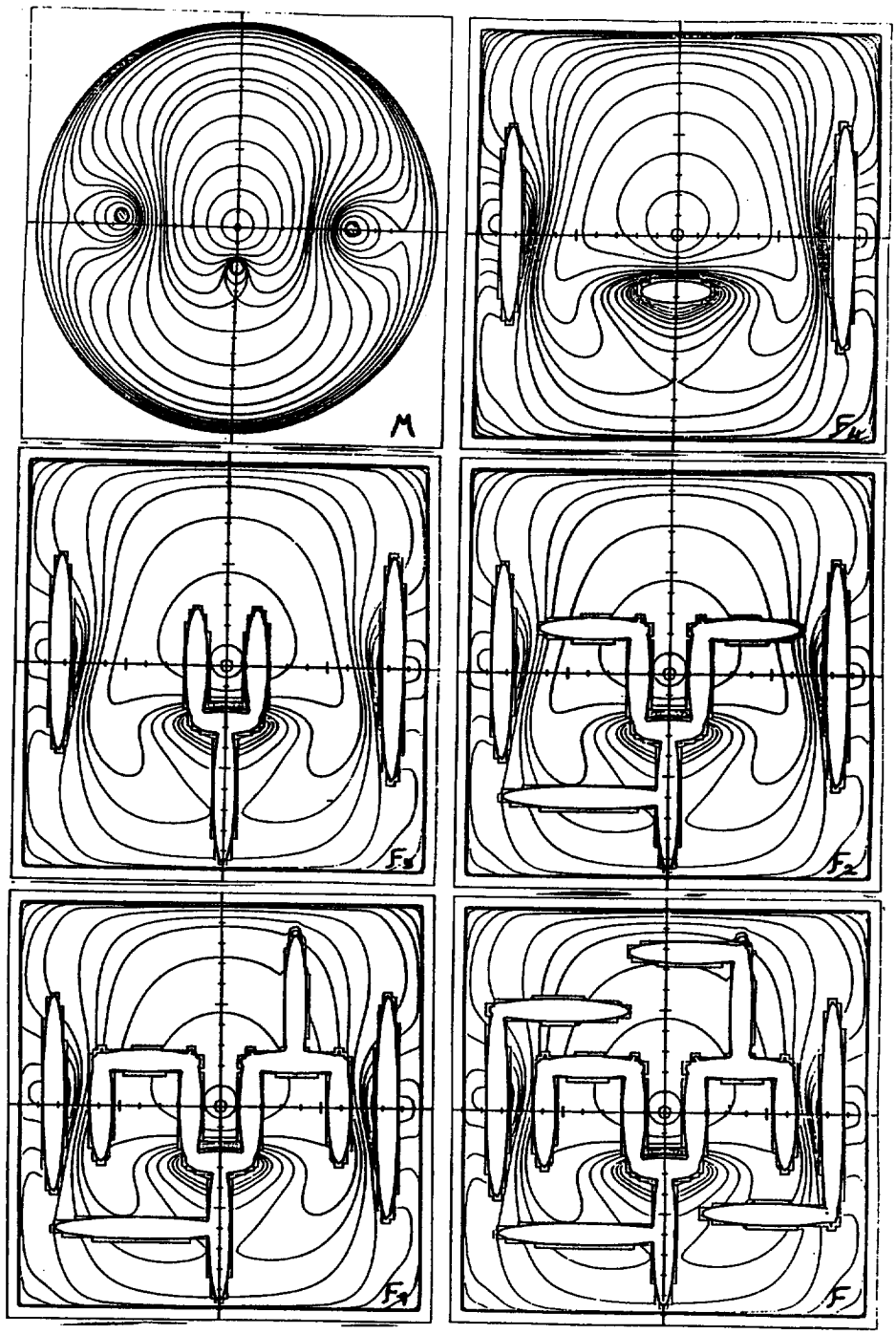


Fig. 11. Planar forest of stars with three internal tree-like obstacles (bottom right), its 'purged' versions, and its model sphere world (top left) [46].

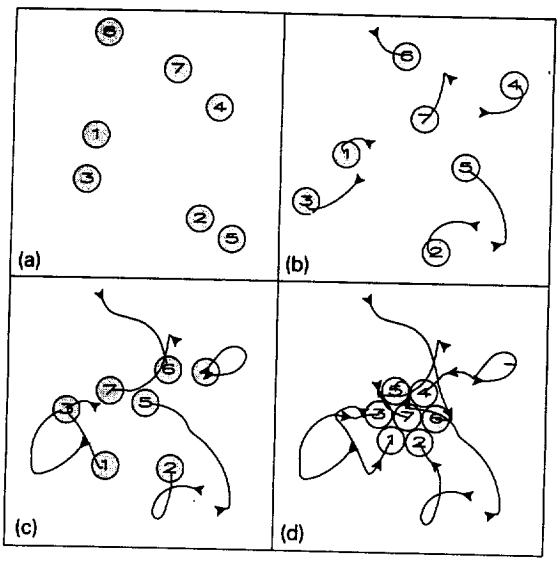


Fig. 12. Coordinated motion of seven independently actuated spherically symmetric robotics toward a goal configuration: Balls indicate current configuration; curves indicate the ball paths. Frame (a) shows random initial configuration, (b) and (c) are intermediate configurations, and (d) shows final (desired) configuration [56].

initial and desired states are 'legal' – no balls touch and all reside within the boundary. *Fig. 12* shows a typical 7 ball motion sequence from (random) initial configuration to 'assembled' final configuration.

A function, φ, is easily constructed according to the standard recipe for such geometrically simple environments, $\mathscr{E}$, given in (26) where, in this case, the function $\gamma(b)$ is given by

$$\gamma(b) = \sum_{i=0}^{i=n} \gamma_i(b); \qquad \gamma_i(b) = \tfrac{1}{2} \| b_i - d_i \|^2, \tag{29}$$

and $\beta(b)$ is given as in (2). According to our rather extensive simulation studies the performance of this construction is surprisingly good – for example φ 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) [57]. 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 *Fig. 12*. We have shown [27] that this construction is indeed a navigation function for high enough values of k when the dimension of the ambient space of the objects in one, for example, as depicted in *Fig. 8*.

4. 'Assembly' problems

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 *Fig. 12* has greater dimensional complexity than that of *Fig. 8*, the latter introduces a heretoforce unexplored problem. Although the robot is actuated by a force that is under our control, the bodies cannot be actuated until the robot approaches them and engages them with its gripper.

This section explores the two concrete instances of the assembly problem introduced in Example 2.3.2.a and Example 2.3.1.c. 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, in both cases, passage to an abstracted 'planning system' – a discrete dynamical controlled process that takes place on the contact set – affords a similar procedure.

4.1. The necessity of a hierarchical representation

4.1.1. Controllable but unstabilizable dynamics

The possibility of solving assembly problems obviously depends on the fact that the dynamical coupling between degrees of freedom is not fixed. More precisely, the allowable infinitesimal motions of the body systems differ from configuration to configuration. One significant feature of such problems is that even while the control systems that arise are completely controllable,

Brockett's [8] stabilizability condition is not met [5], and there is no possibility of constructing a single feedback controller to stabilize an isolated equilibrium state. For example, in Eq. (20) of Example 2.3.1.b $\dot{y} \equiv 0$ away from (the ϵ-thickened neighborhood around) the contact set (since the interpolating function, ξ_∞ from (10) vanishes), hence the vector field is not surjective away from $\mathscr{C}$. Similarly, in (21) and (22), $\dot{y} \equiv 0$ when $x_2 = 0$ and the vector field is not surjective at any zero velocity configuration. The failure of surjectivity implies the impossibility of continuous feedback stabilization. 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 [27].

4.1.2. 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 the contact set, $\mathscr{C}$, in organizing the design and informing the analysis of hierarchical feedback controllers for autonomous manipulation for assembly. Roughly speaking, we are led to re-define a task in terms of the body 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, 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 [44], 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 [28]. The second obtains from our on-going studies of juggling that will form the chief source of examples in the sequel [13,11,51]. 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 [27,56], 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 *return maps* on a Poincaré section and is quite conventional within the dynamical systems literature [22]. However, while a robot that brings a disorganized collection of bodies into an assembly must necessarily visit and re-visit 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.

4.2. Assembly to a periodic orbit through batting

Problems involving batting, in particular, the juggling work described here, are interesting in the their own right, but also touch upon broader issues in dexterous manipulation. Analysis identical to that reviewed here [28] provides an explanatory account of why (a greatly simplified version of) Raibert's hopping algorithm [44] works. In some sense this can hardly be surprising since we have borrowed Raibert's encoding methodology – the idea of servoing around a mechanical energy level to produce a stable limit cycle – in the juggling research. Our chief contribution is to have explicitly worked out the identification and to have offered an explanation in the form of a practicable global stability mechanism for its success. The extension of these ideas to the problem of juggling two bodies simultaneously may similarly turn out to have significance with respect to problems of gait in legged locomotion.

Presumably, our robot 'settles down' to a characteristic steady state juggling pattern because that pattern is an attracting periodic orbit of the closed loop robot–environment dynamics. Very likely, similar 'natural' control mechanisms would make good candidates for gait regulation. Catching studies [12] closely related to juggling (but not presented here) have implications for general robot manipulation of objects in the absence of 'guarded moves'. Prior to the static grasp phase wherein the myriad robot degrees of freedom may be simultaneously engaged to control a (typically) six degree of freedom object there must be a 'fumble' phase – a series of controlled collisions involving unpredictable combinations of the robot link surfaces and the surfaces of the object. During a fumble, far fewer robot degrees of freedom may be engaged with the environment, and only intermittently.

In the following discussion, we will pursue only the specific case of Example 2.3.2.a depicted in *Fig. 5b*. Beyond the extensions of legged locomotion and dynamical manipulation touched upon above, it may be verified by consulting some of our other juggling reports that these ideas generalize in an essentially straightforward manner to the apparently more difficult spatial version of this specific problem depicted in *Fig. 13* [52,51].

4.2.1. Encoding a simple juggling task

The problem of batting a single object into a specified periodic vertical orbit was posed in Example 2.3.2.a as a setpoint regulation problem for a discrete-time nonlinear system (25). Being interested in sensor based manipulation we focus upon solving such problems with feedback based controllers. Thus, a robot feedback strategy is a map $g: \mathcal{Y} \to \mathcal{U}$, from the body's state to the robot's action set, $\mathcal{U}$, resulting in the impact strategy $u_j = g(y_j)$. The robot-environment *closed loop dynamical system* is formed from the composition of f in (25) with g,

$$y_{j+1} = f_g(y_j); \qquad f_g(y) \triangleq f(y, g(y)). \tag{30}$$

4.2.2. A practicable global stability mechanism

Say that an abstract feedback law for (25), $g: \mathcal{Y} \to \mathcal{V} \times \mathbb{R}$, is a verticle one-juggle strategy if it induces a closed loop system, $f_g(y)$, for which $y^* \in \mathcal{Y}$ described in Example 2.3.2.a is an asymp-

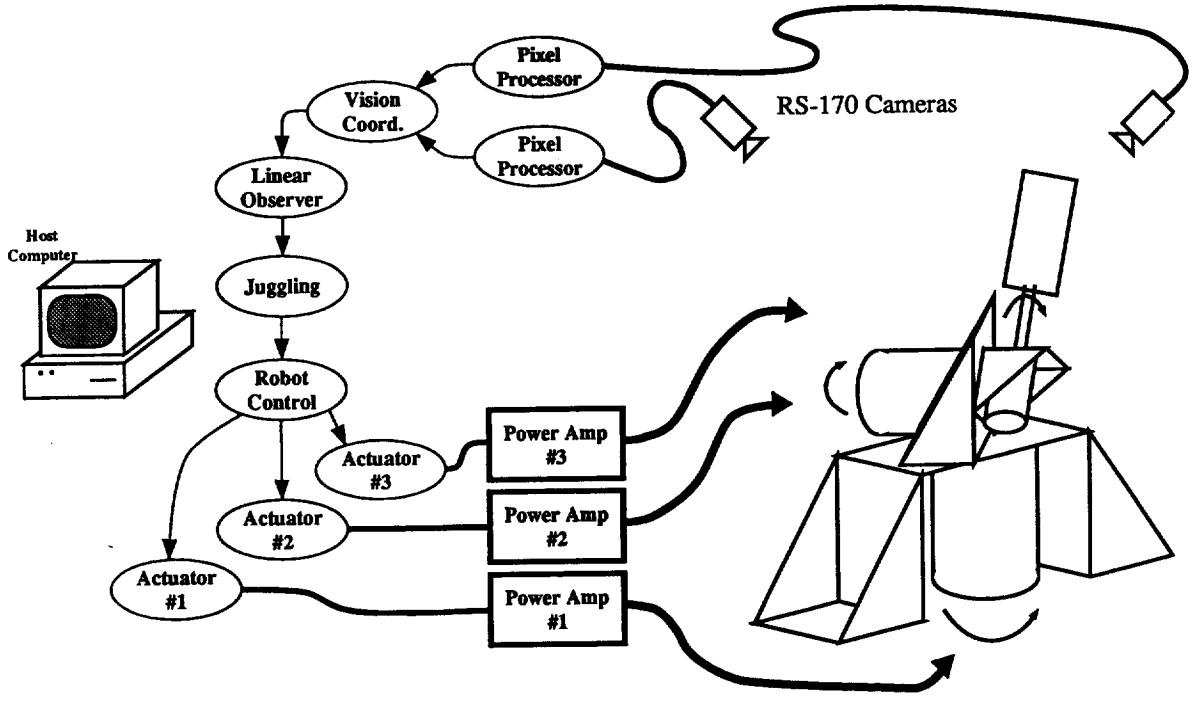


Fig. 13. The Yale spatial juggler.

totically stable fixed point. For our original planar machine [10] it was shown that the linearized environmental control system (25) was controllable around very vertical one juggle task. However, experiments 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 [10]. 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.

In contrast to the notion of energy dissipation that has been known for more than a century [54], 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 [53] and Guckenheimer [23]. 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* [53].

Singer showed that S-unimodal maps can have at most one attracting periodic orbit [53]. 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 [23]. 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.

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 [18,14].

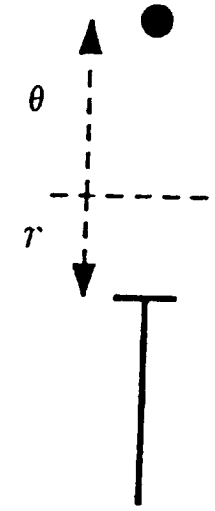


Fig. 14. The line-juggler model.

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.

4.2.3. A continuous controller implementation

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 *Fig. 14*.

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

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

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

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

$$\dot{r} = -\kappa_{10}\dot{\theta}. \tag{31}$$

For simplicity, assume that the desired impact position is always selected to be $\theta^* = 0$. 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 $c(\dot{\theta}^*, \dot{r}(\dot{\theta}^*)) = -\dot{\theta}^*$ for κ_{10} using the one dimensional analogue of (12) [13], 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', κ_{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

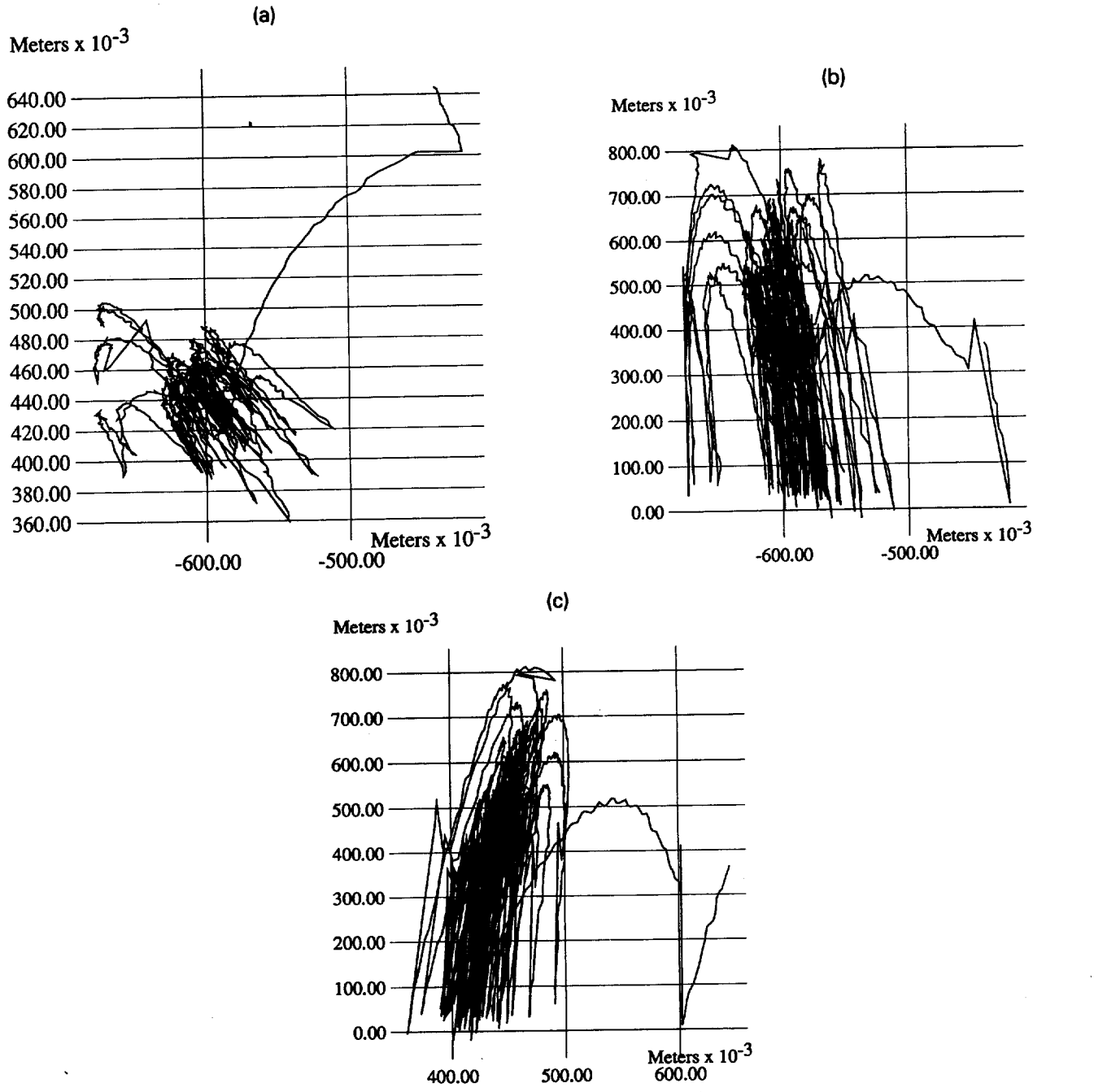


Fig. 15. One-juggle ball trajectory: (a) X–Y projection (b) X–Z projection and (c) Y–Z projection [51].

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 [44], 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,

$$\eta(y) = \tfrac{1}{2}\dot{\theta}^2 + \gamma\theta.$$

This suggests the addition to the original mirror trajectory,

$$r = -\kappa_1(w)\theta;$$

$$\kappa_1(y) \triangleq \kappa_{10} + \kappa_{11}[\eta(y^*) - \eta(y)], \tag{32}$$

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.

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

$$u_1 = \frac{2}{\gamma}\dot{\theta}' \quad \text{and} \quad u_2 = \dot{r} = -\kappa_1\dot{\theta}.$$

Substituting these robot control inputs yields the scalar map of puck impact velocities just before impact at the invariant position $\theta^* = 0$,

$$f(\dot{\theta}) = \dot{\theta}\left(1 - \beta(\dot{\theta}^2 - \dot{\theta}^{*2})\right), \tag{33}$$

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

It is not hard to show [11] that (33) satisfies the conditions of 'S-unimodality' described above. A check of (33) reveals that the fixed point is locally stable when

$$0 < \beta < \frac{2/\zeta - 1}{\dot{b}^{*2}}. \tag{34}$$

There immediately follows,

Theorem 3 ([14]) *The mirror algorithm for the line-juggler results in a successful vertical one juggle which is essentially globally asymptotically stable as long as β satisfies the inequality* (34).

We have shown a gratifying correspondence between theoretical predictions based upon the Singer–Guckenheimer results, simulation studies, and physical data [13,10]. Perhaps the most dramatic depiction of this correspondence is suggested by our bifurcation studies confirming the essential nonlinear nature of the stability mechanism [11]. The mirror algorithm is readily generalized to situations involving higher degrees of freedom as in *Fig. 5a* or *Fig. 13* as

$$r(t) = m(p(t), \dot{p});$$

$$m \triangleq \kappa_1 g^{-1}(p) + \kappa_2(p, \dot{p}), \tag{35}$$

where g^{-1} is the 'inverse kinematics' of the robot's point of contact with the body (6), κ_1 is a gain function analogous to that defined above that measures the body's total energy relative to the desired task energy, and κ_2 contains 'servo' terms that regulate horizontal errors around the desired vertical motion [13,52].

Fig. 15 shows the three projections of the ball's trajectory for a typical run of the mirror algorithm implemented on the apparatus depicted in *Fig. 13*. As can be seen the system is capable of containing the ball within roughly 15cm of the target position above the floor.

4.2.4. Assembly of multiple objects by batting

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 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 was 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 [12] to the spatial system, with the hope that they would prove adequate.

In order to control the temporal separation of the balls we introduce a new variable, *ball phase*,

$$\epsilon(y) \triangleq -\frac{\dot{\theta}_2}{\sqrt{2\eta}}, \tag{36}$$

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,

$$e_{ph}(y_0, y_1) = [\epsilon(y_0) - \epsilon(y_1)]^2 - 1. \tag{37}$$

This measure may be used to modify the energy regulation term of the mirror law,

$$\kappa_1 = \kappa_{10} + \kappa_{11} e_\eta + \kappa_{12} e_{ph},$$

allowing a relaxation of the height criterion when the phase error becomes excessive. That is, this correction 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. Of course proper adjustment of the parameter κ_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 the scalar values analytic switch $s \in [0, 1]$,

$$m_{II}(y_0, y_1) = s(y_0, y_1) m_0(y_0, y_1) + (1 - s(y_0, y_1)) m_1(y_1, y_0). \tag{38}$$

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

$$\sigma(y) = \sigma_p(y)\sigma_v(y), \tag{39}$$

where

$$\sigma_p(y) = \frac{1}{2} - \frac{1}{\pi}\text{ArcTan}\left[k_p(\theta_3 - \zeta)\right], \tag{40}$$

$$\sigma_v(y) = \frac{1}{2} + \frac{1}{\pi}\text{ArcTan}\left[-k_v\dot{\theta}_3\right]. \tag{41}$$

The construction of s is given by

$$s = \frac{\sigma(y_0)}{\sigma(y_0) + \sigma(y_1)}. \tag{42}$$

The motivation for this implementation follows directly from the previous work on the planar juggling system [12], and is based on the belief that $\sigma(y_i)$ encodes a measure the amount of attention required by ball i. By proper choice of the constants, k_p, k_v, and ζ, $\sigma(y_i)$ 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 them smoothly combines these two 'urgencies' causing the robot to track m_i when $\sigma(y_i)$ approaches 1.

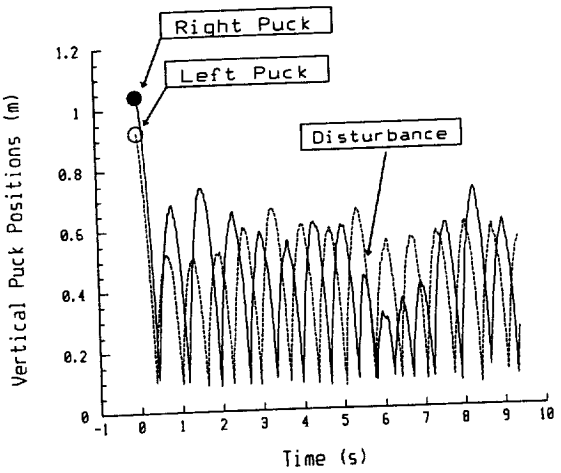


Fig. 16. Continuous vertical positions of the two-juggle.

As in the one-juggle the q_d and its temporal derivatives are directly computed from $F(y_0)$, $F(y_1)$, and m_{II} and its jacobians. Experiments with this scheme for the planar case depicted in *Fig. 5a* result in very reliable two-juggle performance with dramatic recovery from reasonably severe perturbations as shown, for example, in *Fig. 16*. Experiments using this scheme for the spatial robot depicted in *Fig. 13* are now in progress [52].

4.3. Assembly to a fixed configuration through gripping

In contrast to the immediately previous section wherein the mode of contact was batting, now consider the situation that the robot can actually grip its environment. In order to avoid the complexities of proposing a model for v_β in (10) and the significant challenge of developing a true manipulation strategy we will explore variants of the 'logical gripper' proposed in Example 2.3.1.c and depicted in *Fig. 8*. Although the simplification certainly vitiates the physical plausibility of the resulting controllers, it is curious that the same model arises in the context of the physically valid unicycle, Example 2.1.2.b. We are left with a problem equivalent to one of commanding a force

actuated 'bead robot' on a wire to move a collection of unactuated 'beads' on parallel wires into some desired final configuration from arbitrary initial conditions.

This problem is simple enough to admit a complete and provably correct solution but, upon a more than cursory examination, not nearly so trivial as might initially seem when we require the task to be planned and executed by a feedback controller. One degree of freedom assembly tasks, although uninteresting in their own right, seem to incorporate to a significant extent the same features that make the more general problems truly confounding – a combination of holonomic and nonholonomic constraints. Roughly speaking, there are two ideas at work. First, we use *navigation functions* [33] to encode the subgoals characterizing the participation of each piece in a completed assembly. This yields a motion controller for each bead according to the methods of Section 3. Since it has been demonstrated above that no single closed loop controller can result in a completed assembly, one requires some 'higher level' organizing principle to switch between the alternative closed loops. This second idea, autonomous scheduling of conflicting subgoals, is motivated in large measure by the robot juggling ideas presented immediately above. A straightforward encoding of 'urgency' is used to select the most deserving lower level alternative. The discrete time closed loop dynamics presented by the higher level switching progress can be shown to converge by appeal to standard ideas of nonlinear programming. In summary, the present approach to assembly might be characterized as 'juggling a navigation function'.

4.3.1. No obstacles

Consider first the variation on Example 2.3.1.c depicted in *Fig. 17a* where the two bodies when subject to the noholonomic restrictions imposed by the Coulomb friction (15) define the freespace with no obstacles depicted in *Fig. 17b*.

We may now pose the one degree of freedom dual assembly problem. It is desired to place body $\mathcal{B}_1$ at a desired goal location, $d_1 \in \mathcal{E}_1 = \{\theta_1 \in \mathbb{R}^2 : \| \theta_1 \|^2 = l_1\} \approx S^1$, and similarly for $\mathcal{B}_2$. The robot must start from an arbitrary location, relocate the body, and then return to a specified nest location, $n \in \mathcal{R} \approx S^1$. We require an autonomous feedback control strategy for the robot that will enable it to move toward the pucks, 'grab' them, and place them in the designated new location, and then proceed to its nest. Somewhat surprisingly, this is not as easy as it sounds. The problem is entirely trivial if open loop strategies are permitted. If, motivated by the discussion in the introduction, a closed loop strategy is desired, then the situation changes dramatically. According to the results of Section 4.1.1, now, rather than being trivial, the problem is unsolvable in the traditional sense.

In [27] this problem is solved by recourse to a simple hierarchical controller defined on a sampled version of the continuous dynamics (22). At the low level, several different robot controllers with differing goals are defined according to the techniques discussed in Section 3.1.1. A higher

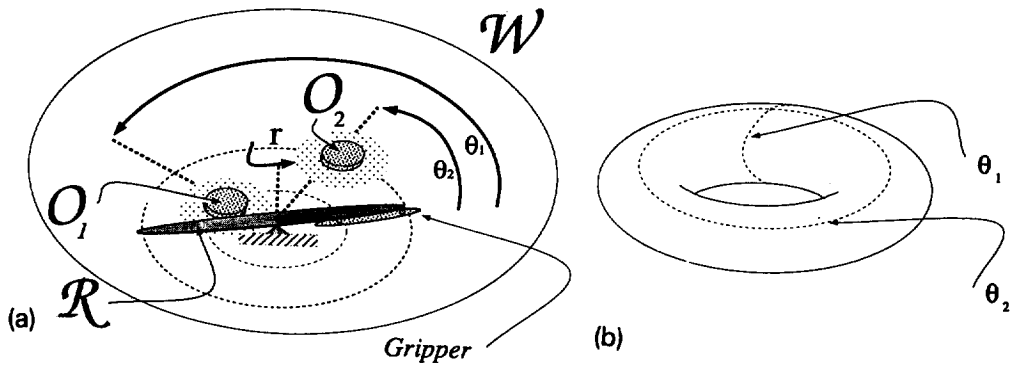


Fig. 17. (a) The unactuated objects' range of motion does not overlap. (b) The objects' configuration space is a torus.

level decision rule chooses between the competing algorithms at discrete times defined by the system entering the contact set. The resulting discrete dynamical system bears some resemblance to the discrete system (25) arising from the juggling problem of Example 2.3.2.a presented above. A more detailed rendering of the scheme is as follows.

First define a navigation function, φ, for the unactuated degrees of freedom, (θ_1, θ_2) following the technique of Section 3.2.1 applied to the desired destination (d_1, d_2). Recall from (7) β_{i3}, a function that vanishes on the portion of the contact set associated with a collision between the robot and $\mathcal{O}_i$ for the scene depicted in *Fig. 5a* when both inhabit the same workplace. This may be used in the present situation in place of a navigation function to encode the need to approach an unactuated object. Namely, the controller

$$u_1(x, y) \triangleq -\dot{r} - \gamma D_{\theta_i}\varphi(\theta_i)\mathsf{c}_i - D_r\beta_{i3}(r, \theta_i)(1 - \mathsf{c}_i)$$

brings the robot toward $\mathcal{O}_i$ when it is not proximal to that object ($\mathsf{c}_i = 0$) and brings the robot toward the desired destination for $\mathcal{O}_i$ otherwise – that is, in the neighborhood ($\mathsf{c}_i \neq 0$) of the portion of the contact set where the robot and $\mathcal{O}_i$ interact. This construction uses the ideas introduced in Section 3.1.1 and is treated carefully in [27]. The obvious gripper strategy associated with this motion controller takes the form

$$u_0(x, y) \triangleq \begin{cases} 0 & |r - \theta_i| = 0, \\ 1 & |r - \theta_i| = 1, \end{cases}$$

and causes the robot to actually grip the object when in proximity. Note that a smooth version of this commonsense strategy may be readily defined [27]. Let $g_i(x, y)$ denote this controller associated with $\mathcal{O}_i$: it drives the rotational joint of the robot toward the object, grasps the object, and then moves towards the object's designated destination, d_i.

The question now at hand concerns the construction of a higher level scheduling algorithm to mediate between the two strategies. Using the coordinate system introduced in Example 2.3.1.c, denote the evolution of the robot and its environment $(x, y)(t)$. Given an initial condition, (x_0, y_0), define a *stage* as a new contact event: an event beginning at a time s before which the trajectory was not in the contact set and (over some open interval) after which the trajectory is in the contact set [27]. A stage terminates according to a condition that measures the total mechanical energy and guarantees it has decreased since the last termination. These definitions form the basis of a series of discrete events that take place on the contact set, $\mathscr{C}$, whose control may now be effected by appropriately scheduling the low level algorithms. There are two questions to

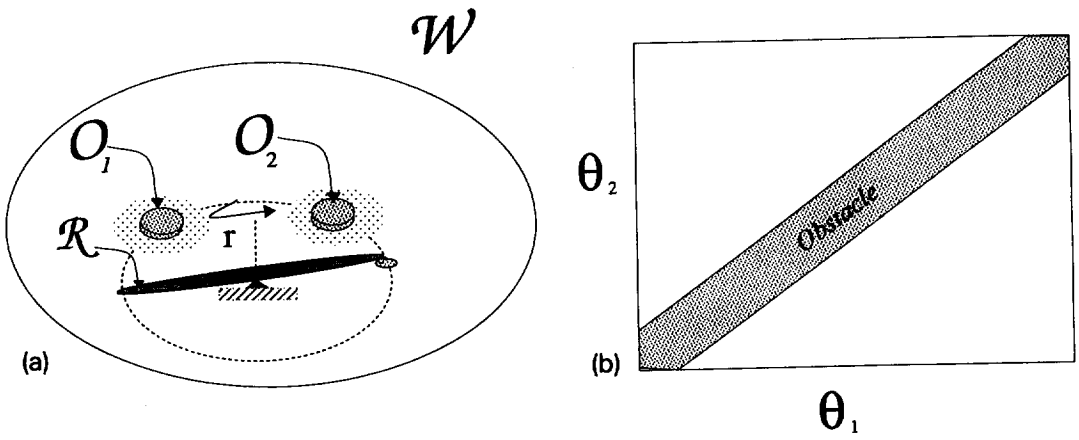


Fig. 18. (a) The unactuated objects' range of motion overlaps so unfettered circular motion is no longer possible. (b) The unactuated objects' configuration space is now a punctured torus.

address in the analysis of the 'high level' scheduler. First, does the 'contact schedule' generate an admissible sequence of low level feedback controllers? Second, can it be shown that the resulting 'contact schedule' achieves the desired final configuration, or terminates when that configuration is found to be unreachable? Both questions are answered affirmatively in [27].

4.3.2. Body obstacles

Next consider first the variation on Example 2.3.1.c depicted in *Fig. 18a* where the two bodies when subject to the noholonomic restrictions imposed by the Coulomb friction (15) define the freespace with no obstacles depicted in *Fig. 18b*.

In [27] it is shown that when a navigation function, φ for the unactuated bodes in the punctured torus of *Fig. 18b* is specified then exactly the same strategy as discussed above seems to work. Curiously, we anticipate, but have not yet proven that a similar strategy should work for a unicycle whose obstacles are defined only with respect to the unactuated degrees of freedom, for example, the scene depicted in *Fig. 19*.

The motive for attempting to apply controllers for Example 2.3.1.c to the apparently dissimilar unicycle of Example 2.1.2.b provided by the similarity in form between their dynamical equations, (21) and (22). The chief question concerns the definition of a 'contact' set since there is no obvious structure provided by the workplace. Instead, once again form a navigation function, φ for the unactuated 'body' degrees of freedom y and define the contact set with respect to it as

$$\mathscr{C}(\varphi) \triangleq \arg \min_{r,y} D_y \varphi \cdot \mathrm{c}.$$

A low level controller analogous to u_1 may be used to steer the robot – in this case, the orientation of the bar in *Fig. 5b* – toward the contact set while 'driving' with a strategy analogous to u_0 as long as the 'contact' is maintained sufficiently strongly. A system of discrete events is once again defined with respect to the robot's entering and exiting the contact set. The choice of which contact to make – in this case, the decision as to which local maximum of the inner product defining $\mathscr{C}(\varphi)$ the robot's angle should be moved to – may be also made analogously according to a switching rule that chooses the most promising magnitude. We conjecture but have not yet proven that this scheme results in convergence to the desired goal with no collisions along the way.

4.3.3. The general case: Assembly as a game played by its pieces

In the previous two versions of the assembly problem, the freespace could be factored into the cross product of a 'punctured' space associated with the unactuated degrees of freedom and an unconstrained space associated with the actuated robot degrees of freedom. The hierarchical controllers described above the take explicit advantage of this factorization by maneuvering freely with the robot in order to approach the contact set in the most favorable manner. In the general case, such a simple factorization will no hold.

More realistic settings of the assembly problem include the robot within the same workplace as the unactuated objects – for example when only one object in Example 2.1.1.b is actuated as in *Fig. 20*. In contrast to the situation depicted in

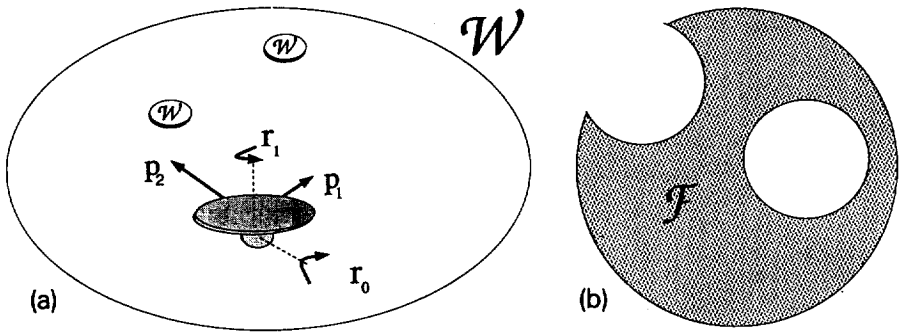


Fig. 19. (a) The unactuated degrees of freedom are obstructed but the robot's directional heading is unconstrained. (b) The unactuated objects' configuration space is the punctured plane from *Fig. 2b*.

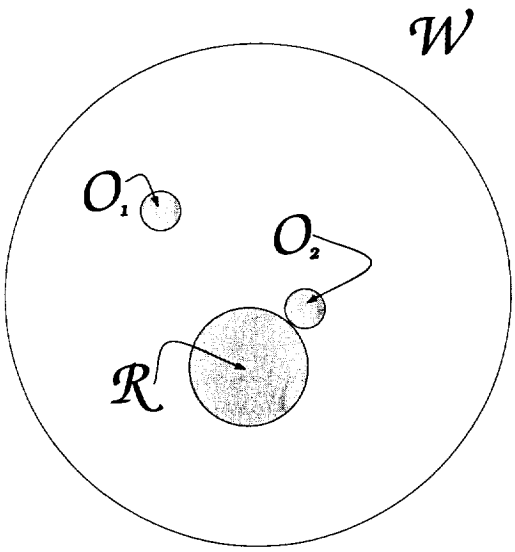


Fig. 20. The robot and the unactuated bodies usually inhabit the same workplace.

Fig. 18, a single navigation function for the desired destination in the four dimensional configuration space formed by the free placements of $\mathcal{O}_1$ and $\mathcal{O}_2$ in $\mathcal{W}$ will not be useful in guiding the robot's actions. When the robot is in contact with one of the objects, their non-spherically symmetric shape admits a very different set of free placements than those allowed by the spherically symmetric object alone. Recent work by Rimon [48] has produced a constructive navigation function for the 'spider'-like object formed by two rigidly attached disks ($\mathcal{R}$ and $\mathcal{O}_2$, in the figure) moving in a Euclidean sphere work. Evidently, two different functions are required depending upon which object the robot is 'gripping'. Thus, assembly in the general case appears to have the properties of a game played by competing objects. We have considered an elementary game theoretic setting of the one degree of freedom assembly problems depicted in *Fig. 18* [27], and are presently working on extensions to problems such as the one described here.

5. Task encoding: Shaping dynamical geometry via feedback

We have argued in Section 2 that robotic tasks may be cast as control problems with a great diversity of dynamical features. We have presented in the subsequent two sections a number of procedures for constructing controllers that solve certain of these tasks. Broadly speaking, one may distinguish between three types of controller (17),

(1) *Memoryless feedback:* u is purely a function of q with no time dependence;
(2) *Dynamical feedback:* u is the output of a time invariant dynamical system whose input is q;
(3) *Feedforward and feedback:* u depends explicitly upon q and t.

The predominant paradigm in contemporary robotics entails a special instance of the third type. In this paradigm, an a priori reference trajectory is planned, $q_r(t; q_0)$, that connects a specified initial condition with the goal. The controller, u, is then computed to force the mechanism to 'track' q_r. While its seems clear that the greater generality of the third type may offer attractive advantages, we have taken the position in this paper that the contemporary tracking paradigm may be realized in essence by a controller of the first type and that there other problems wherein it is not even clear how to generate effective reference trajectories that may be solved by a controller of the first or second type.

The emphasis on dynamical feedback places the primary burden of 'planning' on the properties of a resulting closed loop vector field. This approach offers certain theoretical and practical advantages. A vector field on a manifold, $\mathcal{E}$, is a function that assigns to each element, $e \in \mathcal{E}$, a vector, $f(e) \in T_e\mathcal{E}$, tangent to $\mathcal{E}$ at e. Intuitively, a vector field may be thought of as a rule that assigns a direction and magnitude of motion to each point in $\mathcal{E}$ – a 'law of arrows'. In this sense, vector fields represent the most general event driven means of generating motion, since they take the form 'if you are here now, then go this way'. Under mild technical assumptions vector fields generate a family of time-parameterized paths – their integral curves – each path itself parametrized by the initial condition [24]. [4] from

[4] In general, motion planning problems tend to be cast as the search for a path. Given a start point, one seeks to associate a path of prescribed properties leading to the goal set. In order to support a controller, this path must be parametrized to obtain a motion – a reference trajectory, as discussed above. Generally speaking, one wishes a rule for constructing such goal oriented motions from any legal start point. Thus, motion planning problems seem to be a natural setting within which to introduce plans that take the form of vector fields. We have tried to suggest that more general problems involving dynamical interaction may be amenable to treatment in this fashion as well.

the practical point of view, event driven systems of type 1 or 2 are very attractive since actions are taken without reference to past history of the correct time. The benefits of an ahistorical approach to planning are underscored when considering, for example, the legendary inaccuracy of odometry whose errors accumulate over the course of motion. The benefits of time invariant procedures become particularly important in distributed concurrent controller implementations where a global clock may not be available and where an asynchronous data flow architecture may fit the available hardware most naturally [55].

It might be argued that these advantages are balanced by a new disadvantage. One requires exact knowledge of where the system is at present. Otherwise, the 'arrow' belonging to e will be applied mistakenly when passing through a neighboring point, e'. In fact, when the vector field is smooth and the vectors assigned to nearby points are close to each other, small uncertainties in present state may be treated as small perturbations of the correct vector field. For certain classes of vector fields – those that are structurally stable [42] – it can be guaranteed that sufficiently small perturbations do not affect the qualitative properties of the resulting integral curves. There are various levels of qualitative properties which, as they become more prescriptive of the global properties of the system, yield criteria for structural stability that are harder and harder to check in practice. As far as local behavior is concerned, structural stability is a generic property [42]. Moreover, there are restricted classes – for example the gradient systems [42] – for which global structural stability is generic as well. Thus, placing one's fate in the hands of a 'nicely' shaped dynamical system may be a very reasonable action in the face of uncertainty.

A few theoretical results and laboratory experiments reported in this paper suggest the possibility of a more unified approach to robot command and control than has been seen heretofore in the field. Clearly, more experimental and theoretical work will be required before the validity of the task encoding framework advanced here can be established of refuted.

Acknowledgements

It is a pleasure to thank my students, Martin Bühler, Elon Rimon, Al Rizzi, and Louis Whitcomb who contributed many of the ideas (and, of course, many of the figures) discussed here. I would like to particularly thank Leo Dorst who has urged the creation of this paper, and patiently helped to advise and encourage its writer.

References

[1] R. Abraham and J.E. Marsden, *Foundations of Mechanics* (Benjamin/Cummings, Reading, MA, 1978).

[2] R.L. Andersson, Aggressive trajectory generator for a robot ping-pong player, in: *Proceedings 1988 International Conference on Robotics and Automation*, Philadelphia, PA, (1988) 188–193.

[3] J.R. Andrews and N. Hogan, Impedance control as a framework for implementing obstacle avoidance in a manipulator, in: D.E. Hardt and W.J. Book, eds., *Control of Manufacturing and Robotics Systems*, (A.S.M.E., Boston, MA, 1983) 243–251.

[4] V.I. Arnold, *Mathematical Methods of Classical Mechanics* (Springer-Verlag, New York, 1978).

[5] A.M. Bloch and N.H. McClamroch, Control of mechanical systems with classical nonholonomic constraints, in: *Proc. 28th IEEE Conf. on Decision and Control*, Tampa, FL (1989) 201–205.

[6] W.H. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry* (Academic Press, New York, 1975).

[7] R.W. Brockett, On the computer control of movement, in: *Proc. IEEE Conference on Robotics and Automation* (IEEE, Philadelphia, PA, 1988) 534–540.

[8] R.W. Brockett, Asymptotic stability and feedback stabilization, in: R.W. Brockett, R.S. Millman and H.J. Sussman, eds., *Differential Geometric Control Theory*, chapter 3 (Birkhäuser, 1983) 181–191.

[9] R.C. Brost, Computing metric and topological properties of configuration space obstacles, in: *Proc. IEEE Conference on Robotics and Automation* (IEEE, Scottsdale, AZ, 1989) 170–176.

[10] M. Bühler, D.E. Koditschek and P.J. Kindlmann, A simple juggling robot: Theory and experimentation, in: V. Hayward and O. Khatib, eds., *Experimental Robotics I* (Springer-Verlag, New York, 1990) 35–73.

[11] M. Bühler and D.E. Koditschek, From stable to chaotic juggling, in: *Proc. IEEE International Conference on Robotics and Automation*, Cincinnati, OH (1990) 1976–1981.

[12] M. Bühler, D.E. Koditschek and P.J. Kindlmann, Planning and control of a juggling robot, *Int. J. Rob. Research* (1991).

[13] M. Bühler, D.E. Koditschek and P.J. Kindlmann, A family of robot control strategies for intermittent dynamical environments, *IEEE Control Systems Magazine* 10 (1990) 16–22.
[14] M. Bühler, Robotic tasks with intermittent dynamics, Ph.D. Thesis, Yale University, New Haven, CT (1990).
[15] C.C. Chang and H.J. Keisler, *Model Theory* (North-Holland, Amsterdam, 1973).
[16] J.J. Craig, *Introduction to Robotics Mechanics and Control* (Addison Wesley, Reading, MA, 1986).
[17] M.L. Curtis, *Matrix Groups* (Springer-Verlag, New York, 1979).
[18] R.L. Devaney, *Introduction to Chaotic Dynamical Systems* (Addison Wesley, Reading, MA, 1987).
[19] B.R. Donald, Error detection and recovery for robot motion planning with uncertainty, Ph.D. Thesis, MIT (1987).
[20] M.A. Erdmann, On motion planning with uncertainty, Technical Report AI-TR-810 (MS Thesis), MIT AI Lab (1984).
[21] H. Goldstein, *Classical Mechanics* (Addison-Wesley, Reading, MA, 1980).
[22] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, New York, 1983).
[23] J. Guckenheimer, Sensitive dependence to initial conditions for one dimensional maps, *Communications in Mathematical Physics* 70 (1979) 133–160.
[24] M.W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra* (Academic Press, Orlando, FI 1974).
[25] O. Khatib and J.-F. Le Maitre, Dynamic control of manipulators operating in a complex environment, in: *Proc. Third International CISM-IFToMM Symposium*, Udine, Italy, (1978) 267–282.
[26] C.W. Kilmister and J.E. Reeve, *Rational Mechanics* (Longmans, London, 1966).
[27] D.E. Koditschek, An approach to autonomous robot assembly, *Robotica* (1992) (to appear).
[28] D.E. Koditschek and M. Bühler, Analysis of a simplified hopping robot, *Int. J. Rob. Research* 10 (6) (1991).
[29] D.E. Koditschek, Exact robot navigation by means of potential functions: Some topological considerations, in: *IEEE International Conference on Robotics and Automation*, Raleigh, NC (1987) 1–6.
[30] D.E. Koditschek, Robot control systems, in: S. Shapiro, ed., *Encyclopedia of Artificial Intelligence* (Wiley, New York, 1987) 902–923.
[31] D.E. Koditschek, The application of total energy as a Lyapunov function for mechanical control systems, in: J. Marsden, Krishnaprasad and J. Simo, eds., *Control Theory and Multibody Systems*, vol. 97, AMS Series in Contemporary Mathematics (1989) 131–158.
[32] D.E. Koditschek, The control of natural motion in mechanical systems, *ASME Journal of Dynamics Systems and Measurement* (Dec. 1991).
[33] D.E. Koditschek and E. Rimon, Robot navigation functions on manifolds with boundary, *Advances in Applied Mathematics*, (1990) 412–442.
[34] J.-C. Latombe, *Robot Motion Planning* (Kluwer, Boston, MA, 1991).
[35] T. Lozano-Perez, M.T. Mason and R.H. Taylor, Automatic synthesis of finemotion strategies for robotics, *The International Journal of Robotics Research* 3 (1) (1984) 3–23.
[36] T. Lozano-Perez and M.A. Wesley, An algorithm for planning collision free paths among polyhedral obstacles, *Common. ACM* 22 (10) (1979).
[37] M.T. Mason, Mechanics and planning of manipulator pushing operations, *International Journal of Robotics Research* 5 (3) (1986) 53–71.
[38] M.T. Mason, Compliance and force control for computer controlled manipulators, *IEEE Trans. Syst. Man and Cybernetics* SMC-11 (6) (1981) 418–432.
[39] T. McGeer, Passive dynamic walking, *The International Journal of Robotics Research* 9 (2) (1990) 62–82.
[40] F. Miyazaki and S. Arimoto, Sensory feedback based on the artificial potential for robots, in: *Proceedings 9th IFAC*, Budapest, Hungary (1984).
[41] J.I. Neimark and N.A. Fufaev, *Dynamics of Nonholonomic Systems, Vol. 33: Translations of Mathematical Monographs* (American Mathematical Society, Providence, RI, 1972).
[42] J. Palis, Jr. and W. de Melo, *Geometric Theory of Dynamical Systems* (Springer-Verlag, New York, 1982).
[43] V.V. Pavlov and A.N. Voronin, The method of potential functions for coding constraints of the external space in an intelligent mobile robot, *Soviet Automatic Control* 6 (1984).
[44] M.H. Raibert, *Legged Robots That Balance* (MIT Press, Cambridge, MA, 1986).
[45] E. Rimon and D.E. Koditschek, Exact robot navigation using cost functions: The case of spherical boundaries in E^n, in: *International Conference on Robotics and Automation*, Philadelphia, PA (1988) 1791–1796.
[46] E. Rimon and D.E. Koditschek, Exact robot navigation in topologically simple but geometrically complicated environments, in: *Proc. IEEE International Conference on Robotics and Automation*, Cincinnati, OH (1990) 1937–1943.
[47] E. Rimon and D.E. Koditschek, The construction of analytic diffeomorphisms for exact robot navigation on star worlds, *Transactions of the American Mathematical Society* 327 (1) (1991) 71–115.
[48] E. Rimon, A navigation function for a simple rigid body, in: *Proc. IEEE Int. Conf. Rob. and Aut.*, Sacramento, CA (IEEE Computer Society, 1991) 2–7.
[49] E. Rimon and D.E. Koditschek, Exact robot navigation using artificial potential fields, *IEEE Trans. Robotics and Automation* (1992) (to appear).
[50] E.F. Rimon, Exact robot navigation using artificial potential functions, Ph.D. Thesis, Yale University, New Haven, CT (1990).
[51] A.A. Rizzi and D.E. Koditschek, Progress in spatial robot juggling, in: *IEEE Int. Conf. Robt. Aut.*, Nice, France (1992).
[52] A.A. Rizzi, L.L. Whitcomb and D.E. Koditschek, Distributed real-time control of a spatial robot juggler, *IEEE Computer* 25 (5) (1992).
[53] D. Singer, Stable orbits and bifurcations of maps of the interval, *SIAM J. Applied Mathematics* 35 (2) (1978) 260–267.

[54] Sir W. Thompson and P.G. Tait, *Treatise on Natural Philosophy* (University of Cambridge Press, Cambridge, 1886).

[55] L.L. Whitcomb and D.E. Koditschek, Robot control in a message passing environment, in: *Proc. IEEE International Conference on Robotics and Automation*, Cincinnati, OH (1990) 1198–1203.

[56] L.L. Whitcomb and D.E. Koditschek, Toward the automatic control of robot assembly tasks via potential functions: The case of 2-d sphere assemblies, in: *IEEE Int. Conf. Rob. Aut.*, Nice, France (1992).

[57] L.L. Whitcomb and D.E. Koditschek, Automatic assembly planning and control via potential functions, in: *Proc. IEEE International Workshop on Intelligent Robots and Systems*, Osaka, Japan (IEEE, 1991).

[58] D.E. Whitney, Force feedback control of manipulator fine motions, *ASME J. Dyn. Syst.* (1977) 91–97.