




1986

# Automatic Planning and Control of Robot Natural Motion Via Feedback

Daniel E. Koditschek

University of Pennsylvania, kod@seas.upenn.edu

Follow this and additional works at: [http://repository.upenn.edu/ease\\_papers](http://repository.upenn.edu/ease_papers)

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

---

## Recommended Citation

Daniel E. Koditschek, "Automatic Planning and Control of Robot Natural Motion Via Feedback", *Adaptive and Learning Systems: Theory and Applications*, 389-402. January 1986.

Preprint version. Published in *Adaptive and Learning Systems: Theory and Applications*, 1986, pages 389-402. The original publication is available at [http://link.springer.com/chapter/10.1007/978-1-4757-1895-9\\_28](http://link.springer.com/chapter/10.1007/978-1-4757-1895-9_28).

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

This paper is posted at ScholarlyCommons. [http://repository.upenn.edu/ease\\_papers/680](http://repository.upenn.edu/ease_papers/680)

For more information, please contact [repository@pobox.upenn.edu](mailto:repository@pobox.upenn.edu).

---

# Automatic Planning and Control of Robot Natural Motion Via Feedback

## **Abstract**

A feedback control strategy for the command of robot motion includes some limited automatic planning capabilities. These may be seen as sequential solution algorithms implemented by the robot arm interpreted as a mechanical analog computer. This perspective lends additional insight into the manner in which such control techniques may fail, and motivates a fresh look at requisite sensory capabilities.

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

## **Disciplines**

Electrical and Computer Engineering | Engineering | Systems Engineering

## **Comments**

Preprint version. Published in *Adaptive and Learning Systems: Theory and Applications*, 1986, pages 389-402. The original publication is available at [http://link.springer.com/chapter/10.1007/978-1-4757-1895-9\\_28](http://link.springer.com/chapter/10.1007/978-1-4757-1895-9_28).

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

# Automatic Planning and Control of Robot Natural Motion Via Feedback

Preprint from  
*Adaptive and Learning Systems: Theory and Applications*  
K.S. Narendra (ed.), Plenum, 1985

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

September 5, 1985

## Abstract

A feedback control strategy for the command of robot motion includes some limited automatic planning capabilities. These may be seen as sequential solution algorithms implemented by the robot arm interpreted as a mechanical analog computer. This perspective lends additional insight into the manner in which such control techniques may fail, and motivates a fresh look at requisite sensory capabilities.

---

\*This work is supported in part by the National Science Foundation under grant no. DMC-8505160

# 1 Introduction

Much of classical control theory is concerned with the design of controllers possessing a “feed-forward” structure which filters time-varying reference signals, forcing a closed loop plant compensated by the appropriate feedback structure. The task at hand might be said to be “encoded” by the reference signal (or class of signals) applied; the compensator is designed to force the plant to accomplish the task. While theoretically sound rules of thumb for the design of such controllers are available in the context of linear time invariant plants, there is, at present, no analytically tractable procedure for determining the response of the nonlinear dynamical systems defined by typical industrial robots to time varying inputs. In consequence, classical techniques cannot be applied to their control with confidence.

The last five years have seen a variety of nonlinear control methodologies proposed for the command of robot arm motion and force. It would be fair to say that in most cases, the conceptual basis for task encoding has remained fixed upon the paradigm of a reference trajectory, presumably the output of some arbitrary higher level planning algorithm, which the controller forces the plant to track. However, in the absence of theoretically informed choices for appropriate reference signals, even the best high level planning algorithm may produce poor command inputs: encoding a task in dynamical terms entirely unrelated to those of the plant may not be viable. Moreover, there is mounting evidence that the computational effort required to encode typical tasks (such as moving in a cluttered space) in terms of exact reference trajectories may be prohibitive [22], [25].

This paper will explore aspects of a rather different control methodology which replaces the role of such feedforward reference signals in favor of task characterization in terms of a feedback structure. “Natural Motion” refers to the resulting unforced response of the autonomous closed loop system. Specifically, we focus upon the mechanism by which the system generates a “plan of action” automatically, obviating the need for an independent reference signal. Earlier results of the author, [12], are re-interpreted to demonstrate their foundation upon two simple and very old ideas: that a set of algebraic equalities may be solved sequentially on an analog computer; and that a kinematic chain, in the absence of external torques, is a passive dynamical system. Their re-interpretation leads to a better understanding of how the algorithms may fail, and motivates a discussion concerning the trade-offs between sensing and computation. This understanding seems to be a first step along the way to a methodology of robot task encoding via feedback structures. In the long run, of course, robot controllers probably ought

to be built with both feedforward pre-compensators and feedback. However, the capabilities of rather simple error-driven systems may afford the command of a surprisingly sophisticated range of robot tasks, hence, deserve a much better hearing in the robotics community than has yet been accorded.

The question of how particular robot tasks should be encoded, while not explicitly discussed in this paper, might be seen as a unifying theme for all the papers presented in this session of the Fourth Yale Workshop. The pure feedback based control structures discussed here represent a particularly simple implementation of tasks which have been encoded using potential functions - e.g. Hogan's impedance control methodology [8], or Khatib's obstacle avoidance methodology [11]. The extent to which other robot task paradigms can be implemented by pure feedback controllers is not at all clear, but seems a question well worth pursuing. Its pursuit is bound to shed more light on the general problem of how to translate abstract task goals into computationally efficient and dynamically sound controllers.

## 1.1 A Robot Model

The equations of motion of a kinematic chain have been extensively discussed in the robotics literature, and this paper will rely upon the standard rigid body model of an open chain with revolute joints. Thus, we consider a robot to be the dynamical system

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + k(q) = \tau, \quad (1)$$

where  $B$  is linear in  $\dot{q}$ , and  $M, B, k$  all vary in  $q$  by polynomials of transcendental functions. This system is equivalent to

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1}(x_1)[B(x_1, x_2)x_2 + k(x_1) - \tau] \end{aligned} \quad (2)$$

where the generalized positions and velocities take values  $x \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{P} \triangleq TJ$ , *phase space*.

In this ideal world, the robot is a finite dimensional dynamical plant which accepts torque inputs,  $\tau$ , and delivers joint positions and velocities,  $x_1, x_2$ . Missing from this model, evidently, is any account of actuator dynamics and nonlinearities (including absolute power constraints) as well as viscous and nonlinear friction at the joints. The omission of the former set of

phenomena from the model is problematic, but reflects the currently volatile state of experimentation in the robotics world with a variety of actuating technologies which promise to be “better” than the standard dc-servo with gear-train [4], [10]. The omission of the latter, seems to be more generally defensible on the grounds that, unlike the inverse-dynamics approach to arm control [18], [7], the reliance on the natural motion of an arm - a pure feedback strategy - requires no explicit knowledge of the underlying velocity coefficients. Although dramatic changes in damping over the course of motion are bound to degrade performance, it will be seen below that the feedback schemes being proposed are guaranteed to maintain *stability* at all times (assuming that any “negative” friction is entirely due to the coriolis term,  $B$ ). As with any feedback scheme, however, the capabilities and deployment of sensors will be critical to the reliability of this analysis, and we will return to such questions in the last section.

The space of generalized positions, termed *joint-space*, ( $\mathcal{J}$  in the sequel), is the cross product of a  $k$  dimensional torus with  $\mathbf{R}^{n-k}$  where  $n$  is the total number of joints, and  $k$  is the number of revolute joints. There are two circumstances under which  $\mathcal{J}$  may be accurately modelled as a simply connected compact subset of  $\mathbf{R}^n$ : if each revolute joint is mechanically constrained to prevent a full 360 degree revolution; or, if full revolution is mechanically possible, but joint sensors are available which transduce angular displacement with respect to some absolute position - e.g., from which a revolution and a half displacement is read as 540 degrees rather than 180 degrees. It will be seen in the sequel that such apparently picky details may have significant consequences when pure feedback control is applied.

Of course, equation (??) is an incomplete model of a robot arm in the absence of an explicit output map from  $\mathcal{J}$  to what we will term *work space*, or the set of *frames*. These, of course, represent the position and orientation of some orthonormal coordinate system fixed in the (rigid) end-effector with respect to an inertial system in the base of the arm. Appendix B introduces a notation for and develops some standard results from the algebra and calculus of frames which are modeled as points in  $\mathcal{W} \triangleq \mathbf{R}^3 \times SO(3)$ .

## 1.2 Inverse Dynamics

Before proceeding, it is worth pointing out that a conceptually very straight forward procedure exists which, if practicable, would bring robot control problems back into a domain entirely amenable to the well understood classical methodology. Robots are built with an actuator for each degree of freedom (assuming rigid links), hence, constitute a dynamical system which

may be completely linearized and decoupled by state feedback. This possibility would suggest a return to the standard means of task encoding via reference trajectories.

Specifically, given a reference trajectory,  $q_r$ , and its derivatives,  $\dot{q}_r, \ddot{q}_r$  the control strategy,

$$\begin{aligned}\tau &\triangleq [B - MK_2]\dot{q} + [k(q) - MK_1q] + \tau_r \\ \tau_r &\triangleq M[\ddot{q}_r + K_2\dot{q}_r + K_1q_r],\end{aligned}\tag{3}$$

$K_1, K_2 > 0$ , may be shown *theoretically* to drive the errors,  $e \triangleq \begin{bmatrix} q_r \\ \dot{q}_r \end{bmatrix} - x$  to zero, exponentially in time for arbitrary initial values. An analogous procedure, termed “resolved acceleration control” [18] can be used when the reference signals are expressed in work space coordinates avoiding explicit computation of the inverse kinematics.

This technique has been proposed by a variety of researchers in a diversity of guises [7], [18], [27] and is known most widely as the method of “computed torque” or “inverse dynamics”. Since it “linearizes” the equations of motion by exact cancellation of thousands of nonlinear terms from the rigid body model of robot dynamics, the question arises as to whether the method may be applied in practice at all: can the computation be effected quickly enough; can it be effected accurately enough? A number of researchers have persuasively argued that the answer to the first question is (or may soon be) yes [9], [15], which conclusion we will accept with no further discussion here. In the absence of reported empirical experience or analytical study, however, the second question remains open. Caution seems justified in light of (i) the inaccuracies in the rigid body model upon which such computation would be based; (ii) uncertainty regarding the values of the link dynamical parameters, and (in general) total ignorance of the load dynamical parameters; and (iii) numerical inaccuracies in computation resulting from quantization effects.

## 2 Natural Motion of Robot Arms

Unlike other control methodologies that have emerged in the context of robotics, the feedback based algorithms described here admit a mathematical proof of global convergence with minimal a priori knowledge of the actual plant dynamics. The underlying mechanism of stability was understood as long as a century ago [26], but only introduced to the robotics community in 1981 by Arimoto [1] ( and, independently, by this author in 1984 [12]). The idea of solving algebraic systems of equations by sequential techniques is much older. A direct consequence of the stability properties above is that a particular sequential technique may be modified for implementation on the second order nonlinear mechanical analog computer which we will interpret the robot arm to be. This interpretation of the author's earlier results [13] is partly suggested by a recent paper of Wolovich and Elliott [28].

### 2.1 Sequential Solution of a Set of Equations

Perhaps the most widely used general method for solving a set of nonlinear equations

$$0 = f(x)$$

is Newton-Raphson iteration

$$x_{n+1} = x_n - df_{x_n}^{-1} f(x) \quad (4)$$

which is known to converge to the *solution set*,  $\mathcal{S} \triangleq f^{-1}[0]$ , as long as  $x_0$  is "close" to that set, and  $|df|$  is bounded away from zero on the subsequent trajectory. Recent work of Smale [23] and Hirsch and Smale [24] has established variants of this technique which succeed for almost all initial points,  $x_0$ , arbitrarily far from  $\mathcal{S}$ , given mild assumptions on  $f$ . They point out that equation (4) is a discrete time version of the differential equation

$$\dot{x} = -df^{-1} f(x), \quad (5)$$

whose solutions are confined to curves in the inverse image  $f^{-1}[\mathcal{L}]$  of the ray

$$\mathcal{L} \triangleq \{tf(x_0) \mid t > 0\}$$

connecting the origin of the target space to the image of the first guess,  $f(x_0)$ . They also mention a much older variant of this technique, the algorithm

$$\dot{x} = -df^T f \quad (6)$$



which we will call the *gradient* method, since its trajectories follow the gradient of the scalar “error function”,

$$\varepsilon \triangleq \|f\|,$$

in the target space. This system is also guaranteed to converge to  $\mathcal{S}$  for all initial conditions which are sufficiently close to the solution set. Unfortunately, unlike algorithm (5) it cannot, in general, be modified to converge to  $\mathcal{S}$  globally, even disregarding a set of measure zero, for reasons which will become apparent.

It should be mentioned that the gradient method has a long history in systems science as well as the general world of applied mathematics. For instance, the parameter adjustment algorithms of adaptive control and estimation schemes may be seen as a gradient solution to a set of linear algebraic equations [17]. In the field of robotics, Wolovich and Elliott [28] have recently proposed this algorithm as an off-line numerical solution technique to the same algebraic equations as we consider here - namely, the kinematic transformation of a robot arm. This paper, however, concerns an on-line implementation of the gradient algorithm implicit in the motion of the robot arm itself. In effect, it is proposed to perform the integration called for in equation (6) upon a mechanical analog computer formed by the robot arm itself, whose integrators obtain from its intrinsic dynamics. The arm solves its own inverse kinematics by moving to the desired target.

A number of limitations inhere from the beginning. In particular:

1. the technique is intrinsically time invariant - convergence is only guaranteed if  $\frac{df}{dt} = 0$ ;
2. although the trajectories of system (5) can be shown to move toward their goal along straight lines in the target space, the gradient lines of the error magnitude,  $\varepsilon$ , depend entirely on properties of  $f$  itself, and, in general, will be curved in both the domain and target space;
3. while  $\mathcal{S}$  is the positive limit set of almost every initial condition of system (5), (assuming a suitable adjustment in the sign of the vector field, as described in [24] ) the same cannot be said of system (6);
4. while equation (6) calls for a set of first order integrators, robot arm analog computers, like every other mechanical system, obey Newton’s laws, and come equipped only with a set of double integrators.

These limitations have some obvious and some less obvious implications for the performance of robot motion resulting from the feedback controls based upon the gradient algorithm.

The first caveat does not imply that a run-of-the mill tracking problem cannot be implemented in this fashion - simply that stability cannot be guaranteed. However, in the light of engineers' varied experiences with parametric excitation of forced second order systems ( cf. the fourth caveat) caution is well advised. A determined application of this methodology will do better to rely as much as possible upon encoding robot tasks in terms of static geometric constructs rather than time varying reference signals. Recent work by this author [13], Hogan [8], and Khatib [11] begins to suggest that the set of tasks amenable to such construction is rich indeed.

The second caveat would be entirely irrelevant were we proposing simply another numerical procedure for the solution of inverse kinematics. In this context, solution trajectories describe the physical motion of a robot arm and we will be critically concerned with their position in the target space. The departure of the trajectories of system (6) from straight lines in the target space is determined locally by the eigenvectors and eigenvalues of  $df^T df$  near a limit point in  $\mathcal{S}$ , and, more generally, by the gradient lines of  $\varepsilon$ . These may be "shaped", then, by the construction of suitable error functions,  $\varepsilon$ , [13], [8].

The third caveat has more subtle origins and interpretation. System (6) has equilibrium states at the critical points of the error function,  $\varepsilon$ ,

$$\mathcal{C} \triangleq \{x \mid f(x) \in \ker df^T\}$$

and, in general, that set includes more than the desired  $f^{-1}[0]$ . Namely, there exist points  $\mathcal{U} \triangleq \mathcal{C} - \mathcal{S} \neq \emptyset$  if  $f$  has critical values other than 0. The hessian (second derivatives) matrix of  $\varepsilon$  evaluated at a point in  $\mathcal{C}$  specifies the linearized vector field of (6) around that equilibrium state. Hence, any non-zero valued local minima of  $\|f\|$  define equilibrium states which are locally attractive and possess some open neighborhood within which initial conditions will converge to a useless value. This explains why the set of poor initial guesses does not, in general, have measure zero. If the error function,  $\varepsilon$ , has no local minima outside of  $\mathcal{S}$  then  $\mathcal{U}$  consists of unstable equilibrium states of system (6), and all initial conditions which do not lie on their stable manifold (necessarily a set of measure zero) converge to the solution - i.e. a point in  $\mathcal{S}$ .

Exploration of the final caveat is, of course, a main theme of this paper, and is now addressed explicitly.

## 2.2 Dissipative Mechanical Systems: Implementing a Sequential Algorithm with a Double Integrator

In light of the preceding discussion outlining the weaknesses of algorithm (6) relative to (5), it seems appropriate to inquire as to why we force the robot arm to integrate the former. For a broad range of mechanical systems, the Hamiltonian is an exact expression for total energy. In a conservative force field this scalar function is a constant (defines a first integral of the equations of motion) and, in the presence of the proper dissipative terms, it must decay [13]. If we now regard  $\varepsilon$  as a candidate potential function for the robot, and find a suitable dissipative function to match, then there is reason to hope for convergence results analogous to those of the previous section. Succinctly, then, the vector field of system (6) always admits of a potential function, while that of system (5), in general, will not.

The dissipation of total energy, will not be of much practical use unless it has the properties of a Lyapunov Function. If we assume that kinetic energy is always quadratic in velocity and never zero in position, then any positive definite potential function will work. In the previous section,  $\varepsilon$  was a norm, and hence, certainly positive definite.

For example, suppose we are commanded to move the robot arm to a zero velocity point in the state space,  $(q_d, 0)$ . From the point of view of the previous section, we would like the robot to solve the trivial linear equation in  $\mathcal{J}$ ,

$$f(q) \triangleq q - q_d = 0.$$

To implement a version of algorithm (6), we first define a positive definite error function on the generalized coordinates,

$$\varepsilon \triangleq [q - q_d]^T K_1 [q - q_d]$$

( $K_1$  is positive definite, and might specify the impedance characteristics of the manipulator [8]) and then form a scalar multiple of its gradient

$$d\varepsilon^T = K_1 [q - q_d].$$

Defining a state feedback control law for (??) based upon removing the destabilizing gravitational field (which admits a “useless” indefinite potential function) and matching the desired gradient with a positive definite dissipative term,

$$\tau = k(q) - K_2 \dot{q} - df^T K_1 f. \quad (7)$$

( $K_2$  is positive definite), yields the following result, stated here without proof.

**Theorem 1** (Takegaki and Arimoto [1], Koditschek [12]) *Let  $J$  be a simply connected compact subset of  $\mathbf{R}^n$ . The closed loop system of equation (??), under the state feedback algorithm (7),*

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -M^{-1}[(B + K_2)x_2 + K_1(x_1 - q_d)]\end{aligned}$$

*is globally asymptotically stable with respect to the state  $(q_d, 0)$  for any positive definite symmetric matrices,  $K_1, K_2$ .*

Since  $\varepsilon$  has a unique critical value, convergence to  $(q_d, 0)$  is guaranteed from all initial conditions, including arbitrary initial velocities. Note that the first condition of the hypothesis is required to avoid situations where topological properties of the manifold on which the dynamics is defined preclude the possibility of a smooth vector field with a unique equilibrium state.

There are two serious criticisms to be made of any robot controller based upon this result. First, the control law requires the exact cancellation of any gravitational disturbance. While  $k(q)$  has a much simpler structure than the moment of inertia matrix,  $M(q)$ , or the coriolis matrix,  $B(q, \dot{q})$ , exact knowledge of the plant and load dynamical parameters would still be required, in general, to permit its computation. Since the dynamical parameters enter linearly in  $k$ , some progress has been made in the design of “adaptive gravity cancellation” algorithms [14] removing the need for any *a priori* information concerning the dynamical parameters. Second, this result affords very little understanding of the trajectory that the arm will take as it moves toward  $(q_d, 0)$ . While the singly integrated gradient system

$$\dot{q} = -K_1[q - q_d]$$

converges toward  $q_d$  along the dominant eigenvector of  $K_1$ , even a linear double integrator may drive the projection onto  $J$  of its state trajectory,  $(x_1, x_2)$  quite differently. Research addressing this question is also in progress.

As the title of this paper suggests, the real utility of this approach is not simply in commanding end-point control at the joint level. Any task encoded by means of a gradient in  $\mathcal{W}$  may be commanded by simply composing its potential function with the kinematic map, and applying the same technique. The simplest example is provided by a point-to-point task. Suppose we are commanded to move the robot arm to the position  $r_d \in \mathbf{R}^3$ . The task may be encoded in terms of finding the solution to the set of nonlinear equations

$$f(q) \triangleq r_g(q) - r_d = 0,$$

where  $r_g(q)$  is the position component of the kinematic transformation from  $\mathcal{J}$  to  $\mathcal{W}$  ( see Appendix B). Again, we form a positive definite error function in the target space,

$$\varepsilon \triangleq f^T K_1 f,$$

in order to use a scalar multiple of its gradient,

$$d\varepsilon^T = dr_g^T K_1 f,$$

as the input to the robot dynamics. Proceeding as before, we define a time invariant feedback control law

$$\tau \triangleq k(q) - K_2 \dot{q} - dr_g^T K_1 (r_g(q) - r_d), \quad (8)$$

and state the following result, also without proof.

**Theorem 2** (*Koditschek [13]*)

*The closed loop system of equation (??), under the state feedback algorithm (8),*

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -M^{-1}[(B + K_2)x_2 + [d_q r_g]^T K_1 (r_g - r_d)], \end{aligned}$$

*has an attracting set contained in*

$$\tilde{\mathcal{C}} = \{(q, 0) \in \mathcal{P} : K_1 (r_g - r_d) \in \ker dr_g^T\}.$$

*for any positive definite symmetric matrices,  $K_1, K_2$ .*

This result shows that the price paid for automatic inverse kinematics is the possible addition of undesirable equilibrium states, which inhabit the zero velocity hyperplane of  $\mathcal{P}$  within the set of kinematic singularities,

$$\tilde{\mathcal{C}} \subset \{x \in \mathcal{P} : x_2 = 0, \text{rank}(dr_g) < n\}.$$

For all presently available commercial robots, kinematic singularities may be found in the interior of the workspace, thus the problem is of practical concern. For a lucky choice of  $K_1$  and  $r_d$  it might well turn out that  $\tilde{\mathcal{C}}$  consists only of the points in the solution set,

$$\tilde{\mathcal{S}} \triangleq \{x \in \mathcal{P} : x_2 = 0, r_g(x_1) = r_d\},$$

but, in general, this cannot be expected. Given the existence of additional undersirable equilibrium states outside of the solution set,

$$\tilde{\mathcal{U}} \triangleq \tilde{\mathcal{C}} - \tilde{\mathcal{S}},$$

$\tilde{\mathcal{S}}$  is no longer a global attractor. However, there is some hope that the presence of *stall* points, attractive points of  $\tilde{\mathcal{U}}$ , can be ruled out. Research is now under way to determine a pragmatic approach to this problem. Given the difficulty of merely characterizing the kinematic singularities, it may turn out that the best analysis is an estimate of the volume of the stall basins in the target space, inspired, e.g. by the measure theoretic arguments used in [23]. Simulations may provide a reasonable initial feel for this volume, considered as a function of the task and the damping constants,  $K_2$ .

The failure to construct a globally asymptotically stable solution set,  $\mathcal{S}$ , for position control is seen to be a consequence of kinematic critical points in conjunction with the second order dynamics of the arm. When we pose the complete problem involving position and orientation simultaneously we may run into a still larger set of non-solution critical values which are a consequence of the intrinsic topology of  $\mathcal{W}$ . It is well known that more sophisticated tasks encoded as gradients in task space - e.g. the obstacle avoidance potentials of Khatib - almost inevitably give rise stall points in  $\mathcal{W}$ . This is probably an intrinsic limitation in the “intelligence” of feedback controllers, and must be taken into consideration at a higher level. A more complete characterization of “stall” will be required for this to be possible.

### 3 Feedback Measurements for Natural Control

It is hoped that the ideas presented here will afford the command of initially simple robot tasks - e.g. picking and placing objects of unknown mass, possibly with impedance control, possibly along a specified curve - at much higher speeds and less planning than currently possible. We are setting out in the Robotics Lab at Yale to build or modify an existing arm which will afford a hardware implementation of the feedback control schemes discussed in this paper. In so doing, we have come up against some fundamental questions which, although invisible from the point of view of the formal analytical techniques presented thus far, go to the heart of what a robot is or should be. Having proposed a control methodology which relies heavily upon feedback, we must create a sufficiently rich sensory environment which delivers enough information sufficiently quickly and sufficiently accurately.

#### 3.1 The Structure of a Position Error Gradient

The jacobian of the kinematic map has been generally recognized as being critical to the implementation of feedback from task space for some time [29], and increasing attention has been expended in its analytical study [19], [21]. Since this jacobian is a critical component of the error gradients used by our feedback schemes, its structure will be of great importance in any hardware implementation of such controllers as well. We will use insights similar to those of [21], [29] in this section - refer to Appendix B for definitions and computational details. For ease of exposition, we confine our attention to the control of a manipulator which has all revolute joints in the context of a task which calls for the control of end-effector position only. Insights in the ensuing discussion apply to more general circumstances - robots with prismatic joints; tasks requiring orientation control as well.

It is shown in Appendix B that the jacobian of the forward kinematic position transformation,  $dr_g(q)$ , is given by

$$[J(z_0)r_g, J(z_1)[r_g - r_1], J(z_2)[r_g - r_2], \dots, J(z_{n-1})[r_g - r_{n-1}], ] ,$$

where

$$J(a) \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

is a skew symmetric matrix defined by  $a \triangleq \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \in \mathbf{R}^3$ . Recall, now, that the position error gradient of equation (8) is given by

$$[d_g r_g]^T K_1 (r_g - r_d) = \begin{bmatrix} [r_g - r_d]^T K_1^T J(z_0) r_g \\ [r_g - r_d]^T K_1^T J(z_1) [r_g - r_1] \\ \vdots \\ [r_g - r_d]^T K_1^T J(z_{n-1}) [r_g - r_{n-1}] \end{bmatrix},$$

and this may be interpreted geometrically by various appeals to the following identities from vector algebra. For any  $a, b, c \in \mathbf{R}^3$ ,

$$c^T J(a) b = c^T (a \times b) = |c, a, b| = |a, b, c| = |b, c, a|,$$

where  $|a, b, c|$  denotes the determinant of the array whose columns are  $a, b, c$ . For instance, if  $K_1$  is the identity matrix, then these identities imply that the  $i^{\text{th}}$  entry of the position error gradient is given by the determinant

$$|z_i, (r_g - r_i), K_1(r_g - r_d)|.$$

In other words, when its velocity is zero, the  $i^{\text{th}}$  joint will have a non-zero torque input applied unless its axis of motion is parallel to the plane spanned by the position error vector and the difference vector between its own origin and the base. In the case that  $K_1 = I$ , this determinant takes an even simpler form, since the reference frame of these measurements is unimportant

$$|z_i, (r_g - r_i), (r_g - r_d)| = |R_j| |{}^j z_i, ({}^j r_g - {}^j r_i), ({}^j r_g - {}^j r_d)|,$$

where  $|R_j| = 1$ . In particular, when  $j = i$ , we have

$$|{}^i z_i, {}^i r_g - 0, {}^i r_g - {}^i r_d| = |{}^i z_i, {}^i r_g, {}^i r_d|. \quad (9)$$

with  ${}^i z_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ .



## 3.2 Measure or Compute?

In evaluating possible feedback implementations the performance measure, as always, is some combination of speed and accuracy. Rule of thumb seems to have it that good controllers for linear time invariant systems should have time constants of roughly an order of magnitude greater than those of the system being controlled [6]. Paul [20] cites a factor of 15 in the context of robotic systems. If the time constant of a typical dc-servo is 100 ms, then we must achieve a control rate of better than one command every 10 ms. Since the effects of discretization in the context of nonlinear dynamics are not well understood, it seems preferable to attain a rate closer to 1 ms. On the other hand, traditional insight of the control community has been that a closed loop dramatically lowers sensitivity to reference signal errors at the expense of high sensitivity to feedback errors. While this folklore may be made fairly precise in the context of linear time invariant systems, no such analysis is available in the present context. Although we are attempting to develop a sensitivity analysis for the schemes presented, the obvious strategy is to achieve a computational accuracy at every step which is better than the resolution of the best joint position sensor on the arm.

The feedback controller, (8), requires far less information about the underlying plant than inverse dynamics feedforward schemes, e.g. [18], and, in the absence of gravitationally induced torques, there is not even a need for the dynamical parameters of the arm or its load. Accurate estimates of expressions of the form (9), which were shown to derive from the arm kinematics, however, are critical. The somewhat vague term “derive” is used intentionally here since there is a broad range of sensory modalities which will determine the particular form of computation. This range runs all the way from massive computations based upon a priori estimates of the underlying kinematic parameters and state space measurements of  $q, \dot{q}$ , to a complete reliance upon state space measurements of velocity, and output space measurements of link positions and orientations upon which relatively simple computations might be done using analog electronic components.

Consider the latter extreme first. Equation (9) indicates that two sensors located on every link, one reporting the gripper position, the other reporting the goal position, would suffice to evaluate the jacobian with no computation besides a  $2 \times 2$  determinant. If the impedance matrix,  $K_1$ , is not the identity, then there is a little more computation, but the situation is not essentially different. Unfortunately, it is not clear what sensing technology exists that will deliver unobstructed accurate measurements of  $n$  sets of two cartesian positions every millisecond. A large number of frequency separated acoustic sensors may perform sufficiently

quickly, but may not deliver the required accuracy particularly if the arm is moving very quickly. High resolution cameras may deliver sufficient accuracy, but will likely cause far too costly delays. Certainly, a robot endowed with cartesian sensors for every link will depart significantly from any anthropomorphically motivated design.

Considering the opposite extreme, if no sensors other than state space positions and velocities,  $(q, \dot{q})$  are available - i.e. joint shaft encoders and tachometers - then we must resort to the recursive computation

$${}^0r_{i+1}(q_1, \dots, q_{i+1}) = {}^0r_i(q_1, \dots, q_i) \cdot {}^i r_{i+1}(q_{i+1}),$$

whose details are given in Appendix B. Essentially, this amounts to computing the forward kinematic transformation for the position of each link with respect to the base frame, then taking differences and determinants. This is an obvious application for parallel computation, and we are beginning to evaluate appropriate architectures for implementation at Yale.

A variety of sensing schemes for delivering partial information regarding cartesian link measurements are currently being evaluated within the laboratory as well. For instance a very accurate picture of task space delivered at a rather low data rate might, nevertheless, be useful in correcting the results of computation. Or very accurate relative position error measurements in the gripper frame alone might be very cheaply delivered by a simple sonar device and used to increase the resolution of the absolute distance,  $r_g - r_d$ , computed from joint measurements.

## 4 Conclusion

This paper concerns a stable method of controlling robot arms in the command of tasks which have been defined by geometric descriptions and impedance relations in the task space. The control methodology is entirely feedback based, thus takes into account the intrinsic dynamics of the manipulator without their explicit computation, and obviates the need for inverse kinematic computation. Some insight into a number of difficulties is gained by interpreting these ideas as a scheme to use the robot arm as an analog computer which sequentially solves its own inverse kinematics equations. An initial discussion of instrumentation for these schemes is presented: there is seen to be a tradeoff between sensory and computational complexity, which inverts the traditions of the control community by suggesting a preference for output rather than state feedback.

Although several robotic task methodologies [8], [11], are very simply implemented using such schemes, it is not yet clear how to establish a general framework for translating abstract tasks into the appropriate feedback structure. Very likely, it will prove more effective to resort to some reference signal based encoding for certain task classes, in which cases sensible feedforward pre-compensators will be desirable as well. Since the use and importance of the various sensory modalities available to a robot will depend upon the control methods used in commanding a task, the problem of developing principles for the generation of *dynamically sound* task encoding methodologies should be explicitly addressed in an interdisciplinary fashion by “high level planners” as well as “joint torque level” control theorists.

## A Some Notation

If  $f : \mathbf{R}^n \rightarrow \mathbf{R}^M$  has continuous first partial derivatives, denote its  $m \times n$  jacobian matrix as  $df$ . When we require only a subset of derivatives, e.g. when  $x = \text{vector } x_1 x_2$ , and we desire the jacobian of  $f$  with respect to the variables  $x_1 \in \mathbf{R}^{n_1}$ , as  $x_2$  is held fixed, we may write

$$d_{x_1} f \triangleq df \begin{bmatrix} I_{n_1 \times n_1} \\ 0 \end{bmatrix}.$$

If  $A \in \mathbf{R}^{n \times m}$ , the “stack” representation of  $A \in \mathbf{R}^{nm}$  formed by stacking each column below the previous will be denoted  $A^S$ . If  $C \in \mathbf{R}^{p \times q}$ , and  $A$  is as above then the Kronecker Product of  $A$  and  $C$  is

$$A \otimes C \triangleq \begin{bmatrix} a_{11}C & \dots & a_{1m}C \\ a_{21}C & \dots & a_{2m}C \\ \vdots & & \vdots \\ a_{n1}C & \dots & a_{nm}C \end{bmatrix} \in \mathbf{R}^{np \times mq}.$$

Finally, if  $B \in \mathbf{R}^{m \times p}$ , and  $A$  and  $C$  are as above, then it can be shown that

$$[ABC]^S = (C^T \otimes A)B^S.$$

[5]

## B Kinematic Transformations: the Algebra and Calculus of Frames

Define a *frame* to be the ordered pair

$$r \triangleq (r, R) \in \mathcal{W} \triangleq \mathbf{R}^3 \times SO(3)$$

where

$$SO(3) \triangleq \{R \in \mathbf{R}^{3 \times 3} | R^T R = I \text{ and } |R| > 0\}$$

is the set of rotations on  $\mathbf{R}^3$  with positive orientation. For the purposes of this paper it will do no harm to confuse  $r, R$ , and  $r$  with their matrix representations in  $\mathbf{R}^3, \mathbf{R}^{3 \times 3}$ , and  $\mathbf{R}^{3 \times 4}$ ,

respectively. Thus, the transpose, and stack conventions of the previous appendix may be applied meaningfully to any frame,  $r$ . Moreover, left matrix multiplication of a frame,  $p$  by any rotation,  $R \in SO(3)$  is well defined, and will be used in computation as required, denoted by square brackets as

$$R[p] \triangleq [Rp, RP].$$

At times, we shall refer to the columns of  $r = (r, R) = [r, x, y, z]$  as its *position, and orientation*, or *position, and x axis, y axis, z axis*, respectively. The frames form a group under the binary product

$$p \cdot r \triangleq (Pr + p, PR)$$

whose identity is  $i \triangleq (0, I)$ , and where  $r^{-1} = (-R^T r, R^T)$ .

A *joint transformation* is a map from  $\mathbf{R}^1$  into  $\mathcal{W}$  which relates a coordinate system fixed in link  $i - 1$  to one fixed in link  $i$  through the action of the  $i^{\text{th}}$  joint. According to the standard conventions, the joint transformation may be written

$${}^{i-1}r_i = \left( \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ \delta_i \end{bmatrix}, \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & -\sin \alpha_i \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & \sin \alpha_i \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \right)$$

where either  $\theta_i$  or  $\delta_i$  is the joint variable depending upon whether the joint is revolute or prismatic, respectively, and the other kinematic parameters are defined in the link body, e.g. as in [20]. More generally, a *kinematic transformation* is a map  $r_n : \mathcal{J} \rightarrow \mathcal{W}$  which is the group product of  $n$  joint transformations,

$$r_g(q) = {}^0r_1 \cdot {}^1r_2 \cdot \dots \cdot {}^nr_g,$$

where  $n$  is the number of degrees of freedom of the chain. For any partial product we will use the notation

$${}^i r_j \triangleq {}^i r_{i+1} \cdot {}^{i+1} r_{i+2} \cdot \dots \cdot {}^{j-1} r_j,$$

which is a representation of the coordinate system  $j$  in terms of  $i$ . Individual columns will be labeled similarly: e.g.  ${}^i x_j$  denotes the second column of  ${}^i r_j$ . For  $i = 0$ , we will drop the preceding superscript.

It is well known that any element,  $R \in SO(3)$ , may be written in *exponential form*

$$R = e^J,$$

where  $J$  is skew symmetric. As is also well known, there is an isomorphism between  $\mathbf{R}^3$  and the set of  $3 \times 3$  skew symmetric matrices which uniquely identifies a point,  $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \in \mathbf{R}^3$ , with

$$J(a) \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

whose action on any  $b \in \mathbf{R}^3$ ,  $J(a)b = a \times b$ , represents the vector cross product of  $b$  with  $a$ . Note that for any  $R \in SO(3)$ , we have

$$R(a \times b) = Ra \times Rb$$

which implies

$$R^T J(a) R = J(R^T a).$$

These ideas are presented, for instance, in [3].

After fixing a little more notation, it becomes possible compute the jacobian of kinematic transformations quite readily. From the definition of the joint transformation, above,

$${}^{i-1}r_i = e^{\theta_i J(z_0)}[p]$$

where

$$p \triangleq (x_0 + \delta_i z_0, e^{\alpha_i J(z_0)}),$$

and  $x_0, y_0, z_0$  are matrix representations of the base coordinates with respect to themselves - i.e. unit vectors with the appropriate zero and one entries. It follows that the jacobian of  ${}^{i-1}r_i(q_i)$  - in this case, the ordinary derivative with respect to  $q_i$  - is given by

$$d_{q_i} {}^{i-1}r_i = J(z_0)[{}^{i-1}r_i],$$

if the joint is revolute, and

$$d_{q_i} {}^{i-1}r_i = (z_0, 0_{3 \times 3})$$

if the joint is prismatic. Note that the jacobian of a joint transformation is a linear map from the tangent space of  $\mathbf{R}^1$  at  $q_i$  to the tangent space of  $\mathcal{W}$  at  ${}^{i-1}r_i$ . By a slight abuse of notation

we may represent an image point of this map in frame notation as a point in  $\mathbf{R}^{12}$  for which the group product “ $\cdot$ ” is still defined (although it is not in the group) and write

$$d_{q_i} {}^{i-1}r_i = j_i \cdot {}^{i-1}r_i,$$

where

$$j_i = \begin{cases} (0, J(z_0)) & \text{i revolute} \\ (z_0, 0_{3 \times 3}) & \text{i prismatic} \end{cases}$$

Using this notation, the computation of the jacobian of a single joint transformation under right and left translation in the group is given by

$$d(p \cdot r \cdot s) = P[j_i \cdot {}^{i-1}r_i \cdot s]$$

where  $p$  and  $s$  are constant frames.

The jacobian of the kinematic transformation is a linear map from  $TJ_q \approx \mathbf{R}^n$  to  $T\mathcal{W}_{r_n(q)} \subset \mathbf{R}^{12}$  which may be represented as a  $12 \times n$  matrix if we pass to the stack representation of  $T\mathcal{W}_{r_n(q)}$ . This takes the form

$$dI_g^S = [d_{q_1} r_g^S, d_{q_2} r_g^S, \dots, d_{q_n} r_g^S],$$

and, since the group product is associative,

$$\begin{aligned} &= [d_{q_1} ({}^0r_1 \cdot {}^1r_g)^S, d_{q_2} ({}^0r_1 \cdot {}^1r_2 \cdot {}^2r_g)^S, \dots, d_{q_n} ({}^0r_{n-1} \cdot {}^n r_g)^S] \\ &= [(j_1 \cdot {}^0r_1 \cdot {}^1r_g)^S, (R_1[j_2 \cdot {}^1r_2 \cdot {}^2r_g])^S, \dots, (R_{n-1}[j_n \cdot {}^{n-1}r_g])^S]. \end{aligned}$$

To summarize, the general form of the  $i^{th}$  column of the jacobian,  $dI_n^S$  is given by

$$\begin{bmatrix} R_{i-1} J_i {}^{i-1}r_g + R_{i-1} j_i \\ R_{i-1} J_i {}^{i-1}R_g x_0 \\ R_{i-1} J_i {}^{i-1}R_g y_0 \\ R_{i-1} J_i {}^{i-1}R_g z_0 \end{bmatrix}$$

which evaluates to

$$\begin{bmatrix} R_{i-1} J(z_0) {}^{i-1}r_g \\ R_{i-1} J(z_0) {}^{i-1}x_g \\ R_{i-1} J(z_0) {}^{i-1}y_g \\ R_{i-1} J(z_0) {}^{i-1}z_g \end{bmatrix} = \begin{bmatrix} J(z_{i-1})[r_g - r_{i-1}] \\ J(z_{i-1}) {}^0x_g \\ J(z_{i-1}) {}^0y_g \\ J(z_{i-1}) z_g \end{bmatrix},$$

when the  $i^{\text{th}}$  joint is revolute, and

$$\begin{bmatrix} z_{i-1} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

when the  $i^{\text{th}}$  joint is prismatic. This computation, is, of course, identical to that obtained by Whitney [29] and Sastry and Paden [21].



## References

- [1] M. Takegaki, and S. Arimoto, "A New Feedback Method for Dynamic Control of Manipulators" *J. Dyn. Syst.* Vol 102, pp.119-125, June, 1981.
- [2] F. Miyazaki and S. Arimoto, "Sensory Feedback Based On the Artificial Potential for Robot Manipulators" *Proc. 9th IFAC* Budapest, Hungary, July, 1984.
- [3] V. I. Arnold, *Mathematical Methods of Classical Mechanics* Springer-Verlag, NY, 1978
- [4] H. Asada, T. Kanade, and I. Takeyama, "Control of a Direct Drive Arm" *ASME J Dyn. Syst.* 105(3):136-142., 1983.
- [5] *Introduction to Matrix Analysis* McGraw Hill, NY, 1965
- [6] G. F. Franklin and J. D. Powell, *Digital Control of Dynamical Systems* Addison Wesley, Reading MA, 1980.
- [7] E. Freund, "Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators" *Int. J. Robotics Res.* 1(1): 65-78, 1982.
- [8] N. Hogan, "Impedance Control: An Approach to Manipulation, Part I: Theory" *ASME J. Dyn. Syst.* (to appear) 1985
- [9] J. M. Hollerbach, "A Recursive Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation and Complexity", in Brady, et. al. (eds) *Robot Motion*, pp. 73-87, MIT Press, 1982.
- [10] S. Jacobsen, J. Wood, D. F. Knutti, and K. B. Biggers, "The Utah/MIT Dextrous Hand: Work in Progress" *Int. J. Rob. Res.* Vol.3, No.4, Winter, 1984
- [11] O. Khatib, "Dynamic Control of Manipulators in Operational Space" *Sixth IFTOMM Congress on Theory of Machines and Mechanisms*, New Dehli, 1983, p. 10
- [12] D.E. Koditschek, "Natural Motion for Robot Arms" *IEEE Proc.* 23<sup>rd</sup> CDC, Las Vegas, December, 1984, pp. 733-735
- [13] D. E. Koditschek, "Natural Control of Robot Arms" Yale Center for Systems Science Technical Report No. 8409, Dec. 1984 (revised, Mar. 1985).

- [14] D. E. Koditschek, "Adaptive Strategies for the Control of Natural Motion" *Proc. IEEE 24th CDC*, Fort Lauderdale, Dec. 1985.
- [15] R. H. Lathrop, "Parallelism in Manipulator Dynamics", *Int. J. Robotics Res.* 4:2, pp. 80 - 102, summer 1985.
- [16] K. S. Narendra and Y. H. Lin "Design of Stable Model Reference Adaptive Controllers", in *Applications of Adaptive Control*, Narendra and Monopoli (eds.), Academic Press, 1980.
- [17] K. S. Narendra and L. S. Valavani, "Stable Adaptive Observers and Controllers" *Proceedings of the IEEE* vol. 64, no. 8, August, (1976)
- [18] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "Resolved Acceleration Control of Mechanical Manipulators" *IEEE Tran. Aut. Contr.* AC-25 pp. 468-474, 1980.
- [19] D. E. Orin, and W. W. Schrader "Efficient Computation of the Jacobian for Robot Manipulators" *Int. J. Rob. Res.* 3(4), pp. 66-75, Winter, 1984.
- [20] R. P. Paul, *Robot Manipulators, Mathematics, Programming, and Control* MIT Press, Cambridge, MA, 1981.
- [21] B. E. Paden and S. S. Sastry "Geometric Interpretation of Manipulator Singularities" Memo. No. UCB/ERL M84/76, Electronics Research Laboratory, College of Engineering, UC Berkeley, Sept., 1984
- [22] J. Reif, "Complexity of the Mover's Problem and Generalizations", Proc. 20th Symposium of the Foundations of Computer Science, 1979.
- [23] S. Smale, "The Fundamental Theorem of Algebra and Complexity Theory" *Bull. AMS.* vol. 4, no. 1 pp. 1,36, Jan. (1981)
- [24] M. W. Hirsch and S. Smale, "On Algorithms for Solving  $f(x) = 0$ ", *Comm. Pure and Appl. Math.*, vol. XXXII, pp. 281-312 (1979)
- [25] J. T. Schwartz and M. Sharir, "On the Piano Mover's Problem. II." NYU Courant Institute, Report No. 41, 1982.
- [26] Sir W. Thompson and P. G. Tait, *Treatise on Natural Philosophy*, University of Cambridge Press, Cambridge, 1886.

- [27] T.J. Tarn, A. K. Bejczy, A. Isidori, Y. Chen, "Nonlinear Feedback in Robot Arm Control"  
*Proc. 23rd IEEE Conf. on Dec. and Control*, Las Vegas, Dec. 1984, pp. 736 - 751
- [28] W. A. Wolovich and H. Elliott, "A Computational Technique for Inverse Kinematics"  
*Proceedings of the Twenty Third IEEE Conference on Decision and Control* pp. 1359 - 1364 (1984)
- [29] D. E. Whitney "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators" *ASME. J. Dyn. Syst. Contr.*, Vol 122, pp. 303-309, Dec, 1972