17

Robust and Adaptive Motion Control of Manipulators∗

17.1 Introduction

This chapter concerns the robust and adaptive motion control of robotic manipulators. The goals of both robust and adaptive control are to maintain performance in terms of stability, tracking error, or other specifications, despite parametric uncertainty, external disturbances, unmodeled dynamics, or other uncertainties present in the system. In distinguishing between robust control and adaptive control, we follow the commonly accepted notion that a robust controller is usually a fixed controller, static or dynamic, designed to satisfy performance specifications over a given range of uncertainties, whereas an adaptive controller incorporates some sort of online parameter estimation. This distinction is important. For example, in a repetitive motion task the tracking errors produced by a fixed robust controller would tend to be repetitive as well, whereas tracking errors produced by an adaptive controller might be expected to decrease over time as the plant and/or control parameters are updated based on runtime information. At the same time, adaptive controllers that perform well in the face of parametric uncertainty may not perform well in the face of other types of uncertainty such as external disturbances or unmodeled dynamics. An understanding of the trade-offs involved is therefore important in deciding whether to employ robust or adaptive control design methods in a given situation.

∗Partial support from the National Science Foundation under grant ECS-0122412 is gratefully acknowledged.
Many of the fundamental theoretical problems in motion control of robot manipulators were solved during an intense period of research from about the mid-1980s until the early 1990s during which researchers first began to exploit fundamental structural properties of manipulator dynamics, such as feedback linearizability, passivity, multiple time-scale behavior, and other properties that we discuss below.

Within the scope of the present article, it is impossible to survey the entire range of results in manipulator control. We will focus on analytical design methods, including Lyapunov based design, variable structure control, operator theoretic methods, and passivity-based control for rigid robots. We will briefly touch upon more recent hybrid control methods based on multiple-models but will not discuss control design based on so-called soft computing methods, such as neural networks, fuzzy logic, genetic algorithms, or statistical learning methods. We will also not discuss issues involved in force feedback control, friction compensation, or control of elastic manipulators. The references at the end of this article may be consulted for additional information on these and other topics.

17.2 Background

Robot manipulators are basically multi-degree-of-freedom positioning devices. Robot dynamics are multi-input/multi-output, highly coupled, and nonlinear. The main challenges in the motion control problem are the complexity of the dynamics resulting from nonlinearity and multiple degrees-of-freedom, and uncertainty, both parametric and dynamic. Parametric uncertainty arises from imprecise knowledge of inertia parameters, while dynamic uncertainty arises from joint and link flexibility, actuator dynamics, friction, sensor noise, and unknown environment dynamics.

We consider a robot manipulator with \textbf{n-links} interconnected by \textbf{joints} into a \textbf{kinematic chain}. Figure 17.1 shows a serial link (left) and a parallel link (right) manipulator. A parallel robot, by definition,
Robust and Adaptive Motion Control of Manipulators

contains two or more independent serial link chains. For simplicity of exposition we shall confine our discussion to serial link manipulators with only rotational or revolute joints as shown in Figure 17.2. Most of the discussion in this article remains valid for parallel robots and for robots with sliding or prismatic joints.

We define the joint variables, \( q_1, \ldots, q_n \), as the relative angles between the links, for example, \( q_i \) is the angle between link \( i \) and link \( i - 1 \). A vector \( q = (q_1, \ldots, q_n)^T \) with each \( q_i \in (0, 2\pi) \), is called a configuration. The set of all possible configurations is called configuration space or joint space, which we denote as \( C \). The configuration space for a revolute joint robot is an \( n \)-dimensional torus, \( T^n = S^1 \times \cdots \times S^1 \), where \( S^1 \) is the unit circle.

The task space is the space of all positions and orientations (called poses) of the end-effector. We attach a coordinate frame, called the base frame, or world frame, at the base of the robot and a second frame, called the end-effector frame or task frame, at the end-effector. The end-effector position can then be described by a vector \( x \in \mathbb{R}^3 \) specifying the coordinates of the origin of the task frame in the base frame. The end-effector orientation can be described by a \( 3 \times 3 \) matrix \( R \in SO(3) \). The task space is then isomorphic to the special Euclidean group, \( SE(3) = \mathbb{R}^3 \times SO(3) \), elements of which are called rigid motions \([41]\).

17.2.1 Kinematics

The forward kinematics map is a function

\[
X_0 = \begin{pmatrix} x(q) \\ R(q) \end{pmatrix} = f_0(q) : T^n \rightarrow SE(3)
\]

from configuration space to task space which gives the end-effector pose in terms of the joint configuration. The inverse kinematics map gives the joint configuration as a function of the end-effector pose. The
forward kinematics map is many-to-one, so that several joint space configurations may give rise to the same end-effector pose. This means that the forward kinematics always has a unique pose for each configuration, while the inverse kinematics has multiple solutions, in general.

The kinematics problem is compounded by the difficulty of parametrizing the rotation group, SO(3). It is well known that there does not exist a minimal set of coordinates to “cover” SO(3), i.e., a single set of three variables to represent all orientations in SO(3) uniquely. The most common representations used are Euler angles and quaternions. Representational singularities, which are points at which the representation fails to be unique, give rise to a number of computational difficulties in motion planning and control.

Given a minimal representation for SO(3), for example, a set of Euler angles $\phi, \theta, \psi$, the forward kinematics map may also be defined by a function

$$ X_1 = \begin{pmatrix} x(q) \\ o(q) \end{pmatrix} = f_1(\cdot) : \mathbb{T}^n \rightarrow \mathbb{R}^6 $$

(17.2)

where $x(q) \in \mathbb{R}^3$ gives the Cartesian position of the end-effector and $o(q) = (\phi(q), \theta(q), \psi(q))^T$ represents the orientation of the end-effector. The nonuniqueness of the inverse kinematics in this case will include multiplicities due to the particular representation of SO(3) in addition to multiplicities intrinsic to the geometric structure of the manipulator.

Velocity kinematics refers to the relationship between the joint velocities and the end-effector velocities. If the mapping $f_0$ from Equation (17.1) is used to represent the forward kinematics, then the velocities are given by

$$ V = \begin{pmatrix} v \\ \omega \end{pmatrix} = J_0(q) \dot{q} $$

(17.3)

where $J_0(q)$ is a $6 \times n$ matrix, called the manipulator Jacobian. The vectors $v$ and $\omega$ represent the linear and angular velocity, respectively, of the end-effector. The linear velocity $v \in \mathbb{R}^3$ is just $\frac{d}{dt} x(q)$, where $x(q)$ is the end-effector position vector from Equation (17.1). It is a little more difficult to see how the angular velocity vector $\omega$ is computed since the end-effector orientation in Equation (17.1) is specified by a matrix $R \in$ SO(3). If $\omega = (\omega_x, \omega_y, \omega_z)^T$ is a vector in $\mathbb{R}^3$, we may define a skew-symmetric matrix, $S(\omega)$, according to

$$ S(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} $$

(17.4)

The set of all skew-symmetric matrices is denoted by $so(3)$. Now, if $R(t)$ belongs to SO(3) for all $t$, it can be shown that

$$ \dot{R} = S(\omega(t)) R $$

(17.5)

for a unique vector $\omega(t)$ [41]. The vector $\omega(t)$ thus defined is the angular velocity of the end-effector frame relative to the base frame.

If the mapping $f_1$ is used to represent the forward kinematics, then the velocity kinematics is written as

$$ \dot{X}_1 = J_1(q) \dot{q} $$

(17.6)

where $J_1(q) = \frac{\partial f_1}{\partial q}$ is the $6 \times n$ Jacobian of the function $f_1$. In the sequel we will use $f$ to denote either the matrix $f_0$ or $f_1$. 

Copyright © 2005 by CRC Press LLC
17.2.2 Dynamics

The dynamics of \( n \)-link manipulators are conveniently described via the Lagrangian approach. In this approach, the joint variables, \( q = (q_1, \ldots, q_n)^T \), serve as generalized coordinates. The manipulator kinetic energy is given by a symmetric, positive definite quadratic form,

\[
K = \frac{1}{2} \sum_{i,j=1}^{n} d_{ij}(q) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{q}^T D(q) \dot{q}
\]  

(17.7)

where \( D(q) \) is the inertia matrix. The manipulator potential energy is given by a continuously differentiable function \( P: \mathcal{C} \to \mathbb{R} \). For a rigid robot, the potential energy is due to gravity alone while for a flexible robot the potential energy will also contain elastic potential energy.

The dynamics of the manipulator are then described by Lagrange's equations [41]

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = \tau_k, \quad k = 1, \ldots, n
\]  

(17.8)

where \( L = K - P \) is the Lagrangian and \( \tau_1, \ldots, \tau_n \) represent input generalized forces. In local coordinates Lagrange's equations can be written as

\[
\sum_{j=1}^{n} d_{kj}(q) \ddot{q}_j + \sum_{i,j=1}^{n} \Gamma_{ijk}(q) \dot{q}_i \dot{q}_j + \phi_k(q) = \tau_k, \quad k = 1, \ldots, n
\]  

(17.9)

where

\[
\Gamma_{ijk} = \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\}
\]  

(17.10)

are known as Christoffel symbols of the first kind, and

\[
\phi_k = \frac{\partial P}{\partial q_k}
\]  

(17.11)

In matrix form we can write Lagrange's Equation (17.9) as

\[
D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau
\]  

(17.12)

In addition to the link inertias represented by the inertia matrix, \( D(q) \), the inertias of the actuators are important to include in the dynamic description, especially for manipulators with large gear reduction. The actuator inertias are specified by an \( n \times n \) diagonal matrix

\[
I = \text{diag}(I_1 r_1^2, \ldots, I_n r_n^2)
\]  

(17.13)

where \( I_i \) and \( r_i \) are the actuator inertia and gear ratio, respectively, of the \( i \)th joint. Defining \( M(q) = D(q) + I \), we may modify the dynamics to include these additional terms as

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = \tau
\]  

(17.14)

17.2.3 Properties of Rigid Robot Dynamics

The equations of motion (17.14) possess a number of important properties that facilitate analysis and control system design. Among these are
1. The inertia matrix $M(q)$ is symmetric and positive definite, and there exist scalar functions, $\mu_1(q)$ and $\mu_2(q)$, such that

$$\mu_1(q) ||\xi|| \leq \xi^T M(q) \xi \leq \mu_2(q) ||\xi||$$

for all $\xi \in \mathbb{R}^n$ where $||\cdot||$ denotes the usual Euclidean norm in $\mathbb{R}^n$. Moreover, if all joints are revolute then $\mu_1$ and $\mu_2$ are constants.

2. For suitable definition of $C(q, \dot{q})$, the matrix $W(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric. (See [41] for the appropriate definition of $C(q, \dot{q})$ and the proof of the skew symmetry property.)

3. The mapping $\tau \rightarrow \dot{q}$ is passive, i.e., there exists $\beta \geq 0$ such that

$$\int_0^T \dot{q}^T(\zeta) \tau(\zeta) d\zeta \geq -\beta, \quad \forall \ T > 0$$

To show the passivity property, let $H$ be the total energy of the system

$$H = \frac{1}{2} \dot{q}^T M(q) \dot{q} + P(q)$$

Then it is easily shown using the skew-symmetry property that the change in energy, $\dot{H}$, satisfies

$$\dot{H} = \dot{q}^T \tau$$

Integrating both sides of Equation (17.18) with respect to time gives

$$\int_0^T \dot{q}^T(u) \tau(u) du = H(T) - H(0) \geq -H(0)$$

since the total energy $H(T)$ is nonnegative. Passivity then follows with $\beta = H(0)$.

4. Rigid robot manipulators are fully actuated, i.e., there is an independent control input for each degree-of-freedom. By contrast, robots possessing joint or link flexibility are no longer fully actuated and the control problems are more difficult, in general.

5. The equations of motion (17.12) are linear in the inertia parameters. In other words, there is a constant vector $\theta \in \mathbb{R}^p$ and a function $Y(q, \dot{q}, \ddot{q}) \in \mathbb{R}^{n \times p}$ such that

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + g(q) = Y(q, \dot{q}, \ddot{q}) \theta = \tau$$

The function $Y(q, \dot{q}, \ddot{q})$ is called the regressor. The parameter vector $\theta$ is composed of link masses, moments of inertia, and the like, in various combinations. Historically, the appearance of the passivity and linear parametrization properties in the early 1980s marked watershed events in robotics research. Using these properties researchers have been able to prove elegant global convergence and stability results for robust and adaptive control. We will detail some of these results below.

### 17.3 Taxonomy of Control Design Methods

The remainder of the article surveys several of the most important control design methods for rigid robots beginning with the notion of computed torque or feedback linearization and ending with hybrid control. For reasons of space, we give only the simplest versions of these ideas applied to rigid robots. We will assume that the full state $q(t), \dot{q}(t)$ of the robot is available whenever needed. Several of the results presented here, particularly the passivity based control results, have been extended to the case of output feedback, i.e., to the case that only the position $q(t)$ is directly measured. In this case, either an observer or some sort of velocity...
filter must be incorporated into the control design. We will not discuss the inclusion of actuator dynamics or joint or link flexibility into the control design. We will also not discuss the basic PD and PID control methods, which can be considered as limiting cases of robust control when no nominal model is used. PD and PID control is useful for the regulation problem, i.e., for point-to-point motion, especially when used in conjunction with gravity compensation but is less suitable for tracking time varying trajectories. The reader is referred to the references at the end of this article for further reading in these subjects.

17.3.1 Control Architecture

The motion control problem for robots is generally hierarchically decomposed into three stages, Motion Planning, Trajectory Generation, and Trajectory Tracking as shown in Figure 17.3. In the motion planning stage, desired paths are generated in the task space, \( SE(3) \), without timing information, i.e., without specifying velocity or acceleration along the paths. Of primary concern is the generation of collision free paths in the workspace. The motion planner may generate a plan offline using background knowledge of the robot and environment, or it may incorporate task level sensing (e.g., vision or force) and modify the motion plan in real-time.

In the trajectory generation stage, the desired position, velocity, and acceleration of the manipulator along the path as a function of time or as a function of arclength along the path are computed. The trajectory planner may parametrize the end-effector path directly in task space, either as a curve in \( SE(3) \) or as a curve in \( \mathbb{R}^6 \) using a particular minimal representation for \( SO(3) \), or it may compute a trajectory for the individual joints of the manipulator as a curve in the configuration space \( C \).

In order to compute a joint space trajectory, the given end-effector path must be transformed into a joint space path via the inverse kinematics mapping. A standard approach is to compute a discrete set of joint vectors along the end-effector path and to perform an interpolation in joint space among these points in order to complete the joint space trajectory. Common approaches to trajectory interpolation include polynomial spline interpolation, using trapezoidal velocity trajectories or cubic polynomial trajectories, as well as trajectories generated by reference models. The trajectory generator should take into account realistic velocity and acceleration constraints of the manipulator.

The computed reference trajectory is then presented to the controller, which is designed to cause the robot to track the given trajectory as closely as possible. The joint level controller will, in general, utilize both joint level sensors (e.g., position encoders, tachometers, torque sensors) and task level sensors (e.g., vision, force). This article is mainly concerned with the design of the tracking controller assuming that the path and trajectory have been precomputed.

17.3.2 Feedback Linearization Control

The goal of feedback linearization is to transform a given nonlinear system into a linear system by use of a nonlinear coordinate transformation and nonlinear feedback. The roots of feedback linearization in

---

**FIGURE 17.3** Block diagram of the robot control problem.
robotics predate the general theoretical development by nearly a decade, going back to the early notion of feedforward computed torque [29].

In the robotics context, feedback linearization is also known as inverse dynamics or computed torque control [41]. The idea is to exactly compensate all of the coupling nonlinearities in the Lagrangian dynamics in a first stage so that a second stage compensator may be designed based on a linear and decoupled plant. Any number of techniques may be used in the second stage. The feedback linearization may be accomplished with respect to the joint space coordinates or with respect to the task space coordinates. Feedback linearization may also be used as a basis for force control, such as hybrid position/force control and impedance control.

### 17.3.3 Joint Space Inverse Dynamics

Given the plant model

\[
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{17.21}
\]

a joint space inverse dynamics control is given by

\[
\tau = M(q)a_q + C(q, \dot{q})\dot{q} + g(q) \tag{17.22}
\]

where \(a_q \in \mathbb{R}^n\) is, as yet, undetermined. Because the inertia matrix \(M(q)\) is invertible for all \(q\), the closed loop system reduces to the decoupled double integrator system

\[
\ddot{q} = a_q \tag{17.23}
\]

The term \(a_q\), which has units of acceleration, is now the control input to the double integrator system. We shall refer to \(a_q\) as the outer loop control and \(\tau\) in Equation (17.22) as the inner loop control. This inner loop/outer loop architecture, shown in Figure 17.4, is important for several reasons. The structure of the inner loop control is fixed by Lagrange’s equations. What control engineers traditionally think of as control system design is contained primarily in the outer loop, and one has complete freedom to modify the outer loop control to achieve various goals without the need to modify the dedicated inner loop control. In particular, additional compensation, linear or nonlinear, may be included in the outer loop to enhance the robustness to parametric uncertainty, unmodeled dynamics, and external disturbances. The outer loop control may also be modified to achieve other goals such as tracking of task space trajectories instead of joint space trajectories, regulating both motion and force. The inner loop/outer loop architecture thus unifies many robot control strategies from the literature.

---

**FIGURE 17.4** Inner loop/outer loop architecture.
Let us assume that the joint space reference trajectory, \( q^d(t) \), is at least twice continuously differentiable and that \( q^d(t) \) along with its higher derivatives \( \dot{q}^d(t) \) and \( \ddot{q}^d(t) \) are bounded. Set \( a_q = \ddot{q}^d + v \) and define

\[
e(t) = \begin{bmatrix} \ddot{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} q(t) - q^d(t) \\ \dot{q}(t) - \dot{q}^d(t) \end{bmatrix}
\]
as the joint position and velocity tracking errors. Then we can write the error equation in state space as

\[
\dot{e} = Ae + Bv
\]

where

\[
A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}
\]

A general linear control for the system (17.24) may take the form

\[
\ddot{z} = F\ddot{z} + Ge
\]

\[
v = Hz + Ke
\]

where \( F, G, H, K \) are matrices of appropriate dimensions. The block diagram of the resulting closed loop system is shown in Figure 17.5 with

\[
G(s) = (sI - A)^{-1}B
\]

\[
C(s) = (sI - F)^{-1}G + K
\]

An important special case arises when \( F = G = H = 0 \). In this case the outer loop control, \( a_q \), is a static state feedback control with gain matrix \( K = [K_p, K_d] \) and feedforward acceleration \( \ddot{q}^d \),

\[
a_q = \ddot{q}^d(t) + v = \ddot{q}^d(t) + K_p(q^d - q) + K_d(\dot{q}^d - \dot{q})
\]

Substituting (17.28) into (17.23) yields the linear and decoupled closed loop system

\[
\ddot{q} + K_d\dot{q} + K_pq = 0
\]

Taking \( K_p \) and \( K_d \) as positive definite diagonal matrices, we see that the joint tracking errors are decoupled and converge exponentially to zero as \( t \to \infty \). Indeed, one can arbitrarily specify the closed loop natural frequencies and damping ratios. We will investigate other choices for the outer loop term \( a_q \) below.

17.3.4 Task Space Inverse Dynamics

As an illustration of the importance of the inner loop/outer loop paradigm, we will show that tracking in task space can be achieved by modifying our choice of outer loop control \( a_q \) in (17.23) while leaving the inner
loop control unchanged. Let \( X \in \mathbb{R}^6 \) represent the end-effector pose using any minimal representation of \( SO(3) \). Since \( X \) is a function of the joint variables \( q \in \mathcal{C} \), we have

\[
\dot{X} = J(q)\dot{q}
\]

(17.30)

\[
\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q}
\]

(17.31)

where \( J = J_1 \) is the Jacobian defined in Section 17.2.1. Given the double integrator system, (17.23), in joint space we see that if \( a_q \) is chosen as

\[
a_q = J^{-1}(a_X - \dot{J}\dot{q})
\]

(17.32)

the result is a double integrator system in task space coordinates

\[
\ddot{X} = a_X
\]

(17.33)

Given a task space trajectory \( X^d(t) \), satisfying the same smoothness and boundedness assumptions as the joint space trajectory \( q^d(t) \), we may choose \( a_X \) as

\[
a_X = \ddot{X}^d + K_p(X^d - X) + K_d(\dot{X}^d - \dot{X})
\]

(17.34)

so that the Cartesian space tracking error, \( \dddot{X} = X - X^d \), satisfies

\[
\dddot{X} + K_d\ddot{X} + K_p\dot{X} = 0
\]

(17.35)

Therefore, a modification of the outer loop control achieves a linear and decoupled system directly in the task space coordinates, without the need to compute a joint trajectory and without the need to modify the nonlinear inner loop control.

Note that we have used a minimal representation for the orientation of the end-effector in order to specify a trajectory \( X \in \mathbb{R}^6 \). In general, if the end-effector coordinates are given in \( SE(3) \), then the Jacobian \( J \) in the above formulation will be the Jacobian \( J_0 \) defined in Section 17.1. In this case

\[
V = \begin{pmatrix} \nu \\ \omega \end{pmatrix} = \begin{pmatrix} \dot{X} \\ \dot{\omega} \end{pmatrix} = J(q)\dot{q}
\]

(17.36)

and the outer loop control

\[
a_q = J^{-1}(q) \begin{pmatrix} \dot{a_X} \\ \dot{a_\omega} \end{pmatrix} - \dot{J}(q)\dot{q}
\]

(17.37)

applied to (17.23) results in the system

\[
\ddot{x} = a_x \in \mathbb{R}^3
\]

(17.38)

\[
\dot{\omega} = a_\omega \in \mathbb{R}^3
\]

(17.39)

\[
\dot{R} = S(\omega)R, \ R \in SO(3), \ S \in so(3)
\]

(17.40)

Although, in this latter case, the dynamics have not been linearized to a double integrator, the outer loop terms \( a_x \) and \( a_\omega \) may still be used to achieve global tracking of end-effector trajectories in \( SE(3) \).

In both cases we see that nonsingularity of the Jacobian is necessary to implement the outer loop control. If the robot has more or fewer than six joints, then the Jacobians are not square. In this case, other schemes have been developed using, for example, the pseudoinverse in place of the inverse of the Jacobian. See [10] for details.

The inverse dynamics control approach has been proposed in a number of different guises, such as resolved acceleration control [26] and operational space control [21]. These seemingly distinct approaches have all been shown to be equivalent and may be incorporated into the general framework shown above [24].
17.3.5 Robust Feedback Linearization

The feedback linearization approach relies on exact cancellation of nonlinearities in the robot equations of motion. Its practical implementation requires consideration of various sources of uncertainties such as modeling errors, unknown loads, and computation errors. Let us return to the Euler-Lagrange equations of motion

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (17.41)$$

and write the control input $\tau$ as

$$\tau = \hat{M}(q)(\ddot{q}^d + v) + \hat{C}(q, \dot{q})\dot{q} + \hat{g}(q) \quad (17.42)$$

where $\hat{\cdot}$ represents the computed or nominal value of $\cdot$ and indicates that the theoretically exact feedback linearization cannot be achieved in practice due to the uncertainties in the system. The error or mismatch $\tilde{\cdot} = \cdot - \hat{\cdot}$ is a measure of one's knowledge of the system dynamics.

If we now substitute (17.42) into (17.41) we obtain, after some algebra,

$$\ddot{q} = \ddot{q}^d + v + \eta(q, \dot{q}, \ddot{q}^d, v) \quad (17.43)$$

where

$$\eta = (M^{-1}\ddot{M} - I)(\ddot{q}^d + v) + M^{-1}(\ddot{q} + \ddot{g}) \quad (17.44)$$

and

$$\eta = E\nu + y \quad (17.45)$$

where

$$E = (M^{-1}\ddot{M} - I)$$

and

$$y = F(e, t) = E\ddot{q}^d + M^{-1}(\ddot{q} + \ddot{g})$$

The error equations are now written as

$$\dot{e} = Ae + B[v + \eta] \quad (17.46)$$

where $e, A, B$ are defined as before. The block diagram representation of the system must now be modified to include the uncertainty $\eta$ as shown in Figure 17.6. The robust control design problem is now to design the compensator $C(s)$ in Figure 17.6 to guarantee stability and tracking while rejecting the disturbance $\eta$.

We will detail three primary approaches that have been used to tackle this problem.

17.3.5.1 Stable Factorizations

The method of stable factorizations was first applied to the robust feedback linearization problem in [39, 40]. In this approach, the so-called Youla-parametrization [48] is used to generate the entire class, $\Omega$, of

![FIGURE 17.6 Uncertain double integrator system.](image)
stabilizing compensators for the unperturbed system, i.e., Equation (17.46) with \( \eta = 0 \). Given bounds on the uncertainty, the Small Gain Theorem [17] is used to generate a sufficient condition for stability of the perturbed system, and the design problem is to determine a particular compensator, \( C(s) \), from the class of stabilizing compensators \( \Omega \) that satisfies this sufficient condition.

The interesting feature of this problem is that the perturbation terms appearing in (17.46) are so-called persistent disturbances, i.e., they are bounded but do not vanish at \( t \to \infty \). This is chiefly due to the properties of the gravity terms and the reference trajectories. As a result, one may assume that the uncertainty \( \eta \) is finite in the \( L_\infty \)-norm but not necessarily in the \( L_2 \)-norm since under some mild assumptions, an \( L_2 \) signal converges to zero as \( t \to \infty \).

The problem of stabilizing a linear system while minimizing the response to an \( L_\infty \)-bounded disturbance is equivalent to minimizing the \( L_1 \) norm of the impulse response of the transfer function from the disturbance to the output [30]. For this reason, the problem is now referred to as the \( L_1 \)-optimal control problem [30, 44]. The results in [39, 40] predate the general theory and, in fact, provided early motivation for the more general theoretical development first reported in [44]. We sketch the basic idea of this approach below. See [5, 40] for more details.

We first require some modelling assumptions in order to bound the uncertainty term \( \eta \). We assume that there exists a positive constant \( \alpha < 1 \) such that

\[
||E|| = ||M^{-1} \dot{M} - I|| \leq \alpha \quad (17.47)
\]

We note that there always exists a choice for \( \dot{M} \) satisfying this assumption, for example \( \dot{M} = \mu_1 + \mu_2 I \), where \( \mu_i \) are the bounds on the inertia matrix in Equation (17.15) [40]. From this and the properties of the manipulator dynamics, specifically the quadratic in velocity form of the Coriolis and centrifugal terms, we may assume that there exist positive constants \( \delta_1, \delta_2, \) and \( b \) such that

\[
||\eta|| \leq \alpha ||v|| + \delta_1 ||e|| + \delta_2 ||e||^2 + b \quad (17.48)
\]

Let \( \delta \) be a positive constant such that

\[
\delta_1 ||e|| + \delta_2 ||e||^2 \leq \delta ||e|| \quad (17.49)
\]

so that

\[
||\eta|| \leq \alpha ||v|| + \delta ||e|| + b \quad (17.50)
\]

We note that this assumption restricts the set of allowable initial conditions as shown in Figure 17.7. With this assumption we are restricted to so-called semiglobal, rather than global, stabilization. However, the region of attraction can, in theory, be made arbitrarily large. For any region in error space, \( |e| \leq R \), we may take \( \delta \geq \delta_1 + \delta_2 R \) in order to satisfy Equation (17.49).

Next, from Figure 17.6 it is straightforward to compute

\[
e = G(I - CG)^{-1} \eta =: P_1 \eta \quad (17.51)
\]

\[
v = CG(I - CG)^{-1} \eta =: P_2 \eta \quad (17.52)
\]

The above equations employ the common convention of using \( P \eta \) to mean \( (p * \eta)(t) \) where \( * \) denotes the convolution operator and \( p(t) \) is the impulse response of \( P(s) \). Thus

\[
||e|| \leq \beta_1 ||\eta|| \quad (17.53)
\]

\[
||v|| \leq \beta_2 ||\eta|| \quad (17.54)
\]

where \( \beta_i \) denotes the operator norm of the transfer function, i.e.,

\[
\beta_i = \sup_{x \in \mathbb{L}_\infty^{-\{0\}}} \frac{||P_i x||_{\infty}}{||x||_{\infty}} \quad (17.55)
\]
As the quantities $e, v, \eta$ are functions of time, we can calculate a bound on the uncertainty as follows

\[
\|\eta\|_{\infty} \leq \alpha \|v\|_{\infty} + \delta \|e\|_{\infty} + b \\
\leq (\alpha \beta_2 + \delta \beta_1) \|\eta\|_{\infty} + b
\]

(17.56)

(17.57)

where the norm denotes the truncated $L_\infty$ norm [17] and we have suppressed the explicit dependence on time of the various signals. Thus, if

\[
\alpha \beta_2 + \delta \beta_1 < 1
\]

(17.58)

the uncertainty and hence both the control signal and tracking error are bounded as

\[
\|\eta\|_{\infty} \leq \frac{b}{1 - (\alpha \beta_2 + \delta \beta_1)}
\]

(17.59)

\[
\|e\|_{\infty} \leq \frac{\beta_1 b}{1 - (\alpha \beta_2 + \delta \beta_1)}
\]

(17.60)

\[
\|v\|_{\infty} \leq \frac{\beta_2 b}{1 - (\alpha \beta_2 + \delta \beta_1)}
\]

(17.61)

Condition (17.58) is a special case of the small gain theorem [17, 31]. In [40] the stable factorization approach is used to design a compensator $C(s)$ to make the operator norms $\beta_1$ and $\beta_2$ arbitrarily close to zero and one, respectively. It then follows from Equation (17.60), letting $t \to \infty$, that the tracking error can be made arbitrarily small in norm.

Specifically, the set of all controllers that stabilize $G(s)$ is given by

\[
\{- (Y - \dot{N})^{-1}(X + \dot{D})\}
\]

(17.62)

where $N, D, \dot{N}, \dot{D}$ are stable proper rational transfer functions satisfying

\[
G(s) = N(s)[D(s)]^{-1} = [\dot{D}(s)]^{-1} \dot{N}(s)
\]

(17.63)
$X, Y$ are stable rational transfer functions found from the Bezout Identity

$$
\begin{bmatrix}
Y(s) & X(s) \\
-\bar{N}(s) & \bar{D}(s)
\end{bmatrix}
\begin{bmatrix}
D(s) & -\bar{X}(s) \\
N(s) & \bar{Y}(s)
\end{bmatrix} = I
$$

(17.64)

and $R(s)$ is a free parameter, an arbitrary matrix of appropriate dimensions whose elements are stable rational functions. Defining

$$
C_k = \{-(Y - R_k \bar{N})^{-1}(X + R_k \bar{D})\}
$$

(17.65)

a constructive procedure in [40] gives a sequence of matrices, $R_k$, such that the corresponding gains $\beta_1$ and $\beta_2$ converge to zero and one, respectively, as $k \to \infty$.

### 17.3.5.2 Lyapunov’s Second Method

A second approach to the robust control problem is the so-called theory of guaranteed stability of uncertain systems, which is based on Lyapunov’s second method and sometimes called the Corless-Leitmann [13] approach. In this approach the outer loop control, $a_q$ is a static state feedback control rather than a dynamic compensator as in the previous section. We set

$$
a_q = \ddot{q}^d(t) + K_p(\dot{q}^d - \dot{q}) + K_d(\dot{q} - \dot{\tilde{q}}) + \delta a
$$

(17.66)

which results in the system

$$
\dot{e} = Ae + B(\delta a + \eta)
$$

(17.67)

where

$$
A = \begin{bmatrix}
0 & I \\
-K_p & -K_d
\end{bmatrix}, \quad B = \begin{bmatrix}
0 \\
I
\end{bmatrix}
$$

(17.68)

Thus the double integrator is first stabilized by the linear feedback, $-K_p e - K_d \dot{e}$, and $\delta a$ is an additional control input that must be designed to overcome the destabilizing effect of the uncertainty $\eta$. The basic idea is to compute a time-varying scalar bound, $\rho(e, t) \geq 0$, on the uncertainty $\eta$, i.e.,

$$
||\eta|| \leq \rho(e, t)
$$

(17.69)

and design the additional input term $\delta a$ to guarantee ultimate boundedness of the state trajectory $x(t)$ in Equation (17.67).

Returning to our expression for the uncertainty

$$
\eta = E a_q + M^{-1}(\ddot{q}\dot{\tilde{q}} + \dot{\tilde{g}})
$$

$$
= E \delta a + E(\ddot{q}^d - K_p \dot{\tilde{q}} - K_d \dot{\tilde{q}}) + M^{-1}(\ddot{\tilde{q}} + \dot{\tilde{g}})
$$

(17.70)

(17.71)

we may compute

$$
||\eta|| \leq \alpha||\delta a|| + \gamma_1||e|| + \gamma_2||e||^2 + \gamma_3
$$

(17.72)

where $\alpha < 1$ as before and $\gamma_1$ are suitable nonnegative constants. Assuming for the moment that $||\delta a|| \leq \rho(e, t)$, which must then be checked a posteriori, we have

$$
||\eta|| \leq \alpha \rho(e, t) + \gamma_1||e|| + \gamma_2||e||^2 + \gamma_3 =: \rho(e, t)
$$

(17.73)

which defines $\rho$ as

$$
\rho(e, t) = \frac{1}{1 - \alpha}(\gamma_1||e|| + \gamma_2||e||^2 + \gamma_3)
$$

(17.74)
Since $K_p$ and $K_d$ are chosen so that $A$ in Equation (17.67) is a Hurwitz matrix, we choose $Q > 0$ and let $P > 0$ be the unique symmetric matrix satisfying the Lyapunov equation,

$$A^T P + PA = -Q$$  \hfill (17.75)

Defining the control $\delta a$ according to

$$
\delta a = \begin{cases} 
-\rho(e,t) \frac{B^T P e}{\|B^T P e\|}, & \text{if } \|B^T P e\| \neq 0 \\
0, & \text{if } \|B^T P e\| = 0
\end{cases}
$$  \hfill (17.76)

it follows that the Lyapunov function $V = e^T P e$ satisfies $\dot{V} \leq 0$ along solution trajectories of the system (17.67). To show this result, we compute

$$\dot{V} = -e^T Q e + 2e^T PB \{\delta a + \eta\}$$  \hfill (17.77)

For simplicity, set $w = B^T P e$ and consider the second term, $w^T [\delta a + \eta]$ in the above expression. If $w = 0$ this term vanishes and for $w \neq 0$, we have

$$\delta a = -\rho \frac{w}{\|w\|}$$  \hfill (17.78)

and Equation (17.77) becomes, using the Cauchy-Schwartz inequality,

$$w^T \left( -\rho \frac{w}{\|w\|} + \eta \right) \leq -\rho \|w\| \|w\| \|\eta\|$$  \hfill (17.79)

$$= \|w\| (-\rho + \|\eta\|) \leq 0$$  \hfill (17.80)

and hence

$$\dot{V} < -e^T Q e$$  \hfill (17.81)

and the result follows. Note that $\|\delta a\| \leq \rho$ as required.

Since the above control term $\delta a$ is discontinuous on the manifold defined by $B^T P e = 0$, solution trajectories on this manifold are only defined in a generalized sense, the so-called Filippov solutions [18], and the control signal exhibits chattering. One may implement a continuous approximation to the discontinuous control as

$$\delta a = \begin{cases} 
-\rho(e,t) \frac{B^T P e}{\|B^T P e\|}, & \text{if } \|B^T P e\| > \epsilon \\
-\frac{\rho(e,t)}{\epsilon} B^T P e, & \text{if } \|B^T P e\| \leq \epsilon
\end{cases}$$  \hfill (17.82)

and show that the tracking error is now uniformly ultimately bounded with the size of the ultimate bound being a function of $\epsilon$ [41].

### 17.3.5.3 Sliding Modes

The sliding mode theory of variable structure systems has been extensively applied to the design of $\delta a$ in Equation (17.46). This approach is very similar in spirit to the Corless-Leitmann approach above. In the sliding mode approach, we define a sliding surface in error space

$$s := \ddot{q} + \Lambda \ddot{q} = 0$$  \hfill (17.83)

where $\Lambda$ is a diagonal matrix of positive elements. Return to the uncertain error equation

$$\ddot{\delta a} + K_d \dot{\delta a} + K_p \delta a = \delta a + \eta$$  \hfill (17.84)
the goal is to design the control $\delta a$ so that the trajectory converges to the sliding surface and remains constrained to the surface. The constraint $s(t) = 0$ then implies that the tracking error satisfies

$$\dot{\tilde{q}} = -\Lambda \tilde{q}$$

(17.85)

If we let $K_p = K_d \Lambda$, we may write Equation (17.84) as

$$\dot{\tilde{q}} = -K_d (\dot{\tilde{q}} + \Lambda \tilde{q}) + \delta a + \eta = -K_d s + \delta a + \eta$$

(17.86)

Define

$$V = \frac{1}{2} s^T s$$

(17.87)

and compute

$$\dot{V} = s^T \dot{s} = s^T (\dot{\tilde{q}} + \Lambda \dot{\tilde{q}})$$

(17.88)

$$= s^T (-K_d s + \delta a + \eta + \Lambda \dot{\tilde{q}})$$

(17.89)

$$= -s^T K_d s + s^T (\delta a + \eta + \Lambda \dot{\tilde{q}})$$

(17.90)

$$= -s^T K_d s + s^T (v + \eta)$$

(17.91)

where $\delta a$ has been chosen as $\delta a = -\Lambda \dot{\tilde{q}} + v$

Examining the term $s^T (v + \eta)$ in the above, we choose the $i$th component of $v$ as

$$v_i = \rho_i (e, t) \text{sgn}(s_i), \quad i = 1, \ldots, n$$

(17.92)

where $\rho_i$ is a bound on the $i$th component of $\eta$, $s_i = \tilde{q}_i + \lambda_i \tilde{q}_i$ is the $i$th component of $s$, and $\text{sgn}(\cdot)$ is the signum function

$$\text{sgn}(s_i) = \begin{cases} +1 & \text{if } s_i > 0 \\ -1 & \text{if } s_i < 0 \end{cases}$$

(17.93)

Then

$$s^T (v + \eta) = \sum_{i=1}^{n} (s_i \eta_i - \rho_i \text{sgn}(s_i))$$

(17.94)

$$\leq |s_i|(|\eta| - \rho_i) \leq 0$$

(17.95)

It therefore follows that

$$\dot{V} = -s^T K_d s < 0$$

(17.96)

and $s \to 0$. The discontinuous control input $v_i$ switches sign on the sliding surface (Figure 17.8). In the ideal case of infinitely fast switching, once the trajectory hits the sliding surface, it is constrained to lie on the sliding surface and the closed loop dynamics are thus given by Equation (17.85), and hence the tracking error is globally exponentially convergent. Similar problems arise in this approach, as in the Corless-Leitmann approach, with respect to existence of solutions of the state equations and chattering of the control signals. Smoothing of the discontinuous control results in removal of chattering at the expense of tracking precision, i.e., the tracking error is then once again ultimately bounded rather than asymptotically convergent to zero [37].

**17.3.6 Adaptive Feedback Linearization**

Once the linear parametrization property for manipulators became widely known in the mid-1980s, the first globally convergent adaptive control results began to appear. These first results were based on the inverse dynamics or feedback linearization approach discussed above. Consider the plant (17.41) and
control (17.42) as above, but now suppose that the parameters appearing in Equation (17.42) are not fixed as in the robust control approach but are time-varying estimates of the true parameters. Substituting Equation (17.42) into Equation (17.41) and setting

\[ a_q = \ddot{q}^d + K_d(\dot{q}^d - \dot{q}) + K_p(q^d - q) \]  

(17.97)

it can be shown, using the linear parametrization property, that

\[ \ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q} = \dot{\hat{M}}^{-1}Y(q, \dot{q}, \ddot{q})\dot{\hat{\theta}} \]  

(17.98)

where \( Y \) is the regressor function, and \( \tilde{\theta} = \hat{\theta} - \theta \), and \( \hat{\theta} \) is the estimate of the parameter vector \( \theta \). In state space we write the system (17.98) as

\[ \dot{e} = Ae + B\Phi\tilde{\theta} \]  

(17.99)

where

\[
A = \begin{bmatrix} 0 & I \\ -K_p & -K_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad \Phi = \dot{\hat{M}}^{-1}Y(q, \dot{q}, \ddot{q})
\]  

(17.100)

with \( K_p \) and \( K_d \) chosen so that \( A \) is a Hurwitz matrix. Let \( P \) be that unique symmetric, positive definite matrix \( P \) satisfying

\[ A^T P + PA = -Q \]  

(17.101)

and choose the parameter update law as

\[ \dot{\hat{\theta}} = -\Gamma^{-1}\Phi^T B^T Pe \]  

(17.102)

where \( \Gamma = \Gamma^T > 0 \). Then global convergence to zero of the tracking error with all internal signals remaining bounded can be shown using the Lyapunov function

\[ V = e^T Pe + \frac{1}{2}\dot{\hat{\theta}}^T\Gamma\dot{\hat{\theta}} \]  

(17.103)

Calculating \( \dot{V} \) yields

\[ \dot{V} = -e^T Qe + \dot{\hat{\theta}}^T\{\Phi^T B^T Pe + \Gamma\dot{\hat{\theta}}\} \]  

(17.104)
the latter term following since \( \theta \) is constant, i.e., \( \dot{\theta} = \hat{\theta} \). Using the parameter update law (17.102) gives

\[
\dot{V} = -e^T Q e
\]  

(17.105)

From this it follows that the position tracking errors converge to zero asymptotically and the parameter estimation errors remain bounded. Furthermore, it can be shown that the estimated parameters converge to the true parameters provided the reference trajectory satisfies the condition of **persistence of excitation**, \[
\alpha I \leq \int_{t_0}^{t_0+T} Y^T(q^d, \dot{q}^d, \ddot{q}^d) Y(q^d, \dot{q}^d, \ddot{q}^d) dt \leq \beta I
\]  

(17.106)

for all \( t_0 \), where \( \alpha, \beta, \) and \( T \) are positive constants.

In order to implement this adaptive feedback linearization scheme, however, one notes that the acceleration \( \ddot{q} \) is needed in the parameter update law and that \( \hat{M} \) must be guaranteed to be invertible, possibly by the use of projection in the parameter space. Later work was devoted to overcome these two drawbacks to this scheme by using so-called **indirect** approaches based on a (filtered) prediction error.

### 17.3.7 Passivity-Based Approaches

One may also exploit the passivity of the rigid robot dynamics to derive robust and adaptive control algorithms for manipulators. These methods are qualitatively different from the previous methods which were based on feedback linearization. In the passivity-based approach, we modify the inner loop control as

\[
\tau = \dot{M}(q) \dot{a} + \hat{C}(q, \dot{q}) v + \dot{\hat{g}}(q) - K r
\]  

(17.107)

where \( v, a, \) and \( r \) are given as

\[
v = \dot{q}^d - \Lambda \ddot{q}
\]

\[
a = \dot{v} = \dot{\dot{q}}^d - \Lambda \dddot{q}
\]

\[
r = \ddot{q}^d - v = \dddot{q} + \Lambda \dddot{q}
\]

with \( K, \Lambda \) diagonal matrices of positive gains. In terms of the linear parametrization of the robot dynamics, the control (17.107) becomes

\[
\tau = Y(q, \dot{q}, \dot{a}, v) \dot{\hat{\theta}} - K r
\]  

(17.108)

and the combination of Equation (17.107) with Equation (17.41) yields

\[
M(q) \ddot{r} + C(q, \dot{q}) r + K r = Y \dot{\hat{\theta}}
\]  

(17.109)

Note that, unlike the inverse dynamics control, the modified inner loop control (17.41) does not achieve a linear, decoupled system, even in the known parameter case \( \hat{\theta} = \theta \). However, in this formulation the regressor \( Y \) in Equation (17.109) does not contain the acceleration \( \ddot{q} \) nor is the inverse of the estimated inertia matrix required. These represent distinct advantages over the feedback linearization based schemes.

### 17.3.8 Passivity-Based Robust Control

In the robust passivity-based approach of [36], the term \( \dot{\hat{\theta}} \) in Equation (17.108) is chosen as

\[
\dot{\hat{\theta}} = \theta_0 + u
\]  

(17.110)

where \( \theta_0 \) is a fixed nominal parameter vector and \( u \) is an additional control term. The system (17.109) then becomes

\[
M(q) \ddot{r} + C(q, \dot{q}) r + K r = Y(a, v, q, \ddot{q})(\ddot{\hat{\theta}} + u)
\]  

(17.111)
where \( \hat{\theta} = \theta_0 - \theta \) is a constant vector and represents the parametric uncertainty in the system. If the uncertainty can be bounded by finding a nonnegative constant, \( \rho \geq 0 \), such that

\[
\| \hat{\theta} \| = \| \theta_0 - \theta \| \leq \rho \tag{17.112}
\]

then the additional term \( u \) can be designed according to the expression,

\[
u = \begin{cases} 
-\rho \frac{Y^T r}{||Y^T r||}, & \text{if } ||Y^T r|| > \epsilon \\
-\rho \frac{Y^T r}{\epsilon}, & \text{if } ||Y^T r|| \leq \epsilon
\end{cases} \tag{17.113}
\]

The Lyapunov function

\[
V = \frac{1}{2} r^T M(q) r + \hat{q}^T K \hat{q}
\]

is used to show uniform ultimate boundedness of the tracking error. Note that \( \hat{\theta} \) is constant and so is not a state vector as in adaptive control. Calculating \( \dot{V} \) yields

\[
\dot{V} = -\hat{q}^T \Lambda \hat{q} - \hat{q}^T K \hat{q} + r^T Y (\hat{\theta} + u)
\]

Using the passivity property and the definition of \( r \), this reduces to

\[
\dot{V} = -\hat{q}^T \Lambda \hat{q} - \hat{q}^T K \hat{q} + r^T Y (\hat{\theta} + u)
\]

Uniform ultimate boundedness of the tracking error follows with the control \( u \) from (17.113). See [36] for details.

Comparing this approach with the approach in the section (17.3.5), we see that finding a constant bound \( \rho \) for the constant vector \( \hat{\theta} \) is much simpler than finding a time-varying bound for \( \eta \) in Equation (17.44). The bound \( \rho \) in this case depends only on the inertia parameters of the manipulator, while \( \rho(x,t) \) in Equation (17.69) depends on the manipulator state vector and the reference trajectory and, in addition, requires some assumptions on the estimated inertia matrix \( \hat{M}(q) \).

### 17.3.9 Passivity-Based Adaptive Control

In the adaptive approach the vector \( \hat{\theta} \) in Equation (17.109) is now taken to be a time-varying estimate of the true parameter vector \( \theta \). Combining the control law (17.107) with (17.41) yields

\[
M(q) \dot{r} + C(q, \dot{q}) r + Kr = Y \hat{\theta}
\]

The parameter estimate \( \hat{\theta} \) may be computed using standard methods such as gradient or least squares. For example, using the gradient update law

\[
\hat{\theta} = -\Gamma^{-1} Y^T (\dot{q}, \dot{\hat{q}}, a, v) r
\]

results in global convergence of the tracking errors to zero and boundedness of the parameter estimates since

\[
V = -\hat{q}^T \Lambda \hat{q} - \hat{q}^T K \hat{q} + \hat{\theta}^T (\Gamma \hat{\theta} + Y^T r)
\]

See [38] for details.
17.3.10 Hybrid Control

A Hybrid System is one that has both continuous-time and discrete-event or logic-based dynamics. Supervisory Control, Logic-Based Switching Control, and Multiple-Model Control are typical control architectures in this context. In the robotics context, hybrid schemes can be combined with robust and adaptive control methods to further improve robustness. In particular, because the preceding robust and adaptive control methods provide only asymptotic (i.e., as $t \rightarrow \infty$) error bounds, the transient performance may not be acceptable. Hybrid control methods have been shown to improve transient performance over fixed robust and adaptive controllers.

The use of the term Hybrid Control in this context should not be confused with the notion of Hybrid Position/Force Control [41]. The latter is a familiar approach to force control of manipulators in which the term hybrid refers to the combination of pure force control and pure motion control.

Figure 17.9 shows the Multiple-Model approach of [12], which has been applied to the adaptive control of manipulators. In this architecture, the multiple models have the same structure but may have different nominal parameters in case a robust control scheme is used, or different initial parameter estimates if an adaptive control scheme is used. Because all models have the same inputs and desired outputs, the identification errors $e_i$, are available at each instant for the $j$th model. The idea is then to define a performance measure, for example,

$$J(e_i(t)) = \gamma e_i^2(t) + \beta \int_0^t e_i^2(\sigma)d\sigma \quad \text{with} \quad \gamma, \beta > 0$$  \hspace{1cm} (17.122)

and switch into the closed loop the control input that results in the smallest value of $J$ at each instant.

17.4 Conclusions

We have given a brief overview of the basic results in robust and adaptive control of robot manipulators. In most cases, we have given only the simplest forms of the algorithms, both for ease of exposition and for reasons of space. An extensive literature is available that contains numerous extensions of these basic results. The attached list of references is by no means an exhaustive one. The book [10] is an excellent and
highly detailed treatment of the subject and a good starting point for further reading. Also, the reprint book [37] contains several of the original sources of material surveyed here. In addition, the two survey papers [1] and [28] provide additional details on the robust and adaptive control outlined here.

Several important areas of interest have been omitted for space reasons including output feedback control, learning control, fuzzy control, neural networks, and visual servoing, and control of flexible robots. The reader should consult the references at the end for background on these and other subjects.

References


