ROBOT-CONTROL SYSTEMS

The practice of automatic control has its origins in antiquity (1,2). It is only recently—within the middle decades of this century—that a body of scientific theory has been developed to inform and improve that practice. Control theorists tend to divide their history into two periods. A "classical" period prior to the sixties witnessed the systematization of feedback techniques based on frequency-domain analysis dominated by applications to electronics and telephony. A "modern" period in the sixties and seventies was characterized by a growing concern with formal analytical techniques pursued within the time domain motivated by the more stringent constraints posed by space applications and the enhanced processing capability of digital technology (3). The hallmark of control theory has been, by and large, a systematic exploitation of the properties of linear dynamic systems whether in the frequency or time domain. Its great success in applications is a remarkable tribute to the diverse range of physical phenomena for which such models are appropriate.

The field of robotics (qv) presents control theorists with a fascinating and novel domain. Although numerically controlled kinematic chains with a few degrees of freedom have been available for over two decades, it is only within the last five years that mechanical systems with many degrees of freedom, each independently actuated, have been wedded to dedicated computational resources of considerable sophistication. To begin with, the dynamic behavior of such systems appears to depart dramatically from the familiar linear case. Further, typical robotic tasks involve complex interactions with diverse environments consisting of themselves, kinematics and dynamics, which may change abruptly throughout the course of desired operations. Finally, the complexity of these tasks makes even their specification problematic for purposes of control.

This entry attempts simultaneously to provide general readers an introductory account of the nature of control theory and its application to robotics, and offers control and robot theorists a brief (and necessarily incomplete) look at the contemporary research horizons. For concreteness, the discussion is limited to robot arms—open kinematic chains with rigid links. For reasons of brevity, much of the tutorial discussion is concerned with robot motion alone, even though general manipulation tasks are the more interesting application. The first section, an elementary introduction to the problems and methods of general control theory, introduces some fundamental properties of dynamic systems, the nature of stabilizing feedback structures, and the capabilities of the servomechanisms that result, all in the context of a very rudimentary "robot arm"—the simple pendulum. Hopefully, a general college physics course and a semester's introduction to differential equations will be background enough to gain some appreciation of the material.

The tone and mathematical sophistication of this entry shifts in the sequel in favor of a more specialized reader. General Robot-Arm Dynamics presents a brief review of Lagrange's equations, leading to the definition of the general rigid-body model of robot dynamics, and an equally brief examination of its more significant omissions. Feedback Control of General Robot Arms concerns the application of general feedback techniques to the problem of robot task encoding and control. This treatment departs from traditional control theory by paying as much attention to issues of task definition and encoding as to control algorithms. The intent is to suggest how relatively simple error-driven control structures can afford the command of a surprisingly rich array of robot tasks. Servo Control of General Robot Arms is more in the character of a survey of the contemporary robot-arm-control literature. It will be observed that this literature returns to the traditional control theoretic task-encoding paradigm of the servomechanism. Unlike any of the previous sections, the discussions to be found in these last two sections are incomplete not primarily in consequence of limited space but because the relevant theory has not yet been developed. Some attempt will be made to point out the important gaps.

Throughout the entry technical terms of general importance for the exposition will be italicized, given a reference for more detail reading, as well as defined in subsequent text if the meaning is not clear from the context. Other technical terminology that may be current in the literature will be simply placed within quotation marks and used collegially. More idiosyncratic terminology and notation is defined in the appendix.

Single-Degree-of-Freedom Robot Arm

This section provides an introduction to the established body of control theory as developed during the "classical" period, yet expounded in the "modern" language (referring to the historical periods introduced above). The "frequency-domain" techniques of classical control theory lie at the foundation of the discipline and offer design methods proven over the last 60 years in the context of a great variety of physical problems. Unfortunately, the class of dynamic systems represented by high-performance robots with revolute arms is not amenable to a general rigorous analysis using these tools: strongly coupled nonlinear dynamics do not admit representation by transfer function. Some researchers, e.g., Herowitz (4), have successfully used modified frequency-domain methods for restricted nonlinear control problems. For the purpose of a general tutorial, the central insights of control theory may be translated quite readily into the more widely familiar elementary methods of ordinary differential equations. Fortunately, similar (although rather more advanced) methods afford the translation of some of these into the general robotic domain as well, as will be demonstrated in the last two sections. Thus, by avoiding the language of transfer functions, we seek to reach a broader audience in this section, while motivating the more technical discussion in future sections.

In this section attention is limited to the case of a single-degree-of-freedom mechanical control system—the actuated
simple pendulum. Although this system cannot convey the scope of 60 years of control research, the insights motivated by such second-order linear-time-invariant systems pervade the field. At the same time this system represents the simplest possible revolute robot arm.

The notion of a dynamical model is introduced below, as is the need for control theory, hopefully making clear that the fundamental problem of control is not a consequence of limited power but of limited information. Constraints of space have unfortunately precluded the addition of sections concerning many "modern" techniques such as optimal control, stochastic filtering, or learning theory as applied to this system. The reader may note that each subsection here anticipates a more specialized treatment of analogous material for the general robot arm in later sections.

Dynamics: A Source of Delay and Uncertainty. A simple pendulum consists of a mass \( M \) attached via a rigid (massless) link to a joint that permits rotational motion limited to the plane on which the link lies. Perfect angular position and velocity sensors located at the joint deliver exact measurements, \( \theta, \dot{\theta} \), respectively, continuously and instantaneously. A perfect actuator has been placed at the joint: this idealized device has no power limitations and hence can deliver arbitrarily large torques, \( \tau \), instantaneously. We assume that the plane of motion is horizontal so that there are no gravitational or other disturbance torques. Such artificial assumptions are relaxed very soon and serve here merely to underscore the insight that the fundamental problems of control arise from uncertain information and intrinsic delay rather than power constraints, as discussed above.

The control problem may be rendered roughly as follows: design an algorithm that produces a time profile of torques, \( \tau(t) \), so as to elicit some specified behavior of the simple pendulum. In the context of robotics, the following terminology (which is alien to conventional control theory) proves quite useful. The precise nature of the desired property determines what might be called the task domain, and the particular instance, an encoded task specification, or plan.

Newtonian Dynamic Model. In order to think about control algorithms, we first require some understanding of the relationship between adjustments in \( r \) and resulting changes in \( \theta \). This relationship is completely specified by Newton's law relating torque to angular acceleration:

\[
M \ddot{\theta} = \tau
\]

This is a system with memory: changes in \( \dot{\theta} \) (and, hence, \( \theta \), itself) at any time \( t \) depend on the past history of \( r \) rather than simply on its value at time \( t \). The fact that physical systems give rise to dynamic rather than memoryless relationships necessitates the need for a theory of control, as will be shown directly.

For the purposes of this entry it will prove more convenient to express such relationships, second-order differential equations involving \( n \) variables, in the equivalent form of first-order differential equations involving \( 2n \) variables. Defining to be the state variable expressed in phase space emphasizes the fact obtaining from elementary properties of differential equations that the future behavior of the system, \( x(t), t > t_0 \), is entirely determined by its initial conditions, \( x_{00} \triangleq \theta(t_0), x_{00} \triangleq \dot{\theta}(t_0) \), and future values of the control input, i.e., the torque, \( u(t) \triangleq \tau(t), t > t_0 \). For time-invariant systems such as this, the behavior is independent of initial time, and it will be assumed in the sequel that \( t_0 \triangleq 0 \). These definitions are more carefully discussed in standard control texts (6–7).

The system may now be specified by phase-space dynamics of the form

\[
x' = f(x, u)
\]

with the vector field (8) given as

\[
f(x, u) = A_\theta x + bu
\]

where

\[
A_\theta \triangleq \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad b \triangleq \begin{bmatrix} 0 \\ 1/M \end{bmatrix}
\]

According to the description of the system given above, the entire state variable is measured; thus, the system output takes the form

\[
y(t) = x(t)
\]

A traditional circumstance of linear control theory is that the system output—the set of available measurements—contains incomplete information from which it is required to reconstruct the entire state. In the context of robotics this situation is most typically reversed: the system state (joint positions and velocities) are available, whereas the output (work space positions, velocities, and forces) cannot easily be directly measured.

Need for a Theory of Control. In the task domain of set-point regulation consider the following task specification: bring \( \theta \) to some desired position \( \theta_d \) and keep it there. Given the ideal context described above, it is quite easy to come up with ad hoc control algorithms.

A unit impulse is modeled by the "Dirac delta function," \( \delta_t(\zeta) \), defined by

\[
\int_{-\infty}^{\infty} f(\zeta)\delta_t(\zeta) d\zeta = f(t)
\]

an infinitesimally rapid amount of infinitely large magnitude possessed of unit area (finite energy) that our ideal actuator is able to deliver. An obvious procedure that requires no great body of theory might be the following. Measure the present position and velocity, \( x(0) = [x_{10}, x_{20}]^T \), and apply an impulse torque at the same instant,

\[
u_{\text{start}}(t) \triangleq M(1 - x_{20})\delta(t)
\]

An impulse has the effect of resetting initial conditions; thus Eq. 1 implies that a new constant angular velocity results,

\[
x(t) = \begin{bmatrix} t + x_{10} \\ 1 \end{bmatrix}, \quad t > 0
\]

and the desired position, \( x_1(t) = \theta_d \), is achieved at time \( t' \triangleq \theta_d - x_{10} \). If, at this instant, a second impulse,

\[
u_{\text{stop}} \triangleq -M\delta_{t'}(t)
\]
is applied in the opposite direction of motion, the new velocity is canceled exactly, and the end effector comes to rest in the desired position and remains there for all \( t > t' \). The algorithm,

\[ u = u_{\text{start}}(t) + u_{\text{stop}}(t) \]

requires a priori knowledge of \( M \); instantaneous measurement of and actuation energy exactly proportional to \( x(0) \); and an exact timer for marking \( t' \). Note that despite our best efforts to idealize the capabilities of sensors and actuator, some finite time must necessarily elapse between the application of the control torque and the desired result. This, too, could be obviated by the further idealization of applying a “unit doublet”—the derivative of the delta function. Such a degree of unreality is not introduced here because, unlike the impulse, the doublet cannot be even approximated by real actuators.

Now suppose that the prior estimate of \( M, \dot{M} \), has some error (e.g., suppose the robot is holding an object whose mass is not known a priori). If the same control is applied, substituting \( \dot{M} \) for \( M \), the true response of the system will be

\[ x(t) = \begin{bmatrix} \frac{\dot{M}}{M} + \left(1 - \frac{\dot{M}}{M}\right)x_20 \cdot t' + \left(1 - \frac{\dot{M}}{M}\right)x_20 \cdot t \\ \frac{1}{M} \end{bmatrix} t > t' \]

Thus, finite and increasing error results from arbitrarily small inaccuracy in \( \dot{M} \). It is easy to see that the same problems would result given any inaccuracies in the sensors or magnitude of energy delivered by the actuators. Certainly, a subsequent check of true response at some future time, \( x(t + \epsilon) \), would reveal whether or not such errors had occurred, and a similar course of action could be planned based on the new observation. But there is no systematic procedure for performing such checks and readjustments: a “higher level” of control authority would be required to decide when they should be effected. More disturbing, it is not yet clear that any systematic application and reappraisal of this procedure exists that can guarantee subsequent improvement from one response to the next. This sort of error propagation is characteristic of unstable systems.

The origins of control theory, then, rest in the following observations. Dynamic systems give rise to delay that must be taken into account by any control strategy regardless of available actuator power and sensor accuracy. Moreover, information regarding the real world is inevitably uncertain and may have adverse effect on performance no matter how small the uncertainty or powerful and accurate the apparatus.

Feedback Control: Behavior of Error-Driven Systems. The difficulties described above suggest the desirability of control strategies that make systematic and continual use of actuator information in order to reduce successively the performance errors caused by initial uncertainty. This study—the discovery and elucidation of feedback algorithms—is arguably the most profound contribution of control theory to physical science. Here we present some of the basic results from a purely control-theoretic perspective. Below, suitable generalizations begin to suggest a unified approach to dynamically sound robot task-encoding methodologies.

A feedback algorithm is essentially an error-driven control law. Presented with a linear system, it makes sense to investigate feedback laws that are linear in the errors as well. In the context of set-point regulation, the errors in question are the distance of the true angular position from the desired and the true angular velocity from zero. The most general linear function of these two errors is

\[ u \triangleq \gamma_1(\theta_d - \theta) + \gamma_2(\theta_2 - \theta_2) \]

defining a class of algorithms known as PD ("proportional and derivative") control schemes. According to our model of system dynamics (Eq. 1), the resulting closed-loop system is a homogeneous linear-time-invariant differential equation,

\[ \dot{y} = A_1y \triangleq \begin{bmatrix} 0 & 1 \\ -\gamma_1 & -\gamma_2 \end{bmatrix} y \]

in the translated coordinate system,

\[ y \triangleq \begin{bmatrix} x_1 - \theta_d \\ x_2 \end{bmatrix} \]

The desired end position is an equilibrium state of this closed-loop system; i.e., if the initial position and velocity of the arm were exactly at \((\theta_d, 0)\) to begin with, the resulting future trajectory would remain there for all time. An equilibrium state of a dynamical system that has the property that solutions originating sufficiently near remain near and asymptotically approach it in the future is called asymptotically stable. All those initial conditions that are near enough to asymptotically approach an asymptotically stable equilibrium state are said to lie within its domain of attraction (9).

Stability of Closed-Loop System. We seek to show that this algorithm produces an asymptotically stable closed-loop equilibrium state (the desired end point) whose domain of attraction includes all positions and velocities. This desirable property may be shown to hold in a number of ways: the following demonstration is the only means that may be rigorously extended to the general case of revolute arms with many degrees of freedom, as shown below.

It may be observed that the algorithm and consequent closed-loop dynamics would be the intrinsic result of introducing a physical spring, with constant \( \gamma_1 \), stretched between the desired and true position, along with a viscous damping mechanism opposing motion with force proportional to velocity, with constant \( \gamma_2 \). Accordingly, considerable insight into the asymptotic behavior of the resulting system may be obtained by studying its mechanical energy.

This is defined as the sum, \( \nu = \kappa + \mu \) of kinetic energy, due to the velocity of the mass,

\[ \kappa \triangleq \frac{1}{2} M y^2 \]

and potential energy stored in the spring,

\[ \mu \triangleq \frac{1}{2} \gamma_1 y^2 \]

The change in energy of the closed-loop system is expressed as

\[ \dot{\nu} = \gamma_1 y_1 \dot{y}_1 + M \dot{y}_2 \dot{y}_2 \]

\[ = \gamma_1 y_1 \dot{y}_1 - \gamma_1 y_2 \dot{y}_2 - \gamma_2 y_2^2 \]

\[ = -\gamma_2 y_2^2 \leq 0 \]

(4)
hence, if $\gamma_2 > 0, v$ must decrease whenever the velocity is not zero. If $M, \gamma_1$ are positive as well, $v$ is positive, except at the desired state, $y = 0$, where $\theta = \theta_0$, $\theta = 0$. It is intuitively clear (and will be rigorously demonstrated in greater generality within the proof of Theorem 1 below) that these conditions guarantee $v$ will tend asymptotically toward zero and, hence, that $y(t)$ approaches zero as well.

This argument employs the total energy as a Lyapunov function (9). It demonstrates that the control algorithm succeeds, asymptotically, in accomplishing the desired task: stabilizing the system with respect to $[\theta_0 0]$ solves the set-point regulation problem for that end point. It has been already remarked that this stability property is global in the sense that any initial position and velocity of the robot arm will "decay" toward the desired equilibrium state. Moreover, no exact information regarding the particular value of $M$ has been used to achieve the result other than the assumption that it is positive. Since the choice of $\theta_0$ was arbitrary, it is also clear that the analogous feedback algorithm will stabilize the system around any other desired zero velocity state, $[\theta_0 0]$, as well, with no further readjustment of the "feedback gains," $\gamma_1, \gamma_2$. The PD controller provides a general solution to the set-point regulation problem.

Robust Properties of Stable Systems. The model proposed for the single-degree-of-freedom arm (Eq. 1) cannot be exactly accurate: there will be inevitable small disturbance forces and torques placed on the shaft and arm varying in position and over time. Moreover, real actuators, even those possessed of ample power, are subject to imprecision in the profile of torques or forces output in response to any command. If the cumulative effect of all these uncertainties is small, Eq. 3 may be more accurately written in the form

$$\dot{y} = A_1 y + b e(y, t)$$

where the scalar "noise" function is bounded, $|e(y, t)| < e_0$, by some small $e_0 > 0$. Another advantage of the feedback-control algorithm developed above is that the inevitably resulting errors in performance remain strictly smaller than $e_0$, no matter what the form of the noise function, $e(y, t)$. This, again, may be demonstrated in a variety of ways: in keeping with the philosophy of exposition detailed in the beginning of this section, we appeal to a modified Lyapunov analysis.

Define the symmetric matrix

$$P \equiv \begin{bmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & M_A \end{bmatrix}$$

and define the modified Lyapunov candidate,

$$v \triangleq \frac{1}{2} y^T P y = \frac{1}{2} \gamma_1 y_1^2 + \gamma_2 y_1 y_2 + M y_2^2$$

The analytical results below depend on the fact that this is a positive definite function—i.e., $y^T P y \geq 0$ with equality if $y = 0$. For this, it is necessary and sufficient that $P$ be a positive definite matrix—i.e., $\gamma_1, M > 0$ and

$$M > \frac{\gamma_1^2}{4\gamma_1} \quad (5)$$

which condition requires at least a known lower bound on $M$ to be verified. A slightly more complicated choice of $P$ may be found yielding the same result with no additional information requirement concerning $M$. The present analysis is intended partially to anticipate the discussion of transient performance in the next section, where relative magnitudes of all the parameters become important. Assuming this additional information is available and the conditions of Eq. 5 are met, we have

$$\dot{v} = y^T P A_1 y + y^T P b e$$

$$= -y^T P y \frac{\gamma_2}{2M} + y^T P b e$$

since $y^T P A_1 y = -y^T P y (\gamma_2/2M)$, and, completing the square,

$$y^T P A_1 y = -\frac{\gamma_2}{4M} \left[ y^T P y - \frac{4M^2}{\gamma_2^2} b^T \gamma_2^{-1} b e^2 \right]$$

$$- \frac{\gamma_2}{4M} \left[ y - \frac{2M}{\gamma_2} \gamma_2^{-1} b e \right] y - \frac{2M}{\gamma_2} \gamma_2^{-1} b e^2 \right]$$

$$\leq -\frac{\gamma_2}{4M} y^T P y - \frac{4\delta_1}{\gamma_2^2} \left[ M y_1 - \gamma_2 \right]$$

Thus, $v$, and, hence, $\|y\|$, decrease whenever

$$\|y\|^2 > \frac{4\delta_1}{\gamma_2^2} \left( M y_1 - \gamma_2 \right) \|P\|^2$$

This implies that $\|y\|$ decays asymptotically to a magnitude bounded, at most, by the constant on the right-hand side of the previous inequality. Thus, with no regard to the particular form of the disturbance, $e$, other than a fixed upper bound on its magnitude, $e_0$, the steady-state error resulting from the PD feedback algorithm can be arbitrarily small by increasing the gains.

Adjustment of Transient Response: High-Gain Feedback and Pole Placement. That the resort to stabilizing state feedback results in a system with desirable "steady-state" properties was discussed under Stability of Closed-Loop System. In the previous section that feedback algorithm is seen to reduce the sensitivity of the limiting behavior to unmodeled disturbances, $e$, and this benefit is enhanced by increasing the magnitude of the gains. Recall, however, that the latter result involved an additional assumption (Eq. 5) regarding their relative magnitudes in comparison to $M$. This section presents a more general argument for the benefits of what may be termed "high-gain feedback"—the choice of gains with the largest possible magnitude consistent with the capabilities of actuators along with the satisfaction of conditions on their relative magnitude to be developed below. In contrast to such feedback strategies, if more information is available regarding the parameters of the system to be controlled (in this case knowledge of $M$), the resulting transient behavior may be shaped much more precisely by the technique of pole placement.

A glance at Eq. 4 shows that the magnitude $\gamma_2$ governs the rate of decay of $v$, suggesting, in the first instance, that the rate of convergence to the desired position may be increased by adjusting that constant. Unfortunately, the rate of decay of $v$ is governed as well by $\gamma_1$, and this results in a more complicated situation. The effects of increasing $\gamma_2$ indiscriminately by measuring its relative magnitude according to a parameterization by the positive scalar, $\zeta$, is expressed as
\[ \gamma_2 = 2\zeta(\gamma_1 M)^{1/2} \]

usually called the “damping coefficient” in the systems literature (3). Define a direction in the translated phase plane (i.e., y coordinates),

\[
\mathbf{c} \triangleq \begin{bmatrix} \zeta + (\zeta^2 - 1)^{1/2} \\ \frac{M}{\gamma_1} \end{bmatrix}^{1/2}
\]

(this is the eigenvector of \( A_1^T \) corresponding to the smaller of its two eigenvalues), and by studying the projection onto the unit vector in this direction, \( \xi = c/\|c\|^2 \), note that the rate of decrease of this component along the response trajectories of the closed-loop system is

\[
\frac{d}{dt}(\xi^2) = 2\zeta \xi \xi \xi
\]

= \[\begin{array}{c}
2\zeta \xi \xi \xi \\
\xi \xi \xi \xi \lambda
\end{array}\]

= \[\begin{array}{c}
2\zeta \xi \xi \xi \\
\lambda \xi \xi \xi \xi \xi
\end{array}\]

where

\[
\lambda = -\left( \frac{M}{\gamma_1} \right)^{1/2} (\zeta - \zeta^2 - 1)^{1/2}
\]

(is the smaller of the two eigenvalues of \( A_1^T \)). For fixed values of \( \gamma_1 \), \( M \), \( \lambda \) approaches zero as \( \xi \) increases. Thus, the same initial position will converge to the desired position at a slower and slower rate as \( \gamma_2 \) is made much larger than \( \gamma_1 \). By guaranteeing that \( \xi \) grows at least as the square of \( \gamma_2 \), this possibility is ruled out. Note that the modified Lyapunov analysis of the previous section depends on this assumption, as does the robust tracking result discussed below.

On the other hand, significant problems result from choosing too large a magnitude of \( \gamma_1 \) relative to \( \gamma_2 \), as may be seen in standard control texts (3,10,11). However, if \( \gamma_1 \) is chosen roughly proportional to \( \gamma_2 \), it is generally safe to predict that better transient performance will result from increased gains without the explicit knowledge of \( M \) required to design the feedback algorithm (Eq. 2). It should be noted that the presence of higher order dynamics than modeled in Eq. 1 will virtually guarantee that gain increases past a certain magnitude result in deleterious performance and, ultimately, destabilize the closed-loop system. Although the desired position is never attained exactly in finite time, the time required to reach arbitrarily small neighborhoods of the desired position from a given distance away may be made arbitrarily short by increasing the magnitude of the gains. Of course, in practice, given the inevitable power constraints of the real world there will be some upper limit on the magnitude of \( \gamma_1 \), \( \gamma_2 \) that may be implemented and, hence, on the rate of convergence toward the desired goal that may be attained.

The transient characteristics of a closed-loop system resulting from linear state feedback (Eq. 2) are completely determined by the poles—eigenvalues of the resulting system matrix, \( A_1 \) in Eq. 3—and, hence, by the roots of a second-order polynomial whose coefficients are exactly specified by the second row of that array (3,10,11). Thus, concern regarding transient performance may be precisely addressed only if full information regarding the parameters of the system is available. Namely, if \( M \) is known, the feedback strategy results in a closed-loop system with vector field

\[
\mathbf{u} = M\left( k_1 \theta + k_2 \dot{\theta} \right)
\]

whose eigenvalues are exactly

\[
-k_2 \pm \left( k_2^2 - 4k_1 \right)^{1/2}
\]

is some appropriately conditioned or precompensated form of the reference signal. A thorough treatment of these ideas can be found in standard classical texts (3,10,11).

**Forced Response of Linear Systems.** Linear-time-invariant systems constitute an important exception to the general rule that the output of a forced dynamic system has no closed-form expression involving elementary functions. Consider a general example of such a system,

\[
\dot{x} = Ax + bu
\]

Let the initial position and velocity be specified by \( x = [\theta_d, \dot{\theta}_d]^T \), and define

\[
\exp(tA) \triangleq \sum_{k=0}^{\infty} (tA)^k/k!
\]

Recall that this sum converges for all matrices \( A \) and real values \( t \) and that its (operator) norm decays with increasing values of \( t \) if the eigenvalues of \( A \) have negative real part. It may be verified by direct computation that

\[
x(t) = \exp(tA)x_0 + \int_{0}^{t} \exp(-\tau A)bu(t) \, d\tau
\]

satisfies the closed-loop dynamic equations (Eq. 3). The availability of this exact “input/output description” is the basis for the powerful frequency-domain methods of classical control theory (3,10,11). As has been stated in the introduction, these techniques are deliberately avoided throughout this section since they have no rigorously founded counterparts in the context of the general nonlinear robot dynamics to be discussed in the sequel.

Nevertheless, given a specified trajectory, \( x_d(t) = [\theta_d(t), \dot{\theta}_d(t)]^T \), it is now possible to provide a complete account of the efficacy of any particular control input by examining the resulting error,

\[
e(t) \triangleq x_d(t) - x(t)
\]

Assume a control of the form \( u \triangleq u_0 + u_{pc} \), as above. The closed-loop plant equations are now of the form of Eq. 7.
\[ \dot{x} = A_1 x + b u_{pc} \]  
\[ \text{with } A = A_1, \text{ and } \dot{u} = u_{pc} = \Gamma x_2. \]  
(Since the gains \( \gamma_1, \gamma_2 \) have been chosen strictly greater than zero to stabilize the desired equilibrium position, there is no question as to the invertibility of \( A_1 \).) Integrating the second term in \( x(t) \) of Eq. 8 by parts twice affords the expression

\[ e(t) = x_d(t) - A_1^2 \{ \exp(tA_1) a_0 - A_1 b u_{pc}(t) - b \dot{u}_{pc}(t) \} \]
\[ - \int_0^t \exp((t - r)A_1) b \dot{u}_{pc}(r) \, dr \]  

where we define for convenience the controlled “initial acceleration,”

\[ a_0 \triangleq A_1^2 x(0) + A_1 b u_{pc}(0) + b \dot{u}_{pc}(0) \]

**Inverse Dynamics.** If it is desired that the true response of the closed-loop system track the reference signal exactly, the most obvious recourse is to “inverse dynamics.” Suppose \( \dot{M} \) is exactly known, and \( u_{pc} \) has been chosen in the form of Eq. 6 to achieve some specified set of poles. Assume that the output of the stabilized system Eq. 3 is exactly the desired signal, \( \dot{\theta}(t) = \dot{\theta}_d(t) \); it follows that all derivatives are equivalent as well, \( \ddot{\theta} = \ddot{\theta}_d \), \( \theta = \theta_d \) and, hence, solving for \( u \) in the second line of Eq. 9, that

\[ u_{pc}(t) = M(\ddot{\theta}_d + kT \left[ \begin{array}{c} \dot{\theta}_d \\ \theta_d \end{array} \right]) \]

In terms of the framework above, this corresponds to a choice for \( \Gamma \) of the form

\[ \Gamma_{id}[x_d] \triangleq \frac{1}{b^2} b^T(x_d - A_2 x_d) \]

Using frequency-domain analysis, it is easy to see that this control input is a copy of the reference signal fed through dynamics whose transfer function is the reciprocal of the feedback-stabilized plant. From the point of view of time-domain analysis, the resulting closed-loop system expressed in error coordinates takes the form

\[ \dot{e} = \dot{x}_d - A_2 x - \frac{1}{b^2} b b^T [\dot{x}_d - A_2 x_d] \]
\[ = A_2 e + [I - \frac{1}{b^2} b b^T] (\dot{x}_d - A_2 x_d) \]
\[ = A_2 e \]

since \([I - 1/b^2 b b^T]\) is a projection onto the subspace of \( \mathbb{R}^2 \) orthogonal to \( b \), and \( \dot{x}_d - A_2 x_d \) lies entirely in the image of \( b \). Thus, appealing to a Lyapunov analysis once more, if \( u = \dot{x}_d + \frac{1}{b^2} b b^T \dot{\theta}_d \), which implies that the error is nonincreasing from which it can be deduced as well that (see Theorem 1) \( e \) tends toward zero asymptotically. For purposes of comparison it is worth displaying the actual form of the closed-loop error, [which may be computed from the exact I/O (Eq. 10)]

\[ e(t) = -A_1^2 \{ \exp(tA_1) a_0 - \dot{x}_d \}
+ \int_0^t \exp((t - r)A_1) b \dot{u}_{pc}(r) \, dr \]

\[ = -A_1^2 \{ \exp(tA_1) [a_0 - \dot{x}_d(0)] \}
+ \int_0^t \exp((t - r)A_1) b \dot{u}_{pc}(r) \, dr \]

The only source of error is due to initial conditions \( x_0, a_0 \), which may be cancelled exactly by appropriate choice of \( u_0, \dot{u}_0 \) since \( b, A_1 b \) are linearly independent (the system is controllable). If not cancelled exactly, the term must decay asymptotically under the assumption that \( k_1, k_2 \) are stabilizing gains. Tracking is perfect, or at least asymptotically perfect, for any arbitrary input.

This strategy resembles the first open-loop control scheme introduced above in that it assumes perfect information regarding the plant dynamics. Any uncertainty in the model, its parameters, or the presence of noise will invalidate the result. Moreover, since it requires derivatives of the reference trajectory, \( x_d \), the scheme would be practicable only in cases where the entire reference trajectory is known in advance; differentiating unknown signals on-line generally results in unacceptable noise amplitudes.

**Robust Tracking.** It was seen above that the effect on the steady-state response of bounded noise perturbations could be made arbitrarily small through the use of high-gain feedback. Subsequently, it was shown that high-gain feedback increases the rate at which the system tends toward its steady state. Here, these insights will be combined and an attempt made to track \( x_d(t) \) as if it were a moving set point through the continued exploitation of high-gain feedback. Intuitively, it is hoped that the resulting tendency to steady state will be “faster” than the rate of change of the set point. As usual, the advantage attending the reliance on intrinsic stability properties of the system will be the reduced need for a priori information.

Let the feedback control be chosen using the “high-gain” philosophy as in Eq. 2. If the precompensating function is simply set to be proportional to the desired position,

\[ \Gamma_{id}[x_d] = \gamma_1 \theta_d \]

according to Eq. 10,

\[ e(t) = -A_1^2 \{ \exp(tA_1) [a_0 - \dot{x}_d(0)] \}
+ \int_0^t \exp((t - r)A_1) b \dot{u}_{pc}(r) \, dr \]

and, not surprisingly, there are terms that contribute error in proportion to the desired velocity and “time-averaged” desired acceleration. Note that if the feedback gains are increased according to the high-gain policy, namely, \( \gamma_1 = \gamma_2 \), it seems as though high-gain feedback may reduce the error introduced by velocity. This intuition may be rigorously confirmed, again by appeal to a Lyapunov argument after making the further assumption that there is some (in general unknown) bound on the desired speed,
\[
|\dot{\theta}_d| \leq \rho
\]

Defining the modified error coordinates,
\[
\dot{e} \triangleq \left[ \begin{array}{c}
\theta_d - \theta \\
\frac{\gamma_1}{\gamma_2}
\end{array} \right]
\]

the closed-loop system takes the form
\[
\dot{e} = A_e e - d
\]

where \( d \triangleq \left[ \begin{array}{c}
d \\\\frac{\gamma_1}{\gamma_2}
\end{array} \right] \). Consider, then, the Lyapunov candidate,
\[
v(e) \triangleq e^T P e
\]

where
\[
P \triangleq \left[ \begin{array}{cc}
\gamma_1 & 0 \\
0 & \frac{\gamma_2}{M}
\end{array} \right]
\]

(the same matrix defined above) is positive definite as long as the relation \( 5 \) is satisfied. The derivative along the solutions of the closed-loop error equation is
\[
\dot{v} = e^T P \dot{e} = e^T P \left[ -\frac{\gamma_2}{2M} e - d \right]
\]
\[
\leq -\frac{\gamma_2}{4M} e^T P e + \frac{M}{\gamma_2} d^T P^{-1} d
\]
\[
= -\frac{\gamma_2}{4M} e^T P e + \frac{4M^2 \beta_0^2}{\gamma_2(M\gamma_1 - \gamma_2^2/4)}
\]

If the reference signal is assumed to have a bounded derivative, \( \dot{q}_d \leq \rho \), for some constant scalar \( \rho \), the derivative is negative whenever
\[
\|e\|^2 \geq \frac{4M^2 \beta_0^2}{\|P\|\gamma_2(M\gamma_1 - \gamma_2^2)}
\]

and it follows that the error magnitude decreases monotonically at least until it reaches the constant term in this inequality. The latter may be made arbitrarily small by increasing the magnitude of the feedback gains, \( \gamma_1, \gamma_2 \).

**Adaptive Control.** The final approach to the linear-time-invariant servo problem to be considered in this entry might be said to marshal the power of schemes such as pole placement or inverse dynamics without requiring exact a priori knowledge of the system parameters. The field of adaptive control is relatively new since the first convergence results for general linear-time-invariant systems were reported only in 1980 (12). The method described here falls within the class of “model reference” schemes (13,14).

Suppose a desired trajectory, \( x_d \), has been specified along with a precompensating feedback signal, \( u_{pc} \triangleq \Gamma x_d \), which forces a known model
\[
\dot{x}_m = A_m x_m + b_m u_{pc}
\]
to track \( x_d \) acceptably. Suppose, moreover, that the true system to be controlled,
\[
\dot{x} = Ax + bu
\]
satisfies the condition \( \gamma b = b_m \) for some (unknown) positive scalar, \( \gamma > 0 \), and is known to admit a feedback law, \( u_b \triangleq [\alpha, \beta]^T x \) such that the closed-loop system yields the model behavior, \( A + \beta [\alpha, \beta] = A_m \) for some (unknown) set of gains, \( \alpha, \beta \in \mathbb{R} \). The adaptive control law takes the form
\[
u_{ad} \triangleq k^T \phi \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right]
\]

where \( k(t) = (\alpha(t), \beta(t), \gamma(t)) \) denotes a set of gain estimates that will be continuously adjusted on the basis of observed performance. Let \( k^T = [\alpha, \beta, \gamma] \) denote an unknown “true” vector of gains. The closed-loop system resulting from \( u_{ad} \) may be written in the form
\[
\dot{x} = A_m x + b_m u_{pc} + [k(t) - k]^T \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right]
\]

hence, defining the state error coordinates, \( e \triangleq x_m - x \), and the “parameter error” coordinates, \( \hat{e} \triangleq e - k \), yields the system
\[
\dot{e} = A_m e - \frac{1}{\gamma} b_m k^T \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right]
\]

Notice, further, that \( \dot{\hat{e}} = -\dot{k} \), hence, adjustments in \( \dot{k} \) afford exact adjustments of the opposite sign in the parameter error vector. The question remains, then, concerning the choice of an “adaptive law,” \( k = \Phi \hat{e}, e_{upc} \), that will make the complete error system converge.

At the very least, we may assume that \( A_m \) defines an asymptotically stable closed-loop system since the model is capable of tracking \( x_d \) in the first place. According to the theory of Lyapunov (15), it is therefore guaranteed that a positive definite symmetric matrix, \( P_m \), exists such that \( x^T P_m A_m x \leq 0 \) with equality only for \( x = 0 \). Find such a \( P_m \), and set the adaptive law as
\[
k = -\frac{1}{\gamma} K P_m b_m
\]

To show that the resulting closed-loop error equations converge, consider the extended Lyapunov candidate \( v(e, k) \triangleq \|e^T P_m e + \frac{1}{\gamma} k^T k \| \). Since
\[
\dot{v} = e^T P_m \dot{A}_m e - \frac{1}{\gamma} e^T P_m b_m k^T \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right] + \frac{1}{\gamma} k^T \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right] e^T P_m b_m
\]

it is guaranteed that \( v \), and therefore \( e, \dot{k} \), remain bounded for all time. A further technical argument based on the assumption that \( u \) is bounded for all time may be used is to finally show that \( e \) actually converges to zero as well (16).

Now apply this general method to the particular system representing the one-degree-of-freedom robot (Eq. 1) using the model resulting from the pole-placement feedback scheme,
\[
A_m \triangleq \left[ \begin{array}{cc}
0 & 1 \\
-M & -1
\end{array} \right], b_m \triangleq \left[ \begin{array}{c}
0 \\
1
\end{array} \right]
\]

In this particular context the required constraints outlined above are easily seen to hold. The “true” gain settings that make the closed-loop system behave like the model are given as \( \alpha = -M k_1, \beta = -M k_2, \gamma = M \), and the assumption that \( M > 0 \) is unexceptionable. Note that
\[
P_m \triangleq \left[ \begin{array}{cc}
k_1 & k_2 \\
\frac{1}{k_2} & 1
\end{array} \right]
\]

satisfies the conditions listed above for the Lyapunov matrix—it is positive definite (as long as \( 4k_1 > k_2^2 \)); and its product with \( A_m \) defines a negative definite symmetric matrix. Thus, the full adaptive controller takes the form

\[
u_{ad} \triangleq k^T \left[ \begin{array}{cc}
k_1 & k_2 \\
\frac{1}{k_2} & 1
\end{array} \right] \left[ \begin{array}{c}
x \\
u_{pc}
\end{array} \right]
\]
\[ u_{act} = \alpha(\theta) \dot{\theta} + \beta(\theta) \dot{\phi} + \gamma(\theta) u_{pc} \]

where
\[ \frac{d}{dt} \begin{bmatrix} \alpha(\theta) \\ \beta(\theta) \\ \gamma(\theta) \end{bmatrix} = [H(\theta - \theta_d)]k_2 + [\dot{\theta} - \dot{\theta}_d] \]

**General Robot-Arm Dynamics**

This section generalizes the discussion to the case of multijointed open kinematic chains. First, a brief but fairly general treatment of kinematics affords quick derivation of the general rigid-body model of robot-arm dynamics. In reality, this model is a simplification of empirically observed phenomena. Depending on what class of robot manipulator one studies, additional nonlinearities and dynamics cannot be ignored, and several examples of such phenomena are examined briefly.

**Rigid-Body Model: Lagrangian Formulation of Newton's Laws.** Contemporary robots are built to be rigid, and models of their idealized behavior are based on the geometry of rigid transformations. After reviewing some elementary facts leading to a useful algebraic formalism for manipulating objects that obey this "extrinsic" geometry, robot kinematics—the "intrinsic" geometry of robots—will be investigated and both geometric domains will be used to understand the dynamical properties of robot motion.

**Rigid Transformations and Frames of Reference.** For purposes of this entry, the physical world is an affine space, A^3; each element is a point described by three real numbers; however, there is no predefined origin (17). By taking differences between elements of the affine space, a, b ∈ A^3, we obtain elements of Euclidean vector space, ab ∈ E^3, with vector addition, scalar multiplication, inner product, and norm (17,18). Define a rigid transformation to be a continuous transformation of affine 3-space that preserves distance:

\[ \mathcal{G} \triangleq \{ r \in C[A^3,A^3]; \| r(a)\| = \|ab\| \text{ for all } a, b \in A^3 \} \]

Examples of rigid transformations include the vector translations,

\[ \mathcal{G} \triangleq \{ \tau_v \in \mathcal{G}; \text{ for some } v \in E^3, \tau_v(a) = a + v \text{ for all } a, b \in A^3 \} \]

and rotations around fixed points

\[ \mathcal{R} \triangleq \{ \tau_{R,o} \in \mathcal{G}; \text{ for some } R \in \mathcal{SO}(3), a \in A^3, \tau_{R,o}(o) = o, \text{ and } \tau_{R,o}(b) = Ra \text{ for all } b \in A^3 \} \]

where \( \mathcal{SO}(3) \) is the set of orthogonal linear operators on \( E^3 \) with positive determinant. In fact, it can be shown (18,19) that these examples constitute the entirety of \( \mathcal{G} \): that is, for any arbitrarily chosen "origin," \( o \in A^3 \), every rigid transformation of \( A^3 \) may be uniquely expressed as the composition of a translation with a rotation around \( a; \tau = \tau_{r,\theta} \circ \tau_{\theta,\phi} \). Thus, once a fixed point or "origin," \( a \), has been chosen, the set of rigid transformations may be put into one-to-one correspondence with the set of translations and rotations: \( \mathcal{G} = \mathcal{W} \triangleq \mathbb{R}^3 \times \mathcal{SO}(3) \). Since \[ \tau_{-1} = \tau_{-\theta} \circ \tau_{-\phi} \text{ and } \tau_{-1} = \tau_{R,\phi} \circ \tau_{R,\theta} \], it follows that \( \tau^{-1} = \tau_{R,\phi} \circ \tau_{R,\theta} \) is always defined, and hence \( \mathcal{G} \) is a group.

The position and orientation of any rigid body may be precisely described by fixing four points of \( A^3 \) called a frame of reference

\[ \mathcal{F} \triangleq \{ a, b, c, o \} \]

where \( \mathcal{F} = \{ x, y, z \} \triangleq \{ ox, oy, oz \} \) is a "right-handed" orthonormal basis of \( E^3 \). That is, the direction of \( z \) on the line orthogonal to the xy plane of \( E^3 \) obeys a right-hand rule with respect to rotations on that plane. Note that the image of a frame under a rigid transformation, \( \tau_{\mathcal{F}} = \{ \tau_{\mathcal{F}}(a), \tau_{\mathcal{F}}(b), \tau_{\mathcal{F}}(c), \tau_{\mathcal{F}}(o) \} \) is also a frame since, e.g.,

\[ \tau_{\mathcal{F}}(a) \tau_{\mathcal{F}}(b) = (oa \cdot ob) = 0 \text{ in consequence of the parallelogram rule relating norms and inner products. On the other hand, if } \mathcal{F}_1, \mathcal{F}_2 \text{ are two frames, there exists an orthogonal transformation with positive determinant, } R \in \mathcal{SO}(3) \text{ such that } x_2 = Rx_1, y_2 = Ry_1, z_2 = Rz_1 \text{ since both sets of vectors define a right-hand orthonormal basis. Thus, defining } \tau_{12} \triangleq \tau_{\mathcal{F}_1 \mathcal{F}_2} \circ \tau_{\mathcal{F}_2 \mathcal{F}_1}, \text{ there is } \tau_{12}(o) = o_2 \text{ and, e.g., } \]

\[ o_3 \tau_{12}(a_1) = o_2 \tau_{12}(a_1) - o_2 a_2 = \tau_{12}(o_1) \tau_{12}(a_1) - x_2 = o_1 \tau_{12}(a_1) - x_2 = 0 \]

so that \( \mathcal{F}_2 = \tau_{12}(\mathcal{F}_1) \). Thus, an exact description of the change in position and orientation of a given rigid body is provided equivalently by fixing a frame in the body and specifying either a rigid transformation or a second frame. Alternatively, given any two rigid bodies, an exact description of their relative position and orientation is provided by fixing a frame in each body. Assuming the existence of some "base frame," \( \mathcal{F}_o \), we now have a model of any other rigid position and orientation in terms of \( \mathcal{F}_o \triangleq \mathbb{R}^3 \times \mathcal{SO}(3) \), which we will call "work space."

It has become a tradition in robotics to use homogeneous coordinates in the representation of objects in \( A^3 \). Since rigid transformations (which, of course, are affine rather than linear) admit a matrix representation in these coordinates, the resulting simplicity in notation seems worth the slight conceptual complication. Homogeneous coordinates result in a more complicated representation from the numerical point of view as well. Thus, the matrix representation of a point \( p \in A^3 \), with respect to the any frame, \( \mathcal{F}_j \) = \{ 0, a_j, b_j, c_j \}, is denoted

\[ \pi P = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{bmatrix} \]

and is understood to represent the geometric relation

\[ \pi \dot{\pi} = \pi x_j + \pi \pi_j + \pi \pi j \]

Note that two arrays in \( R^4 \) represent the same point with respect to the same frame if they are scalar multiples. The attending conceptual complications arise because it is useful, at the same time, to employ matrix representations of vectors in \( E^3 \). Confusion between the previous arrays and these may be avoided by treating the latter as constituting "ideal points" of \( A^3 \) which, together, comprise projective space, \( P^3 \) (18). Thus, if \( v \in E^3 \).
where either $\theta_i$ or $\delta_i$ is the joint variable depending on whether the joint is revolute or prismatic, respectively, and the other kinematic parameters are defined in the link body, e.g., as in Ref. 20.

More generally, a kinematic transformation is a map

$$g : \mathcal{F} \rightarrow \mathcal{W}$$

which is the group product of $n$ joint transformations representing the rigid transformation required to align the “base frame” with the “end-effector frame”:

$$g(q) = \mathcal{F}_1(q_1) \mathcal{F}_2(q_2) \ldots \mathcal{F}_n(q_n)$$

Dynamics. The rigid-body model of robot-arm dynamics may be most quickly derived by appeal to the Lagrangian formulation of Newton’s equations. If a scalar function, termed a Lagrangian, $\lambda = \kappa - \nu$, is defined as the difference between total kinetic energy $\kappa$ and total potential energy $\nu$ in a system, the equations of motion obtain from

$$\frac{d}{dt} D_{\dot{\phi} \phi} - D_{\phi \phi} = \tau$$

where $\tau$ is a vector of external torques and forces (17, 22).

First consider the kinetic energy contributed by a small volume of mass $\delta m_i$ at position $p$ in link $\mathcal{L}_i$,

$$\delta \kappa_i = \frac{1}{2} \mathcal{F}_i^T \mathcal{F}_i \delta m_i$$

where $\mathcal{F}_i^T \mathcal{F}_i \delta m_i$ is the matrix representation of the position $p$ in the base frame of reference, $\mathcal{F}_i^T \mathcal{F}_i$ is the matrix representation of the frame of reference of link $\mathcal{L}_i$ in the base frame, and $\delta m_i$ is the matrix representation of the point in the link frame of reference, and, hence,

$$\dot{\mathcal{F}}_i = \mathcal{F}_i \dot{p}$$

since the position in the body is independent of the generalized coordinates. (We will omit the prior superscript 0 when it is clear that the coordinate system of reference is the base.) The total kinetic energy contributed by this link may now be written

$$\kappa_i = \int_{\mathcal{L}_i} \frac{1}{2} \mathcal{F}_i^T \dot{\mathcal{F}}_i \dot{p} \delta m_i$$

(since the frame matrix is constant over the integration), where $\mathcal{F}_i$ is a symmetric matrix of dynamic parameters for the link. Explicitly, if the link has mass $\mu_i$, center of gravity (in the local link coordinate system) $p_i$, and inertia matrix $J_i$,

$$\mathcal{F}_i = \left[ \begin{array}{ccc} \cos \theta_i & -\cos \alpha \sin \theta_i & -\sin \alpha \sin \theta_i & -\cos \theta_i \\ \sin \theta_i & \cos \alpha \cos \theta_i & \sin \alpha \cos \theta_i & \sin \theta_i \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Passing to the stack representation (refer to Appendix A)

$$2\kappa_i = \text{tr} \left( \mathcal{F}_i \mathcal{F}_i^T \right)$$

where $\mathcal{F}_i$ is the stack matrix.

$$\mathcal{F}_i = \left[ \begin{array}{c} J_i \\ \mu_i p_i \\ \tau \\ \mu_i \end{array} \right]$$
= (D_qF^q)\dot{q}P_i(D_qF^q)\dot{q}
= \dot{q}^T M\dot{q}

where we have implicitly defined

M_i(q) = (D_qF^q)P_i(D_qF^q)
\dot{P}_i = \dot{P}^T \otimes I

It follows that the total kinetic energy of the entire chain is given as

\kappa = \dot{q}^T M(q)\dot{q}
M(q) = \sum_{i=1}^{n} M_i(q)

The potential energy contributed by \delta m_i in \mathbb{L}_i is

\delta u_i = z^T_0 F_i \dot{p} g \delta m_i

where g is the acceleration of gravity; hence, the potential energy contributed by the entire link is

u_i = z^T_0 F_i \dot{p} g \delta m_i = z^T_0 F_i \dot{p} g

and \nu = \sum_{i=1}^{n} z^T_0 F_i \dot{p} g. Assume that \nu "points up" in a direction opposing the gravitational field.

To proceed with the computation, note that \( D_qq = D_qq = \dot{q}^T M(q) \); hence,

\frac{d}{dt} D_qq = \dot{q}^T M(q) + \dot{q}^T M(q)

Moreover,

D_qq = \dot{q}^T D_qq
= \dot{q}^T \{ q^T \otimes I \} D_qq

and hence, if all terms from Lagrange's equation involving the generalized velocity are collected, we may express them in the form \( \dot{q}^T B^T \), where

B(q, \dot{q}) = M(q) - \frac{1}{2} \dot{q}^T \{ q^T \otimes I \} D_qq

Finally, by defining \( k(q) = \dot{q}^T D_qq \), Lagrange's equation may be written in the form

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

The "inertia" matrix \( M \) may be shown to be positive definite over the entire work space as well as bounded from above since it contains only polynomials involving transcendental functions of \( \dot{q} \), and \( B \) contains terms arising from "coriolis" and "centripetal" forces and hence is linear in \( \dot{q} \) (these forces are quadratic in the generalized velocities), and bounded in \( \dot{q} \), since it involves only polynomials of transcendental functions in the generalized position. Finally, \( k \) arises from gravitational forces, is bounded, and may be observed to have much simpler structure (still polynomial in transcendental terms involving \( \dot{q} \)) than the other expressions. An important study of the form of these terms was conducted by Bejczy (23).

To bring this into the state-space form discussed above, let

\[ x = (q, \dot{q})^T, \quad u \triangleq \tau, \quad \text{to get} \]

\[ \begin{align*}
x_1 &= x_2 \\
x_2 &= -M^{-1}(x_1)(B(x_1, x_2)x_2 + k(x_1) - u)
\end{align*} \tag{12}

with output map provided by the kinematics,

\[ w = g(x) \]

Omissions in Rigid-Body Model. The theoretical presentation below will rest almost entirely for rigorous justification on the rigid-body dynamic model derived above. In fact, most commercially available robots deviate from this model quite dramatically. Flexibility in the links, slippage in transmissions, and backlash in gear trains introduce stiction, hysteresis, and other nonlinear effects. Harmonic drives and compliance at the joints introduce extra dynamics (24). Demagnetization and potential damage to the windings place limits on the maximum permissible armature current and, therefore, output torque, of a dc servo. Moreover, although it is traditional in the control community to model electric motors as if they were first-order lags (25), it is not impossible to find commercial robot arms employing dc servos whose mechanical and electrical time constants of similar magnitude, and which, in consequence, have second-order dynamics, not uncommonly oscillatory (26). Thus, not only may the model introduced in Eq. 12 have missing functional terms in practice but also its dimensionality, \( 2n \), may too low by at least again as much as the number of actuators, \( n \).

Unfortunately, it does not seem likely that a better general model will be available in the near future. There is no generally accepted understanding of how dynamic effects are significant and which may be ignored beyond the rigid-body model (Eq. 12) (which, itself, is not universally acknowledged to be of greatest importance (24)). In part, this is due to the great diversity of kinematic, actuator, and sensory arrangements that may be found on the commercial market. In part, it is a reflection of the relative novelty of the field. Similarly, there is insufficient understanding of the disturbances resulting from digital implementation of control algorithms—quantization and roundoff errors—to admit of any reliable model for these effects. Even in the control community itself researchers are only beginning to come to terms with quantization problems (27).

In this section models for some of the important dynamic and nonlinear disturbances not captured in Ref. 12 will be introduced. The relative importance of any of these discrepancies can only be determined by empirical investigation in the context of a specific mechanical apparatus. It is worth remarking here that the utility of much of the theoretical work to be presented in subsequent sections will require empirical verification for the same reasons. Clearly, in the absence of trustworthy models, proofs (and even simulations) alone are not terribly convincing.

Additional Dynamics. In the rigid-body model developed above the control input appears as a set of torques, \( \tau \) injected at each joint independently. In reality, all robotic actuators that deliver torque or force to a joint are themselves commanded by a reference voltage computed by the controller. For an extended class of actuators, the torque or force output is not a simple function of this reference voltage but may itself involve a dynamic relationship. This may be seen most easily through a specific example.

A typical actuator for a revolute joint is the dc servo. A dc motor converts electrical to mechanical power through the exchange of energy in two sets of windings via electromagnetic forces (25). For a typical motor the torque delivered to the output shaft is roughly proportional to the current in the "armature windings," \( \tau_e = K_I I_a \), and this is exactly balanced by the d'Alembert torque due the angular acceleration of motor
inertia, \( \tau_m = J\dot{\theta}_m \) as well as the external load torque, \( \tau_l \), placed on the motor (25),

\[
\tau_m + \tau_l = \tau_g
\]

On the other hand, the current in the armature winding results from the application of some command voltage, \( v_a \), through the armature resistance and inductance, \( R_a, L_a \), opposed by the back-generated voltage \( v_b = K_s\beta \),

\[
L_a \frac{dt}{dt} + R_a i_a + v_b = v_a
\]

There results the second-order linear differential equation (25)

\[
\frac{d\beta}{dt} = \nu_\beta \nu_\beta - \gamma_\nu \nu_\beta + \xi_\nu
\]

\[
\frac{d\dot{\theta}}{dt} = \nu_\beta \nu_\beta - \gamma_\beta \beta + \xi_\beta
\]

If the "electric time constant," \( \kappa_e \), is a very large negative number in comparison to the magnitude of the "mechanical time constant," \( \kappa_m \), the second equation is really algebraic,

\[
R_a i_a + v_a = v_a
\]

and the relationship between command voltage and generated torque is given by

\[
\mu \frac{dt}{dt} = \beta \dot{\beta} + \phi \nu_\beta - \tau_l
\]

\[
\nu_\beta \nu_\beta = J \beta \nu_\beta \nu_\beta = \frac{K_s K_b}{R_a} \phi \nu \nu_\beta = \frac{K_s}{R_a} \nu
\]

This is typically the case for common dc servo motors (25), although important exceptions have been noted in the literature (26).

First suppose direct-drive arm, and \( n \) such actuators are mounted directly at each joint being controlled. It follows that \( \tau_l = \tau + \hat{j} \), the load torque seen by the \( j \)th actuator, is exactly the \( j \)th component of the rigid-body external torque vector, \( \tau \), of Eq. 11. Let

\[
M_m \triangleq \begin{bmatrix} \mu_1 & 0 & \cdots & 0 \\ 0 & \mu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mu_n \end{bmatrix}; V_m \triangleq \begin{bmatrix} \nu_1 & 0 & \cdots & 0 \\ 0 & \nu_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \nu_n \end{bmatrix}; B_m \triangleq \begin{bmatrix} \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \beta_n \end{bmatrix}
\]

be the diagonal coefficient matrices corresponding to the \( n \) decoupled motors, and let \( v \triangleq [v_{a1}, \ldots, v_{an}]^T \) be the vector of command voltages at each joint. Then according to the reasoning of Dynamics,

\[
[M(q) + M_m]\dot{q} + (B(q, \dot{q}) + B_m)\dot{q} + k(q) = V_m v
\]

producing a state-space system with very similar characteristics to the original Eq. 12. It is worth noting that most of the theoretical results discussed below are still valid in this case.

Suppose, on the other hand, that an actuator is mechanically coupled to each joint via some mechanical transmission—a chain, a harmonic drive, a gear train—as is the case with most commercially available robots. In this case, \( \tau_l \), the load torque seen by the motor, is that delivered from the joint along the transmission. As a first-order approximation, the transmission will be modeled as a passive intermediate inertial load, \( \mu_i \), lumped at the joint end of the transmission on a shaft with torsional spring constant \( \kappa_i \) and torsional damping constant \( \beta_i \). Letting \( \rho_j \) denote the position of the rotor, and \( \theta_j \) the position of the joint, this assumption may be written as

\[
\tau_l = \kappa_i (\rho_j - \theta_j) + \beta_i (\dot{\rho}_j - \dot{\theta}_j)
\]

with \( \tau_l \triangleq \mu \dot{\beta}_j \) comprising the d'Alembert torque due to acceleration of the transmission inertia. The latter is balanced by the transmitted motor torque, \( \tau_m \), and the nonlinear coupling torques, \( \tau_j \); thus, the complete torque equations are

\[
\tau_g = \tau_m + \tau_j \quad \tau_l = \tau_i + \tau_j
\]

Now define the transmission diagonal arrays, \( M_i, B_i \), in terms of the inertial and viscous damping coefficients, respectively, as before. Define the motor shaft angle vector \( r \triangleq [\rho_1, \cdots, \rho_n]^T \). The torque balance equations take the vector form

\[
-B_m \dot{r} + V_m v = M_m \ddot{r} + \dot{B}(\ddot{r} - \dot{q}) + K_i (r - q)
\]

\[
B_i (\ddot{r} - \dot{q}) + K_i (r - q) = (M(q) + M_i) \ddot{q} + B(q, \dot{q}) \dot{q} + h(q)
\]

Writing this in state-space notation yields a much more complex dynamic system. Letting

\[
z = \begin{bmatrix} q \\ r \dot{q} \ddot{q} \dddot{q} \end{bmatrix} \triangleq \begin{bmatrix} z_{11} \\ z_{12} \\ z_{21} \\ z_{22} \end{bmatrix}
\]

yields

\[
\begin{array}{ccc}
z_{11} & = & z_{21} \\
\end{array}
\]

\[
\begin{array}{ccc}
z_{12} & = & z_{22} \\
\end{array}
\]

\[
\begin{array}{ccc}
z_{21} & = & -[M + M_i]^{-1} \{B + B_i\} z_{21} + K_i z_{21} - B_i z_{22} - K_i \dot{z}_{21} + h(z_{11})
\end{array}
\]

\[
\begin{array}{ccc}
z_{22} & = & -M_m^{-1} \{B_m + B_i\} z_{22} + K_i z_{22} - B_i z_{21} - K_i \dot{z}_{21} + V_m v
\end{array}
\]

**Local Nonlinearities.** As well as (at least) doubling the dimensionality of the underlying dynamics, the presence of a mechanical transmission introduces a variety of memoryless nonlinearities—backlash, hysteresis, etc.—that depend very much on the nature of the mechanism. Returning to the direct-drive arm, probably the two most significant sources of nonlinearities distinct from those due the rigid-body equations (Eq. 11) are friction and saturation.

A solid object in contact with any surface is subject to a variety of frictional forces that may be observed, in general, to vary with its velocity relative to that surface. The simplest of these is **viscous damping**, a force exerted in the opposite direction of motion in direct proportion, \( \beta_\sigma \), to the velocity magnitude. Further, it may be observed that at low speeds the magnitude of these opposing forces ceases to diminish beyond a certain level, and hence, a constant term, \( \beta_c \), called **Coulomb** friction must be added to the viscous term. Finally, the force \( \beta_\sigma \) required to bring a motionless object to some nonzero velocity typically exceeds that needed to overcome the friction forces at nonzero velocity: this is termed **stick friction**. Thus, an appropriate model for actually observed frictional forces might be given as
\[ \tau_{ctrl}(q, \dot{q}) = \beta_c + \beta_d \dot{q} + \beta_q \delta(q) \]

Recall that \( \delta \) denotes the delta function introduced earlier.

The second source of additional nonlinearities unmodeled in the previous section is a consequence of the fact that all real devices can deliver only finite power. In this light the admissible set of controls must be modified to include magnitude constraints, most easily modeled in the form of a saturation nonlinearity on the command input at the \( j \)th joint,

\[
s_j(u_j) \triangleq \begin{cases} 
\bar{s}_j & u_j \geq \bar{s}_j \\
\bar{s}_j & u_j \in (\bar{s}_j, \bar{s}_j) \\
\bar{s}_j & u_j \leq \bar{s}_j 
\end{cases}
\]

These forces act on each joint independently of the motion of the others (except, of course, through a dynamic coupling of velocities) and are termed "local disturbances" for that reason. Let \( B_C, B_w, B_d \) be diagonal matrices containing, respectively, the Coulomb, viscous, and stiffness coefficients for each joint, let \( \delta(q) = \delta(q_1) \times \ldots \times \delta(q_n) \) be the vector of delta functions in each generalized velocity, and let \( s(u) \triangleq [s_1(u_1), \ldots, s_n(u_n)] \) be the vector of saturation nonlinearities. Combining these within the vector of external disturbances, \( \tau \) in Eq. 13, yields a new version of the state-space dynamics of the form

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -[M(x_1) + M(x_1)]^{-1} [B(x_1, x_2) + B_w x_2 + B_d x_2 + P_u \delta(q) + k(x_1) + B_C - V_m s(u)]
\end{align*}
\]  

Feedback Control of General Robot Arms

Feedback Control: Behavior of Error-Driven Systems presented an account of the behavior of classical feedback controllers in the context of set-point regulation. Here, the attempt is made to generalize that account in two rather different directions. First, the breadth of task domain is considerably widened beyond set-point regulation, and the possibility is explored of formalizing a task-encoding methodology based on feedback control structures that generalize the error-driven characteristics of the PD controller. Second, the underlying dynamics of the system to be controlled are specified by a \( n \)-degree-of-freedom rigid-body model (Eq. 12) of which the simple pendulum (Eq. 1) was a particularly easy example.

Three successively more generalized methods of task encoding are presented beyond the set-point error introduced above: specification in terms of the extrema of objective functions; in terms of the "fail lines" of gradient vector fields arising from objective functions; in terms of general first-order dynamics. By interpreting an objective function as potential energy, its gradient is shown to determine a stabilizing feedback control structure for a general robot arm. The implications of this result for achieving the tasks specified by the various encoding methods are discussed.

A Generalized Robot Task-Encoding Methodology. By a "task-encoding methodology" is meant any procedure through which an abstract goal is translated into robot-control strategies resulting in its achievement. Here, we explore techniques that do not involve task specification via reference trajectory.

In part, the exploration is motivated by the difficulties involved in generalizing classical servo theory to the rigid-body dynamics (Eq. 12), as will be explored below. In part, it is motivated by the large set of task domains within which determination of an appropriate reference trajectory may involve unnecessary work, or even be effectively impossible. Typical tracking algorithms require the availability of velocity and acceleration reference information; in practice, differentiating noisy signals is impossible; such schemes are not applicable to tracking unknown time-varying signals. Moreover, evidence mounts that the computational effort required to encode typical static tasks (such as moving in a cluttered space) in terms of exact reference trajectories may be prohibitive (28,29). Finally, since many tasks involve interacting forces and motions between the robot and its environment, a task-encoding methodology limited to the production of reference trajectories may be entirely inoperable. In summary, encoding a task in dynamic terms unrelated to those characterizing the robot or its environment may not be viable.

For the purposes of this entry, an objective function, \( \varepsilon : \mathbb{W} \to \mathbb{R}^+ \) is a non-negative scalar-valued map on \( \mathbb{W} \) that has isolated critical points. Its associated gradient vector field is given by the system

\[
\dot{w} = -D_w \varepsilon(w)
\]

and the resulting trajectory through any initial condition will be called a fall line of the system. It can be shown that fall lines are perpendicular to the level surfaces of \( \varepsilon \) (30). The equilibrium states of the gradient system are exactly the critical points—i.e., the extrema—of \( \varepsilon \); and since the linearized vector field is symmetric, an equilibrium state of the gradient system is either a source, a sink, or a saddle depending on whether it is a local maximum, minimum, or saddle point of the objective function, \( \varepsilon \). Thus, gradient systems display very simple dynamic behavior. It will be shown that certain gradient behavior may be duplicated, at least asymptotically, by appropriately compensated Hamiltonian systems and, hence, that such gradients are a particularly convenient feedback structure for robot control.

Task Encoding Via Objective Functions. Evidently, tasks within the domain of set-point regulation—reaching and remaining at some desired end point, \( w_d \)—may be encoded as the objective of minimizing

\[
\varepsilon \triangleq [w - w_d]^T [w - w_d]
\]

The desired end point is the globally asymptotically stable unique equilibrium state of the associated gradient system.

Conversely, by designing a cost function with an isolated global maximum at some undesired Cartesian position, a gradient system may be constructed whose fall lines, from any initial condition different from that point, define motion away from it. This specifies the task of avoiding the undesired position. In the event that there are several obstacles in the work space, each of relatively small physical extent, cost functions attaining an isolated global maximum at the centroid of each obstacle may be summed, and the resulting vector field will specify motions that avoid all of them. A plausible form for such cost functions is the familiar Newtonian potential, which varies in the inverse square of the distance from the obstacles.
This and related methodologies have been suggested independently by workers in Japan (31), the Soviet Union (32), and the United States (33). In particular, Khatib has developed a rather general procedure for defining obstacle-avoidance potentials for arbitrary rigid bodies (33). It is not clear, however, that the computational complexity of this procedure makes it any more attractive than the algorithms developed for generating reference trajectories that solve piano-mover-type problems (29).

**Task Encoding Via Gradient Dynamics.** Task domains involving curve tracing may be specified by full lines of gradient vector fields. Suppose it is desired to reach \( w_d \) via some parameterized curve,

\[
\mathbf{c}(\xi) \equiv \begin{bmatrix} \xi \\ c_\theta(\xi) \\ c_\phi(\xi) \end{bmatrix}
\]

where \( w_d = c(0) \). Assume, to avoid technical details, that only the Cartesian position components are considered: errors in orientation may be measured similarly, although the justification requires more discussion. Then the *shaping function*

\[
e(\xi) \equiv w_d^2 + c_\theta(w_\theta - c_\theta(w_d))^2 + c_\phi(w_\phi - c_\phi(w_d))^2
\]

gives rise to a gradient system for which \( w_d \) is again the globally asymptotically stable unique equilibrium state, and whose fall lines "hug" the curve \( c \) more or less sharply depending on the magnitudes of \( a_1, a_2 \).

Fundamental work by Hogan (34) advances persuasive arguments for encoding general manipulation tasks in the form of "impedances." Impedances and admittances are formal relationships between the force exerted on the world at some Cartesian position and the motion variables—displacement, velocity, acceleration, etc.—at that position with respect to some reference point [or "virtual position" in Hogan's terminology (34)]. He argues that for purposes of modeling manipulation tasks, the kinematic and dynamic properties of a robot's contacted environment must be understood as admittances, systems for which the relationship operates as a function describing a specified displacement for any input force. Arguing further, that physical systems may only be coupled via port relationships that match admittances to impedances and that robots can violate physics no more than any other objects with mass, he arrives at the conclusion that the most general model of manipulation is the specification of an impedance, a system that returns force as a function of motion. By construing motions relative to a virtual position as defining tangent vectors at that position, Hogan notes that an impedance may be defined in terms of a scalar-valued function on the cross product of two copies of the tangent space at each virtual position whose gradient covector determines the relationship between motion and force. Thus, an impedance may be reinterpreted as the gradient covector field of an "objective function," whose fall lines specify the desired dynamic response of the robot end effector in response to infinitesimal motions imposed by the world. In this context, unlike the other gradient vector field task definitions, it is intended a priori that the dynamics be second order, i.e., define changes of velocity (force) rather than changes of position.

To conclude this brief discussion of task encoding via objective functions or their gradients, it is worth noting that the gradient is a linear operator, and hence combinations of tasks already encoded by means of objective functions are easily specified by appropriately weighted sums. For instance, if it is desired to shape an arm motion around a specified curve while simultaneously avoiding a set of known obstacles, the shaping function may be summed with the avoidance cost functions for each particular obstacle, and the gradient of the sum will preserve the local properties of each. In such cases, however, depending on the nature of the individual objective functions, their sum may define a gradient with unintended stable or partially stable critical points. Thus, globally, the fall lines of complex gradients may specify "stall" behavior at undesired equilibrium states.

**More General Feedback Structures.** It is certainly possible to imagine the desirability of "first-order dynamic behavior" more complex than can be encoded via the gradient vector field of an objective function. For instance, it has been proposed within the neurobiology community to model phenomena such as animal gait in terms of a dynamic system possessed of a stable limit cycle (35). It would seem equally attractive to use such models as task-encoding feedback structures for robot activities that require repetitive motion. Unfortunately, although "useful" gradient vector field behavior may be embedded "asymptotically" in dissipative Hamiltonian systems, as will be demonstrated in the next section, it is not clear how to do the same for more general dynamics.

A second failing of the objective function methodology is its intrinsic time-invariant character. Even if \( e(w, t) \) defines a good objective for each \( t \), it is not clear how to proceed except in the quadratic case. For example, let \( w_d(t) \) describe the trajectory of a fly the robot should swat as encoded via the objective

\[
e(w, t) \equiv |w - w_d|^2[w - w_d]
\]

Then the "gradient system"

\[
\dot{w} = -w + w_d(t)
\]

is a forced, exponentially stable, linear system, and further information concerning the nature of \( w_d \) may afford statements concerning the "steady-state error." In the special case that \( w_d \) is, itself, the output of some linear system, according to the internal model principle, a higher order linear dynamic compensator may be added to the gradient system with the assurance of asymptotic tracking. The question remains, again, as to how to transplant the internal model principle to the second-order Hamiltonian setting.

**Gradient Vector Fields and Hamiltonian Systems.** It is well known that Lagrangian mechanics may be placed within the more general framework of Hamiltonian dynamics (17). To form the Hamiltonian, we return to the scalar energy functions \( \lambda, \kappa, \) and \( \nu \) and define the *generalized momenta,*

\[
p \triangleq D_\nu^T
\]

and the *Hamiltonian*

\[
\eta \triangleq p^T \dot{q} - \lambda
\]

It is not hard to show that the Hamiltonian dynamic system
is equivalent to the rigid-body model (Eq. 12), with all external forces and torques set to zero, \( \tau = 0 \) (17). Moreover, it is easy to see that any trajectory of this system satisfies \( \eta = \eta_0 \), a constant since

\[
D\dot{\eta} = D\eta \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} = 0
\]

Finally, note that when the potential energy is free of generalized velocity, \( D\dot{\eta} = D\kappa \), and when, additionally, the kinetic energy is quadratic in velocity,

\[
p^T \dot{q} = D\kappa \dot{q} = 2\kappa
\]

and, hence, the Hamiltonian represents the total energy of the system,

\[
\eta = 2\kappa - \kappa + u = \kappa + u
\]

It should be apparent from the derivation of Dynamics that the robot energy terms satisfy these conditions.

By taking this slightly more general perspective, we are able to again use the total energy as a Lyapunov function obtaining a rather simple generalization of the stabilizing PD controller.

Stability of Dissipative Hamiltonian Systems. The central result presented in this section has been known for at least a century: Lagrange demonstrated the stability of motion around the equilibrium state of a conservative system in 1788 (36); asymptotic stability resulting from the introduction of dissipative forces to a conservative system was discussed by Lord Kelvin in 1886 (37). Over the years these ideas seem to have been rediscovered several times by different engineering communities. For instance, a similar set of observations was made in the context of satellite control in 1986 (38). Credit for first introducing these ideas to the general robotics literature would appear to be due Anil and colleagues (39). Similar independent work has appeared more recently by Van der Schaft (40) and this author (41).

It has been shown above that for a broad range of mechanical systems, including actuated kinematic chains, 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 (42). By replacing the gravitational potential term in the energy function with the objective function that defines a task and constraining the resulting total energy as a Lyapunov function for the closed-loop robot, set-point regulation may be achieved as follows. Let the input be defined as

\[
u \triangleq k(q) + x_2 + K_1(x_1 - q_d)
\]

**Theorem 1.** Let \( \mathcal{S} \) be a simply connected subset of \( \mathbb{R}^n \). The closed-loop system of Eq. 12, under the state feedback algorithm (Eq. 16),

\[
\begin{align*}
x_1 &= x_2 \\
x_2 &= -M^{-1}(B + K_2)x_2 + K_1(x_1 - q_d)
\end{align*}
\]

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

**Proof.** The Lyapunov function

\[
\psi \triangleq \frac{1}{2} [x_1^T K_1 x_1 + x_2^T M(x_1) x_2]
\]

has time derivative

\[
\dot{\psi} = x_1^T K_1 x_2 - x_2^T (B + K_2) x_2 + K_1 x_1 + \frac{1}{2} x_2^T M x_2
\]

and since \( x_2^T (B - B)x_2 = 0 \), as shown in Ref. 42, this evaluates to

\[
\dot{\psi} = -x_2^T K x_2 \leq 0
\]

According to LaSalle’s invariance principle, the attracting set is the largest invariant set contained in \( \{x_1, x_2 \in \mathcal{S} : \dot{\psi} = 0\} \), which, evidently, is the origin since the vector field is oriented away from \( x_2 = 0 \) everywhere else on that hyperplane.

Note that this control law requires the exact cancellation of any gravitational disturbance. Although \( k(q) \) 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 dynamic parameters would still be required, in general, to permit its computation. Since the dynamic parameters enter linearly in \( k \), some progress has been made in the design of “adaptive gravity cancellation” algorithms (43) as will be discussed below. A successful adaptive version of this algorithm would remove the need for any a priori information concerning the dynamic parameters.

Integrating Gradient Systems by Means of Dissipative Hamiltonian Systems. The real utility of dissipative Hamiltonian systems in the present context arises from the possibility of embedding a first-order gradient system—the task definition—in the second-order robot arm dynamics with no change in limiting behavior. Let the objective function \( \phi : \mathcal{W} \to \mathbb{R} \) be defined according to some description in task space, as described above, and let \( \tilde{e} \triangleq e \circ g \) be the composition of this objective with the kinematics map. The encoded description now takes the form of a gradient system over joint space,

\[
\dot{q} = -D\tilde{e}(q)^T \quad (17)
\]

among whose equilibrium states are desired end points and whose full lines define a desirable spatial curve or mechanical response function. The desired performance might be simulated on any analog computer with programable first-order integrators. Instead, it is appealing (and correct) to think of “solving” this gradient system on the “programable” second-order integrators defined by the intrinsic dynamics of a robot arm.

Among the extrema of \( \phi \),

\[
\mathcal{E} \triangleq \{w \in \mathcal{W} : D_w e = 0\}
\]

are the desired task space positions, \( \mathcal{E} \), the optimal points of the objective function. The task is accomplished if joint space variables tend toward the inverse image of this set.
\[ \dot{z} \in \{ q \in \mathcal{J} : g(q) \in \mathcal{P} \} \]

along the trajectories determined by the fall lines of \( \epsilon \). Now define the feedback control structure

\[ u = k(q) - K q' + D \epsilon(q) \epsilon(q)^T \]

To fix the idea, consider the simple example introduced above of end-point control defined in task space. Based on the error-minimizing objective defined in Eq. 15, the required feedback control law is

\[ u = k(q) + K q' + D \epsilon(q) \epsilon(q)^T \]

resulting in convergence from any initial position and velocity toward critical points of \( \epsilon \). Explicitly, the closed-loop system of Eq. 12, under the state feedback algorithm (Eq. 18),

\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = -M^{-1} \{ (B + K) x_1 + D x_1 \epsilon(q)^T \} \]

has a globally attracting set defined by the critical points

\[ \mathcal{G} = \{ (q, 0) \in \mathcal{P} : D \epsilon(q) \epsilon(q)^T = 0 \} \]

for any positive definite symmetric matrices \( K_1, K_2 \). This may be seen as follows. The non-negative Lyapunov function

\[ u = \dot{\epsilon}(x_1) + \frac{1}{2} x_2 M(x_1) x_2 \]

has time derivative

\[ \dot{u} = (D \epsilon \epsilon^T - x_2 [(B + K) \epsilon + D \epsilon(q)^T]) = -x_2 K x_2 \leq 0 \]

as in the proof of Theorem 1. According to LaSalle's invariance principle, the attracting set is the largest invariant set contained in \( \{ (x_1, x_2) \in \mathcal{P} : \dot{\epsilon} = 0 \} \), which, evidently, is the set of equilibria, \( \mathcal{G} \), as claimed.

This result shows that local convergence and global boundedness are assured but that a characterization of the stability properties of individual equilibrium points may be complicated. Namely, if \( \epsilon \) has critical points outside of \( \mathcal{G} \), it must be shown that these are not locally attracting equilibrium states of the closed-loop system in order to guarantee global convergence to the desired optima. Expressing the objective function gradient as \( D \epsilon = D x_1 \epsilon \) affords the equivalent formulation of the set of equilibria of the closed loop

\[ \mathcal{G} = \{ (q, 0) \in \mathcal{P} : D \epsilon \epsilon^T \in \ker D \epsilon(q)^T \} \]

This makes clear the two distinct causes of such stall points: local extrema and saddle points of the task space objective, \( \epsilon \), and critical points of the output map, \( g \), i.e., the set of kinematic singularities.

As discussed above, stall behavior due to excess critical points of \( \epsilon \) is an artifact of the task-encoding methodology. Sophisticated tasks encoded as summed gradients almost inevitably give rise to stall points in \( \mathcal{W} \). Although it is possible that a more careful construction of \( \epsilon \) from constituent objectives might mitigate the problem, this is probably an intrinsic limitation in the "global intelligence" of feedback controllers. Research addressing the interplay between higher planning levels and lower control levels should result in guidelines for the degree of supervision required to assure global convergence. For all presently available commercial robots, kinematic singularities may be found in the interior of the work space; thus, the second problem is more intrinsic to an arm and, potentially, of considerable practical concern. For a lucky choice of \( \epsilon \) it might well turn out that \( \mathcal{G} \) consists only of the points in the solution set; however, this would be unlikely. Some aspects of these questions have been addressed in a recent paper by this author (44). Of equal pragmatic importance and theoretical interest is a characterization of transient behavior obtaining from those "useful" feedback structures described above. The tasks encoded in terms of gradient dynamics are not achieved merely by asymptotic approach to an extremum of \( \epsilon \). Although the theory of linear-time-invariant controllers includes excellent analytical and graphical methods for determining the appropriate magnitude of damping in relation to position gains, as discussed above, no such theory is available for nonlinear Hamiltonian systems. However effective the methodology is in command of steady-state behavior, a gripper that oscillates wildly toward the specified end point is of no practical use. Thus, one of the most important aspects of control research in this project is the study of \( K_2 \) relative to \( K_1 \) and their nonlinear analogs in Eq. 18. More generally, it would be of great interest to know how to choose a damping function so that the motion of the second-order system projected down onto the zero-velocity plane of \( \mathcal{P} \) follows the fall lines of the original gradient system as closely as possible: i.e., what is the analogy to a critically damped linear-time-invariant system?

**Servo Control of General Robot Arms**

We return here to the paradigm of the servomechanism, wherein tasks are encoded by means of a reference trajectory. For linear-time-invariant dynamics such tracking problems comprise the central arena for the classical control theory discussed above. Unfortunately, there is as yet no generally valid means of translating the full range of these techniques into the nonlinear regime of revolute robot arms. Although the insights of that theory are often applied intuitively with some success—most commercial robots are built with decoupled linear PD servo systems at each joint—theoretical progress has been slow. Ideas that generalize the inverse dynamics approach to tracking are presented below. This approach to robot control, which assumes complete information about the reference trajectory and exact computation of the rigid body model (Eq. 12), is the only tracking paradigm for which global convergence has yet been analytically guaranteed. Then a discussion follows several tracking controllers designed for time-varying linear dynamic systems based on the substitution of its first-order (linear) approximation for the underlying nonlinear model.

It should be noted that there is currently no theoretical understanding of how to generalize the high-grain feedback techniques.

**Exact Linearization by Coordinate Transformation.** In the last decade a significant body of work has developed within the field of nonlinear systems theory concerning the question of when a specified control system has a dynamic structure that is intrinsically linear, or at least, "linearizable." More precisely, presented with a system
it would be of considerable interest to know whether there exists an invertible (memoryless) change of coordinates,

$$\dot{x} = f(x, u)$$

under which the resulting dynamics are linear time invariant,

$$\dot{z} = A z + B u$$

For this being the case, the well-understood servo techniques of classical control theory sketched above could be applied to the reference input expressed in the new coordinate system, and the resulting control, $u_d$, translated through the inverse coordinate transformation, $T^{-1}$, would result in an effective control, $u_d$, to be applied to the original system. Early discussion of this question is provided in Refs. 45 and 46, while more recent results have been presented in Ref. 47. More general discussion of this literature may be found in the recent monograph of Isidori (48) or the text by Casti (49).

It will be observed that this policy amounts to exact cancellation of the underlying dynamics of the original system. Thus, as has been remarked, such schemes represent a generalization of the method of pole placement presented above and necessitate exact knowledge of all kinematic and dynamic parameters (including those of the load); a priori knowledge of the reference trajectory; and the ability to compute exactly and implement through a set of actuators the entire dynamics in real time. Work by a number of researchers, most notably Hollerbach (60), has persuasively demonstrated that such computation is possible in real time, and computational architectures have already been designed to do so (51,52). Recent empirical results (53) suggest that this methodology may achieve good results when the requisite a priori information concerning dynamic parameters and reference trajectory is available.

**Computed-Torque Algorithm.** In the context of the robot equations (Eq. 12) these ideas lead to the technique of "computed torque," which has been proposed independently under a variety of names by several different researchers over the last five years (54–56). It seems most instructive to present the variations in the computed torque algorithm as particular examples of the following exact linearization scheme. Let $h: \mathcal{F} \rightarrow \mathbb{R}^n$ be a local diffeomorphism. Under the change of coordinates, defined by $T: \mathcal{F} \times \mathbb{R} \rightarrow \mathbb{R}^n,$

$$\begin{bmatrix} z_1 \\ z_2 \\ u \\ v \end{bmatrix} = T(x_1, x_2, u) \triangleq \begin{bmatrix} h(x_1) \\ Dx_2 - DhM^{-1}(Bx_2 + k(x_1) - u) \end{bmatrix}$$

(19)

system 12 has linear-time-invariant dynamics given by

$$\dot{z}_1 = z_2 \quad \dot{z}_2 = u$$

with output map

$$w = g \circ h^{-1}(x_1)$$

(20)

This may be seen according to the definition of $z_1, z_2$ by applying the chain rule,

$$\dot{z}_1 = Dhx_2 = z_2$$

and by noting

$$\dot{z}_2 = Dhx_2 - DhM^{-1}(Bx_2 + k - u)$$

This is not the most general class of transformations that might be used to linearize Eq. 12, according to the nonlinear systems literature cited above (e.g., Ref. 47), but it includes methods commonly encountered in the field of robotics. In particular, for nonredundant kinematics, if we identify $h$, the first component of $T$ with the kinematic map,

$$h(x_1) \triangleq g(x_1),$$

then, locally, $T$ not only linearizes Eq. 12 but also dynamically decouples each input and output pair, e.g., as reported in Refs. 54 or 56, since $w = z_1$.

As suggested above, the advantage of this approach is that the servo design problem may now be addressed by the classical methods introduced in Single-Degree-of-Freedom Robot Arm, shifting the problem of task specification to lie within the domain of some independent "higher level" process. Namely, suppose such a higher level algorithm produces a desired trajectory in work space, $w(t)$, which it is required that the robot reproduce. Defining $z_d \triangleq (h \circ g^{-1}(w_d), DhDg^{-1}w_d)$, it is quite straightforward to choose some linear feedback compensator, $u_h \triangleq Kz$, and feedforward precompensator, $u_{pp} \triangleq \Gamma z_d$, that determine a "classical" linear control law,

$$u_d \triangleq -Kz + \Gamma z_d$$

(21)

under whose action the output of Eq. 20 behaves in a desired fashion with respect to the reference input, $z_d$. The particular choice of linear control scheme determines the nature of overall performance along the lines explored. Since the relationship between $(x, u)$ and $(z, v)$ has no dynamics (is "memoryless"), the input to the robot (Eq. 12) defined by the inverse coordinate transformation for $u$ under $T$ (obtained by solving for $u$ in the last row of Eq. 19),

$$u_d \triangleq Bx_2 + k(x_1) + MDh^{-1}u_d$$

(22)

forces the output $w(t) = g(h^{-1}(z_d))$.

Perhaps the best known example of this approach in the robotics literature is provided by the "resolved acceleration" method of Ref. 55. A control law, $\Gamma z_d$, is chosen for the linearized system using the "inverse filter" method,

$$\Gamma z_d \triangleq Kz_d + \dot{z}_{dd} = K \left[ h \circ g^{-1}(w_d) \right] + \left[ \frac{d}{dt} DhDg^{-1}w_d \right]$$

i.e., the inverse of the filter specified by the equivalent closed loop linear-time-invariant system (Eq. 20). The control law $u_d$, to be applied to the robot is then given by Eq. 22. Note that this choice, $h = g$, satisfies the conditions for a local diffeomorphism almost everywhere in $\mathcal{F}$; the condition fails at the "kinematic singularities."

$$C \triangleq \{ q \in \mathcal{F}; \text{rank}(Dg)(< \text{dim } \mathbb{W}) \}$$

the critical points of $g$. Most realistic robots have kinematic singularities whose image under $g$ is in the interior of $\mathbb{W}$ in most cases that may not be easily located; hence, such a transformation may be impracticable.
As an alternative example, if the task is specified as a trajectory in joint space, \( x_d \triangleq (q_d, \dot{q}_d)^T \), this methodology corresponds to a trivial change of coordinates under the identity map, \( h = I \), and the control law (Eq. 22) reduces to
\[
u_d \triangleq Bx_d + k(x) + Mu_d
\]
The experimental results of exact linearization schemes in robotics reported to date (53) have employed this version of the algorithm.

**Robust Cancellation via Sliding Modes.** A number of researchers have explicitly addressed the problems likely to attend schemes that rely on exact cancellation, as mentioned in the introduction of this section. Here, we present an interesting approach based on the methods of “variable-structure systems.” The classification of this technique under the rubric of “robust” methods is justified in that the method requires complete knowledge only of the terms in the existing dynamics that may become unbounded.

The theory of "variable-structure systems" has been developed in the Soviet Union over the last two decades (57) and applied to the problem of robot control in recent years (58–60). For purposes of this entry, the approach may be presented in the context of objective function methods as follows. In phase space geometric relationships between state variables imply dynamic behavior, as has been seen in previous sections. Rather than specifying a goal in terms of an output objective function, the sliding mode objective specifies a desired error dynamics:
\[
\nu(x, t) \triangleq \dot{x}_2 - \dot{q}_d + \lambda(x_1 - q_d)
\]
which, if zero, implies that the position and velocity errors between the desired and true trajectory are asymptotically approaching zero. It will be noted that the attendant objective function,
\[
\nu(x, t) \triangleq s^T s
\]
is explicitly time varying; thus, the natural control strategies cannot be applied with confidence. Instead, the existing dynamics are cancelled or forced toward those desired by making systematic use of the rigid-body model (Eq. 11).

Specifically, the acceleration equations of rigid-body-model dynamics (Eq. 12) may be rewritten,
\[
\dot{z}_2 = -M^{-1}B\dot{x}_2 + k(x) - \tau = H(x)p(x) + M^{-1}\tau
\]
where \( H(x) \) is an array containing only bounded terms (constants and transcendental functions in the state) and \( p(x) \) is a vector containing in each entry a monomial, \( z_1, z_2, \ldots \), consisting of (unbounded) powers in the state variables. Although exact knowledge of the latter is required, the only information concerning the bounded terms takes the form of a superior magnitude for each entry, \( \tilde{H} \), with the property
\[
\tilde{H} > \sup_{x \in \mathbb{R}} |H(x)|
\]
Letting the control input be
\[
u \triangleq M(-H[p(x)] \text{sgn}(s) + \dot{q}_d - \lambda(x_2 - q_d))
\]
it follows that
\[
\tilde{\nu} = s^T \tilde{s} = s^T [x_2 - \dot{q}_d + \lambda(x_1 - q_d)]
\]
\[
= s^T [(H[p] \text{sgn}(s)) - (\dot{H}[p]) \text{sgn}(s)]
\]
\[
\leq s^T [\text{sgn}(s) \text{sgn}(\tilde{H} - \dot{H}[p])]
\]
\[
= -|s|^T \dot{h}^T (z) < 0
\]
where all entries in \( h^+ \) are positive so that the derivative is always negative. Implicit in these equations is the convention that the absolute value, \( \text{sgn} \), and inequality notations applied to arrays are understood to hold component by component and that the product of two vectors denotes their component-by-component (commutative) product. Thus, all trajectories originating away from the objective surface, \( \varepsilon = 0 \), tend toward it asymptotically. Since the vector field resulting from this control law is discontinuous, additional mathematical justification is required (60) in order to infer that the theoretical behavior on the surface is the desired simplified dynamics.

In practice, as has been discussed previously, limitations in accuracy and the finite response time of sensors and actuators preclude the possibility of realizing the specified control discontinuity. In consequence, physical implementations of this method give rise to chattering—rapid fluctuations of the state across the objective surface (58). Such high-frequency artifacts are extremely undesirable: excitation of (typically underdamped) unmodeled dynamics is the inevitable result. It may be noticed, moreover, that the success of the method depends on sufficiently large control magnitudes of the appropriate sign. Thus, exact knowledge of \( M \) is required to preserve sign information, \( |\nu| \), through the input coupling, and it is likely that prohibitively large input magnitudes would be specified by conservative dependence on a priori bounds, \( \tilde{H} \).

An interesting modification of this method addressing these problems has been proposed by Sastry and Slotine (59,60). They replace the discontinuous switching rule, \( |\nu| \), with a smooth version that trades ultimate tracking precision for improved transient response. They investigate, as well, the possibility of abandoning exact cancellation of the input coupling terms, \( M \), in favor of "sign preservation" through suitable diagonally dominant positive approximations.

**Other Coordinate Transformation Schemes.** Other choices for \( h \) might be imagined: if the kinematics and dynamic parameters that give rise to system 11 define a moment of inertia matrix whose square root is the Jacobian of some map, a much simpler coordinate transformation, \( T \), results. This has been explored by the author in Ref. 61. For ease of discussion, define the set of square roots of a smooth positive definite symmetric matrix valued function, \( M(q) \), as
\[
\mathcal{N}(M) \triangleq \{ N \in C^\infty(\mathbb{R}, \mathbb{R}^{n \times n}); NN^T = M \}\]
Note that since \( M \) is assumed to be positive definite, \( \mathcal{N}(M) \) is not empty, and any \( h \) satisfying the hypothesis is an immersion.

Suppose there exists a smooth map \( h: \mathcal{J} \to \mathbb{R}^n \) such that \( DH^T - N \in \mathcal{N}(M) \). Then under the change of coordinates, defined by \( T: \mathcal{J} \times \mathcal{U} \to \mathbb{R}^n \),
\[
\begin{bmatrix}
z_1 \\
z_2 \\
v
\end{bmatrix} = T(x_1, x_2, u) \triangleq \begin{bmatrix}
h(x_1) \\
n^T x_2 \\
-N^{-1}u - k(x_1)
\end{bmatrix}
\tag{23}
\]

the system has linear-time-invariant dynamics given by Eq. 20. To show this, note that \(z_2 \triangleq Dh x_2 = \dot{z}_1\). Moreover,
\[
\dot{z}_2 = Dh x_2 + Dh \dot{x}_2 = N^T x_2 - N^T [NN^T]^{-1} [B x_2 + k(x_1) - u] \quad \text{from Eq. 12}
\]
\[
= [N^T - N^{-1}B] x_2 + N^{-1} u
\]
and it remains to show that \([N^T - N^{-1}B] = 0\). To see this,
\[
B x_2 = \triangleq D x_2 - [D_x z_1]^{T} \quad \text{from Eq. 22}
\]
\[
= [NN^T + NN^T] x_2 - [D_x d z_1]^{T} z_2
\]
\[
= NN^T x_2
\]
from which the result follows. Note that the exchanged order of differentiation in the third line is justified since \(z_1\) is continuously differentiable in both \(q\) and \(t\).

Some of the advantages of a coordinate transformation based on the square root of the moment of inertia matrix are immediately evident. Given the choice of classical controller, \(u \triangleq u(t)\), from Eq. 21, the inverse transformation for \(u\) in terms of \(z_2\) is considerably simplified,
\[
u \triangleq Dh \dot{z}_2 + k(x_1)
\]
in comparison to Eq. 22. Moreover, since \(h\) is an immersion, \(T\) may be computed everywhere on \(\mathcal{X}\). The conditions for the existence of such a map, \(h\), whose Jacobian is in \(\mathcal{N}(M)\), were given by Riemann in 1854 (62) and amount to the question of whether an apparently non-Euclidean metric is “flat” — i.e., gives rise to a space of zero curvature. The transformation in question is a particular instance of a local isometry, and a more general problem of some interest concerns the existence of other isometries that simplify the dynamics of Eq. 12. Research exploring whether any useful class of robot arms gives rise to a flat metric and whether more general isometries might be helpful for purposes of control continues (61).

**Linear Approximations.** Since the equations of motion of a robot are analytic in the state variables, it follows that all partial derivatives, \(A(x, u) \triangleq \frac{\partial}{\partial x} f; A_x, u \triangleq \frac{\partial}{\partial u} f; \) exist and are analytic as well. Suppose that \(x(t), x'(t)\) are two motions governed by Eq. 12, resulting from the application of the control inputs \(u(t), u'(t)\), respectively, starting from the same initial conditions. If their differences, \(z \triangleq x' - x; \dot{u} \triangleq u' - u\), are small throughout the time interval of interest, according to Taylor’s formula, \(f(x', u') = f(x, u) + A(x, u) z + B(x, u) \dot{u}\). Since \(\dot{x} = f(x', u') - f(x, u)\), it follows that the evolution of \(\dot{x}\) is governed approximately by \(\dot{u}\) according to the dynamic law
\[
\dot{x} = A(x(t), u(t)) \hat{x} + B(x(t), u(t)) \hat{u}
\]
This is a time-varying linear differential equation, and it might be anticipated that some of the insights explored above carry over to such a system more readily than to the actual robot equations (Eq. 12). In point of fact, while a formal expression for the I/O properties of such a system may be readily exhibited, its analytical properties (much less its explicit computation) may not be any better understood than the origin nonlinear system from which it has been derived. However, since these functions are all smooth, the linear coefficients approximately constant over sufficiently small intervals of time; if control action is possible on such rapid time scales, then insights of the section on single-degree-of-freedom robot arm may be of some use.

**Gain Scheduling.** Suppose a control \(u\) has been devised in such a fashion that the resulting trajectory \(x_d\) exhibited desired properties. For example, \(u\) might be chosen accorded the ideal computed torque method discussed above; it might instead be the result of some iterative off-line optimization procedure. The relevant point is that off-line numerical simulation of system 12 for specified values of \(u\), while computationally costly, is a straightforward procedure. In turn, if availability of \(u_d(t)\), \(x_d(t)\) affords off-line computation of the Jacobian matrices \(A, B\) along that motion. According to the reasoning stated above, if \(u_d\) is applied to the true system, at any instant of time \(t_0\) small deviations \(z\) around \(x\) are governed by the instantaneous linear-time-invariant dynamics
\[
\dot{z} = A_0 z + B_0 \hat{u}
\]
where \(A_0 \triangleq A(x_d(t_0), u_d(t_0)), B_0 \triangleq B(x_d(t_0), u_d(t_0)).\) Over an interval around \(t_0\) small enough to admit these approximations then, a linear-time-invariant feedback controller can be constructed with gains set according to the well-understood design principles sketched in the beginning of this entry. At some later time \(t_1\) a different set of gains to achieve the same purpose will be required to compensate for the dynamic caused by \(A_1, B_1\). The technique of assigning a different set of linear-time-invariant compensator gains to different regime of system behavior is widely known as “gain scheduling.”

The chief objection to this method is that stable compensation is guaranteed only in some local neighborhood of the nominal trajectory. If excessive disturbances are present—whether due to inaccuracies in the model, i.e., its computation or as a result of unexpected contingencies in the environment—it is possible that tracking will fail by the true system response might even become unbounded (i.e., saturate). Another practical cause for degraded performance is a consequence of the continuous evolution of the Jacobian coefficients over the time interval that the compensator is assigned a set of constant gains according to schedule. This might be ameliorated by increasing the scheduling rate, but it is possible that the time variation of the linearized model, governed by the rate at which the reference signal changes as well as the sensitivity of the linearized model to changing nominal operating points, could prove simply too fast for the controller.

The technique of local linearization is frequently encountered in the robotics control literature, and it is worth noting a few representative examples. A study of control issues arising in the context of the linearized dynamic model around specific fixed positions in the workspace is given in Ref. 63. A computer-automated environment for designing controllers based on linearization around the “computed torque” trajectory is described in Ref. 64. An optimization method is described in
Ref. 65 for solving obstacle-avoidance problems which returns a numerically computed nominal input–response pair that might be used as the basis for a local linearization feedback controller.

**Local Adaptive and Learning Techniques.** If the local behavior of a nonlinear system is essentially governed by shifting linear-time-invariant dynamics in the sense explored in the preceding section, it seems fruitful to inquire regarding the existence of a single compensator that would suffice to control all the disparate linear-time-invariant systems over that range. Alternatively, if a linear compensator exists that adequately controls the locally linearized plant at every instant of time along some trajectory, it might be possible to build a time-varying compensator that adjusts its parameters accordingly—an “automatic gain scheduler.” The first of these ideas leads to the notion of robust nonlinear controllers. This topic properly deserves a separate section here. Unfortunately, although most commercial robots are built using the linear “PD” techniques, there is very little understanding of how these ideas should generalize to the nonlinear setting, and, accordingly, no material to fill such a section here.

The second idea leads to the notion of adaptive control, which is treated here in its “local” form. As shown in the previous section, the nonlinear dynamics of the robot arm at any point in the work space are locally characterized by an instantaneous linear-time-invariant vector field. If a standard adaptive algorithm converges quickly enough, it is plausible that adaptive estimates of the current instantaneous linearized coefficients will result in an effective control. Thus, theoretical questions introduced by such algorithms reduce to the study of the sort of schemes presented in Adaptive Control. Implementation of these ideas has been considered, e.g., in Refs. 66–68. An interesting hybrid on-line off-line closed-loop local adaptive strategy for robot arms has been proposed in recent years and is worth independent mention.

The original proponent of such a scheme, which uses a sequence of open-loop control inputs to force the output to track a desired signal, seems to have been Arimoto (69). The error between the true and desired response resulting from a previous control input is used to modify that signal and produce the next control input in the sequence. The entire time history of the error function resulting from a particular iterate is used in the construction of the next. Explicitly, suppose a control input, \( u(t) \), has been chosen to track the desired output, \( x_d(t) \), starting from initial conditions \( x^* \) with resulting velocity errors

\[
e(t) = \dot{q}_d(t) - \dot{x}_d(t)
\]

The simplest version of the scheme calls for the next control candidate to be defined by

\[
u_{i+1}(t) = u_i(t) + \Phi \epsilon_i(t)
\]

and applied from the same initial condition, \( x^* \). When the underlying plant dynamics are linear, Arimoto shows that the error sequence does indeed converge uniformly for arbitrary initial conditions \( x^* \in \Phi \). The same convergence result may be applied locally—i.e., by assuming no matter how bad the intermediate behavior, the state variables remain within some compact domain in \( \Phi \)—to the nonlinear robot dynamics of Eq. 11.

**Global Adaptive Controllers.** It has been mentioned above that a rigorous theory of adaptive control for linear-time-invariant systems is a relatively recent development. Accordingly, the prospects for theoretically sound adaptation algorithms for general nonlinear systems seems quite remote in the near future. Fortunately, in contrast, the possibility of developing practicable globally stable adaptive robot control algorithms within the next few years seems quite bright. This optimism is grounded on the following observations.

The rigid-body model (Eq. 11) is highly nonlinear in the state \( \mathbf{x} \) and kinematic parameters but linear in the dynamic parameters (as will be verified shortly). Future robots will probably be built using the direct-drive technology (70). This implies that the omissions in rigid-body model take the form presented in Additional Dynamics which is still linear in (an augmented set of) the dynamic parameters in contrast to the additional nonlinearities presented in Local Nonlinearities. Adaptive problems with linear parameter dependence are much more tractable than general nonlinear problems, as will be seen below.

To see that the system of Eq. 11 is linear in the dynamic parameters, note that

\[
M(q) = \sum_{i=1}^{n} (\mathbf{\hat{H}}(q))^T \mathbf{P}_i \mathbf{\hat{H}}(q)
\]

where \( \mathbf{\hat{H}}(q) \) depends entirely on joint positions and kinematic parameters, and

\[
\mathbf{P}_i = \begin{bmatrix} J_i^T \mu_i \mu_i^T \end{bmatrix} \otimes \mathbf{I}
\]

depends on the dynamic parameters. Since

\[
((\mathbf{\hat{H}}(q))^T \mathbf{\hat{H}}(q))^S = ((\mathbf{\hat{H}}(q))^T \otimes (\mathbf{\hat{H}}(q))^T)^S
\]

(refer to Appendix A for material concerning the “stack representation”), defining

\[
\mathbf{\hat{H}}(q) = \begin{bmatrix} ((\mathbf{\hat{H}}_1(q))^T \otimes (\mathbf{\hat{H}}_1(q))^T) & \cdots & 0 \\
0 & \ddots & 0 \\
0 & \cdots & ((\mathbf{\hat{H}}_n(q))^T \otimes (\mathbf{\hat{H}}_n(q))^T)
\end{bmatrix}
\]

\[
\mathbf{p} = \begin{bmatrix} \mathbf{p}_1^T \\
\vdots \\
\mathbf{p}_n^T
\end{bmatrix}
\]

yields \( M(q)^S = H(q)p \). Since \( M \) is linear in \( p \), its derivatives must be as well, and hence, \( B(q, q) \dot{q} = H(q)p \). It is clear from the derivation in Dynamics that \( k \) is linear in \( p \). \( k(q) \otimes H^T(q)p \).

**Adaptive Computed Torque.** Now consider the general problem of adaptive control for the robot serve problem. Suppose a desired trajectory, \( q_d \), is given, along with a linear precompensating scheme, \( u_{pc} \), which makes \( x_{m1} \), the output of a forced model reference dynamic system,

\[
x_{m1} = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix} x_{m1} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} u_{pc}
\]

track \( q_d \) in an acceptable fashion. Assume, moreover, that the structure of the robot dynamics \( H, H^T, H^T \) is entirely known although the dynamic parameters \( p \) are not.

If \( p \) were known, the appropriate control strategy would be a generalization of the pole placement scheme,
\( u_d \triangleq k(q) + B(q, \dot{q})\dot{q} + M(q) - K_1q - K_2\dot{q} + u_{pe} \)
\[
= H^*(q)p + (H'p + (A(q, \dot{q}) + K_2\dot{q} + u_{pe}F \otimes I)Hp
\]
\[
= \hat{H}(q, \dot{q}, u_{pe})p,
\]
where this "linearizes" the robot dynamics in the sense of exact linearization by coordinate transformation and places the poles such that the closed-loop system has the dynamics of the reference model. Thus a generalization of the reasoning in Adaptive Control suggests that the appropriate adaptive control take the form
\[
u_d \triangleq \hat{H}\dot{\hat{p}} = u_d + \hat{H}(\hat{p} - p)
\]
where \( \hat{p} \), the parameter estimate, will be continuously adjusted over the course of the robot's motion. The closed-loop error equations, \( e \triangleq x_m - x \), under this control take the form
\[
\dot{e} = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix} e + \begin{bmatrix} 0 \\ I \end{bmatrix} (M^{-1}\hat{H}(\hat{p} - p))
\]
Again generalizing from the earlier section on adaptive control, the adaptive law should depend on a Lyapunov function for the reference plant. A convenient choice is \( v \triangleq x_m^TP_m x_m \), where
\[
P_m \triangleq \begin{bmatrix} K_1 & -K_1 \\ I & I \end{bmatrix}
\]
may be shown to be positive definite as long as the reference system is chosen such that \( MK_1 - K_2 \) is positive definite. Note that \( \dot{v} = -x_m^TP_m \dot{x}_m \) along the motion of the reference system, and this is easily seen to be negative definite under the further assumption that \( K_1, K_2 \) commute. Given these assumptions, the choice of adaptive law constant should be
\[
\dot{\hat{p}} = \hat{H}^T M^{-1}(K_2\dot{e}_1 + \varepsilon_2)
\]
Unfortunately, this law is entirely impracticable since by involving \( M \) explicitly, it presupposes the availability of the very information that necessitated an adaptive approach in the first place.

It is equally appealing to consider instead the adaptive law,
\[
\dot{\hat{p}} = \hat{H}^T \hat{M}^{-1}(K_2\dot{e}_1 + \varepsilon_2)
\]
where \( \hat{M} \) is computed according to the recipe for \( M \) using the current value of the parameter estimate, \( \hat{H} \). Unfortunately, while \( \hat{M} \) is known to be positive definite and, therefore, invertible over all \( q \in \mathcal{J} \), the estimate at any given instant, \( \hat{M} \), does not enjoy such a guarantee. Even if this condition could be assured, it is no longer obvious how to demonstrate convergence of the overall scheme.

Interesting recent work by Craig, Hsu, and Sastry, (71) presents an approach to this problem based on the inverse dynamics precompensator,
\[
u_{pe} \triangleq \dot{q}_d + K_2\dot{q}_d + K_1q_d
\]
Their analysis examines a "reverse causal" precompensator-robot forward path system—i.e., the reference model driven with inverse dynamics involving the true robot position and velocity. This leads to error equations of the form
\[
\dot{e} = \begin{bmatrix} 0 & I \\ -K_1 & -K_2 \end{bmatrix} e + \begin{bmatrix} 0 \\ I \end{bmatrix} (\hat{M}^{-1}\hat{H}^*(\hat{p} - p))
\]
and an adaptive law of the form
\[
\dot{\hat{p}} = \hat{H}^T \hat{M}^{-1}(K_2\dot{e}_1 + \varepsilon_2)
\]
where \( \hat{M} \) is prevented from becoming singular or unbounded by explicitly arresting the adaptation when \( \hat{p} \) leaves a predetermined compact region in the positive orthant of parameter space. Convergence obtains after a finite number of "adaptation resets." Unfortunately, as \( \hat{H} \) contains reference trajectory acceleration terms, so does \( \hat{H}^* \)—a portion of the adaptive law to be synthesized on-line—contain true response acceleration terms \( \dot{q} \) that would require either use of accelerometers or instantaneous differentiation of real-time signals in any practical implementation.

Adaptive Gravity Cancellation for a PD Controller. Since the complete paradigm of adaptive control does not easily translate into the nonlinear robotic setting, it is sensible to consider schemes where adaptation plays a reduced role. Assume that the desired task is achieved by a natural control law of the kind examined above, and consider the problem of adaptive cancellation of the gravity term mentioned in Stability of Dissipative Hamiltonian Systems.

Specifically, it is required to cancel the destabilizing portion of the vector field,
\[
k(q) = H\dot{\hat{p}}
\]
The appropriate control input is given as
\[
u_{at} \triangleq -H_{\gamma\rho} - K_2x_2 - K_1x_1
\]
with \( x_1 \triangleq q_\gamma - q \), and \( \hat{\rho} \), the present estimate of the unknown gains that will be adjusted continuously during the robot's motion. According to the earlier discussion of adaptation, the construction of the adaptive law requires use of an extended Lyapunov function. Unfortunately, the only presently available candidate is the total energy function,
\[
v \triangleq \frac{1}{2} x_1^T M x_2 + \frac{1}{2} x_1^T K_1 x_1
\]
whose time derivative along trajectories of the (perfectly gravitationally canceled) closed-loop system was shown to be negative semidefinite rather than negative definite. Proceeding anyway, in analogy to that discussion, set the adaptive law to be
\[
\dot{\hat{p}} = -H^T x_2
\]
Notice that this is a practicable procedure since all explicit dependence on \( p \) is canceled. The closed-loop behavior is the governed by the equation
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -M^{-1}(B + K_2)x_2 + K_1(x_1) + H\dot{\hat{p}}
\end{align*}
\]
It is shown in Ref. 43 that this system has a stable origin as it gives rise to bounded solutions whose limit set is contained in the subspace
\[
\mathcal{P} \triangleq \left\{ (x_1) : x_2 = 0 \right\}
\]
thus, each physical trajectory will converge to some spati position \( q_0 \in \mathcal{J} \), and the parameter estimate \( \hat{\rho} \) will converge
some constant $\beta_0 \in \mathbb{R}^{12n}$. Unfortunately, the result says nothing about the relation of these constants to their desired values. In fact, the most likely result of this procedure would be entirely unsatisfactory. For all those positions $q_0 \in \mathcal{F}$ at which $H(q_0)$ has full rank, the origin of the system of Eq. 24 lies in the interior of a smooth submanifold of $\mathcal{L}$ specified by

$$
\mathcal{M} \ni \begin{bmatrix} q \\ 0 \\ \dot{p} \end{bmatrix} \in H^{-1}K(q_0 - q)
$$

which is a set of equilibrium states. Thus, not only is the origin non-attracting but also solutions will converge to constants in $\mathcal{M}$ however distant from the origin that manifold extends. Physically, this corresponds to a command torque based on a spatial error whose corruption by the parameter error exactly balances the gravitational force vector at a particular point in $\mathcal{W}$. Research attempting to improve this result is currently in progress.

Appendix: Some Notation

If $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 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 $x = [x_1, x_2]$, and we desire the Jacobian of $f$ with respect to the variables $x_1 \in \mathbb{R}^m$, as $x_2$ is held fixed, we may write

$$
D_{x_1} f \equiv Df[I_1 \times n_1 \quad 0]
$$

If $A \in \mathbb{R}^{m \times m}$, the "stack" representation (72) of $A \in \mathbb{R}^{m \times m}$ formed by stacking each column below the previous will be denoted $A^S$. If $C \in \mathbb{R}^{p \times q}$, and $A$ is as above, the Kronecker product of $A$ and $C$ is

$$
A \otimes C = \begin{bmatrix}
a_{11}C & \cdots & a_{1m}C \\
\vdots & \ddots & \vdots \\
a_{n1}C & \cdots & a_{nm}C
\end{bmatrix} \in \mathbb{R}^{mp \times nq}
$$

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

$$
(ABC)^S = (CT \otimes A)B^S
$$

A second useful identity between matrices and their stack representation is

$$
\text{tr}(DK^T) = (D^S)^T L^S
$$

BIBLIOGRAPHY


The spread of robotic technology will change the structure of both manufacturing and service organizations in the United States during the next decade. In the manufacturing sector, product design, process planning, materials handling, and management activities, both individually and in combination, will be affected. In service organizations the introduction of robotics will improve the quality of service and reduce the cost of providing such service. The implementation of robotic technology will alter the job market because of the increased use of complex equipment in manufacturing. The activity of the operator will change from manual labor, such as loading parts in a machine or manually guiding spray paint equipment, to activities related to the control of segments in the manufacturing process. The operator must know the manufacturing process as well as the operational aspects of the computerized robotic equipment. Organizations providing educational services will be affected since the basic education provided by public and private elementary and secondary schools must be upgraded. Increasing the math, physics, and computer programming abilities of those entering the work force is essential. Any school system that fails to respond will quickly lose public support as their students will be unable to compete for jobs and undertake advanced training provided by higher education. At the same time, a means of retraining the existing work force must be developed; it has not yet been established whether this activity will be delegated to the colleges or to private commercial organizations.

The introduction of robotics requires an improvement in the quality of materials and parts supplied to the final manufacturing system. The cost of defective or unacceptable parts is extremely high as they can be very disruptive to robotic assembly units. Management is directly affected; robotic manufacturing requires a concentration of production facilities. Entire new management control and merchandise distribution systems will have to be devised and implemented.

BACKGROUND

Robotic technology has been entering different facets of the economy for the past two decades in a fragmented approach, with the result that few people recognize its total potential impact. This has been an era in which the economy has been robust, manpower has been relatively abundant, low-cost computational power has been in an embryonic state, and individuals with the knowledge and ability to interface the computer to equipment have been very scarce.

The 1964 New York World’s Fair exhibits included major efforts to apply this technology to entertainment and service. Walt Disney introduced robotic technology in the form of character animation and computer control of the Small World exhibit. The cost, although massive, was relatively low for the entertainment field, and it was possible to utilize a large commercial computer system to achieve the artistic effects.

Because the exhibits were highly independent, the temporary failure of different components was not catastrophic. In addition, the public was not cognizant of the desired movement of each element—the total effect was the product of music, color, and action. Each area in the exhibit was a new experience and was highly action packed, carrying the audience to some far-off land. Thus, the viewers were not sensitive to any simple malfunctions. The Disney organization continued to apply this technology to the world of entertainment with little, if any, significant transfer of the technology to the industrial manufacturing sector applications of the economy.

The basic automation technology was understood in the 1960s, but the necessary low-cost hardware and an abundant supply of real-time programmers did not exist. During this period, the progress that was being made in the fields of solid-state electronics, integrated circuits, and very large scale integration (VLSI) was driven by the needs of the space and military activities. In the early 1970s, the industry had progressed to the point where there was a transfer of the military/spaceship technology to the general market, and reliable 4-bit integrated circuits (microprocessors) became commercially available. The development of microprocessors in the 1970s was of major importance. In 1974, the application of 4-bit units began, followed by 8-bit units in 1976, and then 16-bit units in 1980. This provided the industry with machines having the ability to adapt to changing conditions by a simple modification of the program controlling the unit; these machines were called “robots.” They were limited in that they did not sense the output and were passive or open-loop.

The application of simple, passive robots to improve productivity and thus help retain manufacturing in the United States became a national goal in the early 1980s. Once this basic technology was mastered, a need was recognized for intelligent robots, or machines that were able to sense and adapt to changing conditions. Parallel to the expansion of robots in manufacturing systems was a dramatic improvement in cameras and the publication of academic textbooks (1-4). Hence, machine vision became a viable candidate for achieving the required sensing capability for “intelligent” robots and provided a bonus by way of appropriate input data to the computer-aided manufacturing (CAM) systems.

Machine vision technology is well within the grasp of undergraduate students, and applications are at the cutting edge of engineering. Machine vision is destined to find a place in most industrial plants by the early 1990s, for robotic control, quality assurance, and CAM data input. The applications span a large spectrum of complexity, from simple feature and mea-