CHAPTER

FIVE

CONTROL OF ROBOT MANIPULATORS

> Let us realize that what happens around us
> is largely outside our control, but that
> the way we choose to react to it
> is inside our control.
> *Quoted by J. Petty in "Apples of Gold"*

5.1 INTRODUCTION

Given the dynamic equations of motion of a manipulator, the purpose of robot arm control is to maintain the dynamic response of the manipulator in accordance with some prespecified performance criterion. Although the control problem can be stated in such a simple manner, its solution is complicated by inertial forces, coupling reaction forces, and gravity loading on the links. In general, the control problem consists of (1) obtaining dynamic models of the manipulator, and (2) using these models to determine control laws or strategies to achieve the desired system response and performance. The first part of the control problem has been discussed extensively in Chap. 3. This chapter concentrates on the latter part of the control problem.

From the control analysis point of view, the movement of a robot arm is usually accomplished in two distinct control phases. The first is the gross motion control in which the arm moves from an initial position/orientation to the vicinity of the desired target position/orientation along a planned trajectory. The second is the fine motion control in which the end-effector of the arm dynamically interacts with the object using sensory feedback information to complete the task.

Current industrial approaches to robot arm control system design treat each joint of the robot arm as a simple joint servomechanism. The servomechanism approach models the varying dynamics of a manipulator inadequately because it neglects the motion and configuration of the whole arm mechanism. These changes in the parameters of the controlled system are significant enough to render conventional feedback control strategies ineffective. The result is reduced servo response speed and damping, limiting the precision and speed of the end-effector and making it appropriate only for limited-precision tasks. As a result, manipulators controlled this way move at slow speeds with unnecessary vibrations. Any significant performance gain in this and other areas of robot arm control require the consideration of more efficient dynamic models, sophisticated control techniques, and the use of computer architectures. This chapter focuses on deriving strategies which utilize the dynamic models discussed in Chap. 3 to efficiently control a manipulator.

Considering the robot arm control as a path-trajectory tracking problem (see Fig. 5.1), motion control can be classified into three major categories for the purpose of discussion:

1. Joint motion controls

   Joint servomechanism (PUMA robot arm control scheme)
   Computed torque technique
   Minimum-time control
   Variable structure control
   Nonlinear decoupled control

2. Resolved motion controls (cartesian space control)

   Resolved motion rate control
   Resolved motion acceleration control
   Resolved motion force control

3. Adaptive controls

   Model-referenced adaptive control
   Self-tuning adaptive control
   Adaptive perturbation control with feedforward compensation
   Resolved motion adaptive control

For these control methods, we assume that the desired motion is specified by a time-based path/trajectory of the manipulator either in joint or cartesian coordinates. Each of the above control methods will be described in the following sections.

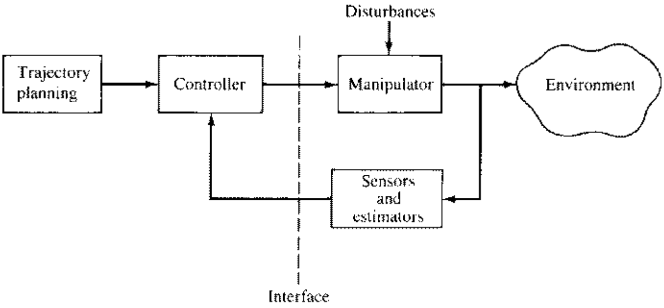

Figure 5.1 Basic control block diagram for robot manipulators.

5.2 CONTROL OF THE PUMA ROBOT ARM

Current industrial practice treats each joint of the robot arm as a simple servomechanism. For the PUMA 560 series robot arm, the controller consists of a DEC LSI-11/02 computer and six Rockwell 6503 microprocessors, each with a joint encoder, a digital-to-analog converter (DAC), and a current amplifier. The control structure is hierarchically arranged. At the top of the system hierarchy is the LSI-11/02 microcomputer which serves as a supervisory computer. At the lower level are the six 6503 microprocessors—one for each degree of freedom (see Fig. 5.2). The LSI-11/02 computer performs two major functions: (1) on-line user interaction and subtask scheduling from the user's VAL† commands, and (2) subtask coordination with the six 6503 microprocessors to carry out the command. The on-line interaction with the user includes parsing, interpreting, and decoding the VAL commands, in addition to reporting appropriate error messages to the user. Once a VAL command has been decoded, various internal routines are called to perform scheduling and coordination functions. These functions, which reside in the EPROM memory of the LSI-11/02 computer, include:

1. Coordinate systems transformations (e.g., from world to joint coordinates or vice versa).
2. Joint-interpolated trajectory planning; this involves sending incremental location updates corresponding to each set point to each joint every 28 ms.
3. Acknowledging from the 6503 microprocessors that each axis of motion has completed its required incremental motion.
4. Looking ahead two instructions to perform continuous path interpolation if the robot is in a continuous path mode.

At the lower level in the system hierarchy are the joint controllers, each of which consists of a digital servo board, an analog servo board, and a power amplifier for each joint. The 6503 microprocessor is an integral part of the joint controller which directly controls each axis of motion. Each microprocessor resides on a digital servo board with its EPROM and DAC. It communicates with the LSI-11/02 computer through an interface board which functions as a demultiplexer that routes trajectory set points information to each joint controller. The interface board is in turn connected to a 16-bit DEC parallel interface board (DRV-11) which transmits the data to and from the Q-bus of the LSI-11/02 (see Fig. 5.2). The microprocessor computes the joint error signal and sends it to the analog servo board which has a current feedback designed for each joint motor.

There are two servo loops for each joint control (see Fig. 5.2). The outer loop provides position error information and is updated by the 6503 microprocessor about every 0.875 ms. The inner loop consists of analog devices and a com-

† VAL is a software package from Unimation Inc. for control of the PUMA robot arm.

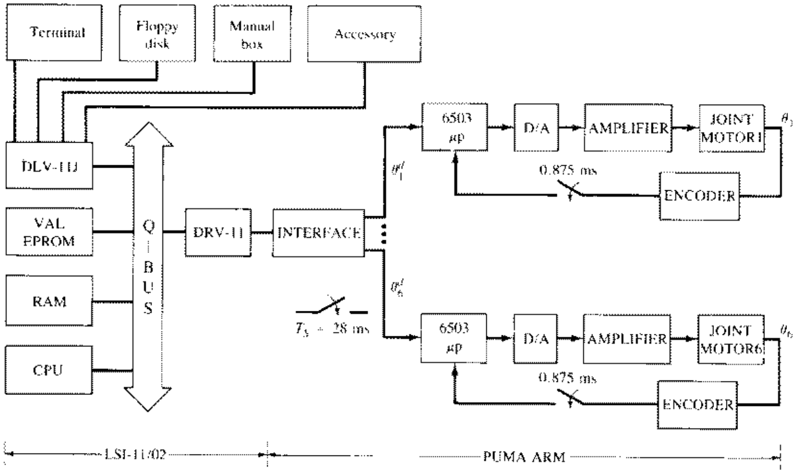

Figure 5.2 PUMA robot arm servo control architecture.

pensator with derivative feedback to dampen the velocity variable. Both servo loop gains are constant and tuned to perform as a "critically damped joint system" at a speed determined by the VAL program. The main functions of the microprocessors include:

1. Every 28 ms, receive and acknowledge trajectory set points from the LSI-11/02 computer and perform interpolation between the current joint value and the desired joint value.
2. Every 0.875 ms, read the register value which stores the incremental values from the encoder mounted at each axis of rotation.
3. Update the error actuating signals derived from the joint-interpolated set points and the values from the axis encoders.
4. Convert the error actuating signal to current using the DACs, and send the current to the analog servo board which moves the joint.

It can be seen that the PUMA robot control scheme is basically a proportional plus integral plus derivative control method (PID controller). One of the main disadvantages of this control scheme is that the feedback gains are constant and prespecified. It does not have the capability of updating the feedback gains under varying payloads. Since an industrial robot is a highly nonlinear system, the inertial loading, the coupling between joints and the gravity effects are all either position-dependent or position- and velocity-dependent terms. Furthermore, at high speeds the inertial loading term can change drastically. Thus, the above control scheme using constant feedback gains to control a nonlinear system does not perform well under varying speeds and payloads. In fact, the PUMA arm moves with noticeable vibration at reduced speeds. One solution to the problem is the use of digital control in which the applied torques to the robot arm are obtained by a computer based on an appropriate dynamic model of the arm. A version of this method is discussed in Sec. 5.3.

5.3 COMPUTED TORQUE TECHNIQUE

Given the Lagrange-Euler or Newton-Euler equations of motion of a manipulator, the control problem is to find appropriate torques/forces to servo all the joints of the manipulator in real time in order to track a desired time-based trajectory as closely as possible. The drive motor torque required to servo the manipulator is based on a dynamic model of the manipulator (L-E or N-E formulations). The motor-voltage (or motor-current) characteristics are also modeled in the computation scheme and the computed torque is converted to the applied motor voltage (or current). This applied voltage is computed at such a high rate that sampling effects generally can be ignored in the analysis.

Because of modeling errors and parameter variations in the model, position and derivative feedback signals will be used to compute the correction torques which, when added to the torques computed based on the manipulator model, provide the corrective drive signal for the joint motors.

5.3.1 Transfer Function of a Single Joint

This section deals with the derivation of the transfer function of a single joint robot from which a proportional plus derivative controller (PD controller) will be obtained. This will be followed by a discussion of controller design for multijoint manipulators based on the Lagrange-Euler and/or Newton-Euler equations of motion. The analysis here treats the "single-joint" robot arm as a continuous time system, and the Laplace transform technique is used to simplify the analysis.

Most industrial robots are either electrically, hydraulically, or pneumatically actuated. Electrically driven manipulators are constructed with a dc permanent magnet torque motor for each joint. Basically, the dc torque motor is a permanent magnet, armature excited, continuous rotation motor incorporating such features as high torque-power ratios, smooth, low-speed operation, linear torque-speed characteristics, and short time constants. Use of a permanent magnet field and dc power provide maximum torque with minimum input power and minimum weight. These features also reduce the motor inductance and hence the electrical time constant. In Fig. 5.3, an equivalent circuit of an armature-controlled dc permanent magnet torque motor for a joint is shown based on the following variables:

V_a	Armature voltage, volts
V_f	Field voltage, volts
L_a	Armature inductance, Henry
L_f	Field inductance, Henry
R_a	Armature resistance, ohms
R_f	Field resistance, ohms

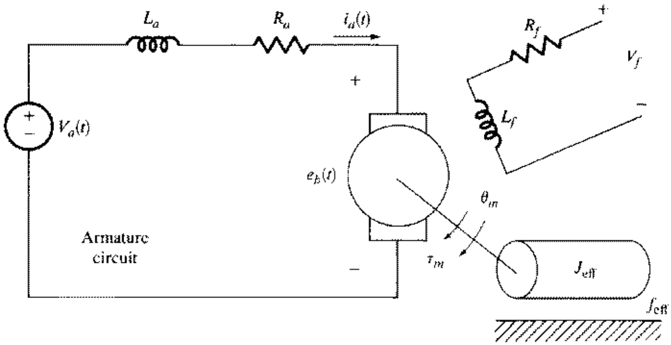

Figure 5.3 Equivalent circuit of an armature-controlled dc motor.

i_a	Armature current, amperes
i_f	Field current, amperes
e_b	Back electromotive force (emf), volts
τ	Torque delivered by the motor, oz·in
θ_m	Angular displacement of the motor shaft, radians
θ_L	Angular displacement of the load shaft, radians
J_m	Moment of inertia of the motor referred to the motor shaft, oz · in · s^2/ rad
f_m	Viscous-friction coefficient of the motor referred to the motor shaft, oz · in · s/rad
J_L	Moment of inertia of the load referred to the load shaft, oz · in · s^2/rad
f_L	Viscous-friction coefficient of the load referred to the load shaft, oz · in · s/rad
N_m	Number of teeth of the input gear (motor gear)
N_L	Number of teeth of the output gear (load gear)

The motor shaft is coupled to a gear train to the load of the link. With reference to the gear train shown in Fig. 5.4, the total linear distance traveled on each gear is the same. That is,

$$d_m = d_L \qquad \text{and} \qquad r_m\theta_m = r_L\theta_L \tag{5.3-1}$$

where r_m and r_L are, respectively, the radii of the input gear and the output gear. Since the radius of the gear is proportional to the number of teeth it has, then

$$N_m\theta_m = N_L\theta_L \tag{5.3-2}$$

or

$$\frac{N_m}{N_L} = \frac{\theta_L}{\theta_m} = n < 1 \tag{5.3-3}$$

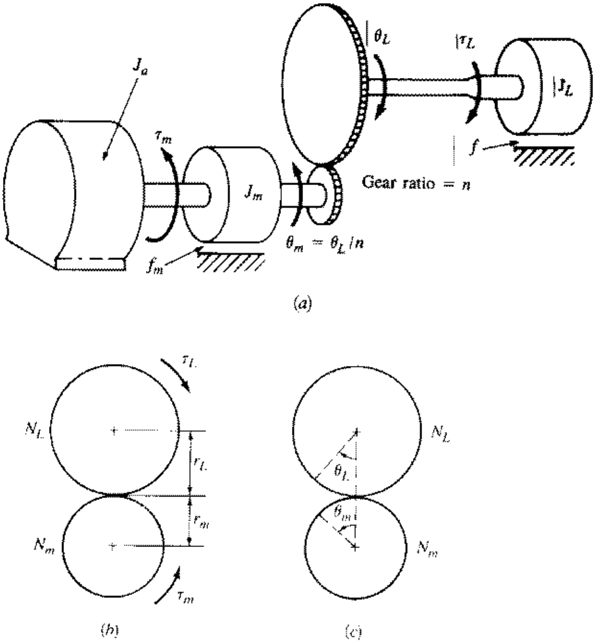

Figure 5.4 Analysis of a gear train.

where n is the gear ratio and it relates θ_L to θ_m by

$$\theta_L(t) = n\theta_m(t) \tag{5.3-4}$$

Taking the first two time derivatives, we have

$$\dot{\theta}_L(t) = n\dot{\theta}_m(t) \tag{5.3-5}$$

and

$$\ddot{\theta}_L(t) = n\ddot{\theta}_m(t) \tag{5.3-6}$$

If a load is attached to the output gear, then the torque developed at the motor shaft is equal to the sum of the torques dissipated by the motor and its load. That is,

$$\begin{bmatrix}\text{Torque from}\\ \text{motor}\\ \text{shaft}\end{bmatrix} = \begin{bmatrix}\text{torque}\\ \text{on}\\ \text{motor}\end{bmatrix} + \begin{bmatrix}\text{torque on load}\\ \text{referred to}\\ \text{the motor shaft}\end{bmatrix} \tag{5.3-7}$$

or, in equation form,

$$\tau(t) = \tau_m(t) + \tau_L^*(t) \tag{5.3-8}$$

The load torque referred to the load shaft is

$$\tau_L(t) = J_L\ddot{\theta}_L(t) + f_L\dot{\theta}_L(t) \tag{5.3-9}$$

and the motor torque referred to the motor shaft is

$$\tau_m(t) = J_m\ddot{\theta}_m(t) + f_m\dot{\theta}_m(t) \tag{5.3-10}$$

Recalling that conservation of work requires that the work done by the load referred to the load shaft, $\tau_L\theta_L$, be equal to the work done by the load referred to the motor shaft, $\tau_L^*\theta_m$, leads to

$$\tau_L^*(t) = \frac{\tau_L(t)\theta_L(t)}{\theta_m(t)} = n\tau_L(t) \tag{5.3-11}$$

Using Eqs. (5.3-9), (5.3-5), and (5.3-6), we have

$$\tau_L^*(t) = n^2[J_L\ddot{\theta}_m(t) + f_L\dot{\theta}_m(t)] \tag{5.3-12}$$

Using Eqs. (5.3-10) and (5.3-12), the torque developed at the motor shaft [Eq. (5.3-8)] is

$$\begin{aligned}\tau(t) &= \tau_m(t) + \tau_L^*(t) = (J_m + n^2J_L)\ddot{\theta}_m(t) + (f_m + n^2f_L)\dot{\theta}_m(t)\\ &= J_{\text{eff}}\,\ddot{\theta}_m(t) + f_{\text{eff}}\,\dot{\theta}_m(t)\end{aligned} \tag{5.3-13}$$

where $J_{\text{eff}} = J_m + n^2J_L$ is the effective moment of inertia of the combined motor and load referred to the motor shaft and $f_{\text{eff}} = f_m + n^2f_L$ is the effective viscous friction coefficient of the combined motor and load referred to the motor shaft.

Based on the above results, we can now derive the transfer function of this single joint manipulator system. Since the torque developed at the motor shaft increases linearly with the armature current, independent of speed and angular position, we have

$$\tau(t) = K_a i_a(t) \tag{5.3-14}$$

where K_a is known as the motor-torque proportional constant in oz · in/A. Applying Kirchhoff's voltage law to the armature circuit, we have

$$V_a(t) = R_a i_a(t) + L_a\frac{di_a(t)}{dt} + e_b(t) \tag{5.3-15}$$

where e_b is the back electromotive force (emf) which is proportional to the angular velocity of the motor,

$$e_b(t) = K_b\dot{\theta}_m(t) \tag{5.3-16}$$

and K_b is a proportionality constant in V · s/rad. Taking the Laplace transform of the above equations and solving for $I_a(s)$, we have

$$I_a(s) = \frac{V_a(s) - sK_b\Theta_m(s)}{R_a + sL_a} \tag{5.3-17}$$

Taking the Laplace transform of Eq. (5.3-13), we have

$$T(s) = s^2 J_{\text{eff}}\Theta_m(s) + s f_{\text{eff}}\Theta_m(s) \tag{5.3-18}$$

Taking the Laplace transform of Eq. (5.3-14), and substituting $I_a(s)$ from Eq. (5.3-17), we have

$$T(s) = K_a I_a(s) = K_a \left[\frac{V_a(s) - sK_b\Theta_m(s)}{R_a + sL_a} \right] \tag{5.3-19}$$

Equating Eqs. (5.3-18) and (5.3-19) and rearranging the terms, we obtain the transfer function from the armature voltage to the angular displacement of the motor shaft,

$$\frac{\Theta_m(s)}{V_a(s)} = \frac{K_a}{s[s^2 J_{\text{eff}}L_a + (L_a f_{\text{eff}} + R_a J_{\text{eff}})s + R_a f_{\text{eff}} + K_a K_b]} \tag{5.3-20}$$

Since the electrical time constant of the motor is much smaller than the mechanical time constant, we can neglect the armature inductance effect, L_a. This allows us to simplify the above equation to

$$\frac{\Theta_m(s)}{V_a(s)} = \frac{K_a}{s(sR_a J_{\text{eff}} + R_a f_{\text{eff}} + K_a K_b)} = \frac{K}{s(T_m s + 1)} \tag{5.3-21}$$

where

$$K \triangleq \frac{K_a}{R_a f_{\text{eff}} + K_a K_b} \qquad \text{motor gain constant}$$

and $$T_m \triangleq \frac{R_a J_{\text{eff}}}{R_a f_{\text{eff}} + K_a K_b} \qquad \text{motor time constant}$$

Since the output of the control system is the angular displacement of the joint $[\Theta_L(s)]$, using Eq. (5.3-4) and its Laplace transformed equivalence, we can relate the angular position of the joint $\Theta_L(s)$ to the armature voltage $V_a(s)$,

$$\frac{\Theta_L(s)}{V_a(s)} = \frac{nK_a}{s(sR_a J_{\text{eff}} + R_a f_{\text{eff}} + K_a K_b)} \tag{5.3-22}$$

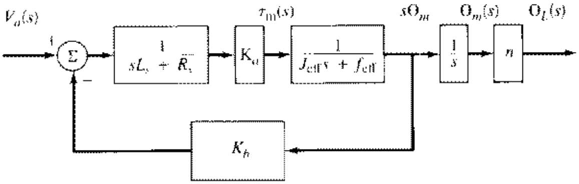

Figure 5.5 Open-loop transfer function of a single-joint robot arm.

Eq. (5.3-22) is the transfer function of the "single-joint" manipulator relating the applied voltage to the angular displacement of the joint. The block diagram of the system is shown in Fig. 5.5.

5.3.2 Positional Controller for a Single Joint

The purpose of a positional controller is to servo the motor so that the actual angular displacement of the joint will track a desired angular displacement specified by a preplanned trajectory, as discussed in Chap. 4. The technique is based on using the error signal between the desired and actual angular positions of the joint to actuate an appropriate voltage. In other words, the applied voltage to the motor is linearly proportional to the error between the desired and actual angular displacement of the joint,

$$V_a(t) = \frac{K_p e(t)}{n} = \frac{K_p[\theta_L^d(t) - \theta_L(t)]}{n} \tag{5.3-23}$$

where K_p is the position feedback gain in volts per radian, $e(t) = \theta_L^d(t) - \theta_L(t)$ is the system error, and the gear ratio n is included to compute the applied voltage referred to the motor shaft. Equation (5.3-23) indicates that the actual angular displacement of the joint is fed back to obtain the error which is amplified by the position feedback gain K_p to obtain the applied voltage. In reality, we have changed the single-joint robot system from an open-loop control system [Eq. (5.3-22)] to a closed-loop control system with unity negative feedback. This closed-loop control system is shown in Fig. 5.6. The actual angular position of the joint can be measured either by an optical encoder or by a potentiometer.

Taking the Laplace transform of Eq. (5.3-23),

$$V_a(s) = \frac{K_p[\Theta_L^d(s) - \Theta_L(s)]}{n} = \frac{K_p E(s)}{n} \tag{5.3-24}$$

and substituting $V_a(s)$ into Eq. (5.3-22), yields the open-loop transfer function relating the error actuating signal $[E(s)]$ to the actual displacement of the joint:

$$\frac{\Theta_L(s)}{E(s)} \triangleq G(s) = \frac{K_a K_p}{s(sR_a J_{\text{eff}} + R_a f_{\text{eff}} + K_a K_b)} \tag{5.3-25}$$

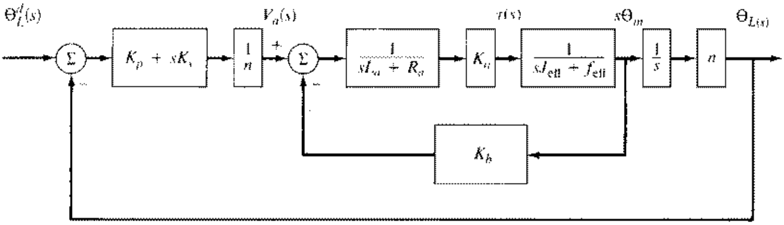

Figure 5.6 Feedback control of a single-joint manipulator.

After some simple algebraic manipulation, we can obtain the closed-loop transfer function relating the actual angular displacement $\Theta_L(s)$ to the desired angular displacement $\Theta_L^d(s)$:

$$\frac{\Theta_L(s)}{\Theta_L^d(s)} = \frac{G(s)}{1 + G(s)} = \frac{K_a K_p}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b) + K_a K_p}$$

$$= \frac{K_a K_p / R_a J_{\text{eff}}}{s^2 + [(R_a f_{\text{eff}} + K_a K_b)/R_a J_{\text{eff}}]\, s + K_a K_p / R_a J_{\text{eff}}} \tag{5.3-26}$$

Equation (5.3-26) shows that the proportional controller for the single-joint robot is a second-order system which is always stable if all the system parameters are positive. In order to increase the system response time and reduce the steady-state error, one can increase the positional feedback gain K_p and incorporate some damping into the system by adding a derivative of the positional error. The angular velocity of the joint can be measured by a tachometer or approximated from the position data between two consecutive sampling periods. With this added feedback term, the applied voltage to the joint motor is linearly proportional to the position error and its derivative; that is,

$$V_a(t) = \frac{K_p[\theta_L^d(t) - \theta_L(t)] + K_v[\dot{\theta}_L^d(t) - \dot{\theta}_L(t)]}{n}$$

$$= \frac{K_p e(t) + K_v \dot{e}(t)}{n} \tag{5.3-27}$$

where K_v is the error derivative feedback gain, and the gear ratio n is included to compute the applied voltage referred to the motor shaft. Equation (5.3-27) indicates that, in addition to the positional error feedback, the velocity of the motor is measured or computed and fed back to obtain the velocity error which is multiplied by the velocity feedback gain K_v. Since, as discussed in Chap. 4, the desired joint trajectory can be described by smooth polynomial functions whose first two time derivatives exist within $[t_0, t_f]$, the desired velocity can be computed from the polynomial function and utilized to obtain the velocity error for feedback purposes. The summation of these voltages is then applied to the joint motor. This closed-loop control system is shown in Fig. 5.6.

Taking the Laplace transform of Eq. (5.3-27) and substituting $V_a(s)$ into Eq. (5.3-22) yields the transfer function relating the error actuating signal $[E(s)]$ to the actual displacement of the joint:

$$\frac{\Theta_L(s)}{E(s)} \triangleq G_{PD}(s) = \frac{K_a(K_p + sK_v)}{s(sR_aJ_{\text{eff}} + R_af_{\text{eff}} + K_aK_b)}$$
$$= \frac{K_aK_vs + K_aK_p}{s(sR_aJ_{\text{eff}} + R_af_{\text{eff}} + K_aK_b)} \tag{5.3-28}$$

Some simple algebraic manipulation yields the closed-loop transfer function relating the actual angular displacement $[\Theta_L(s)]$ to the desired angular displacement $[\Theta_L^d(s)]$:

$$\frac{\Theta_L(s)}{\Theta_L^d(s)} = \frac{G_{PD}(s)}{1 + G_{PD}(s)}$$
$$= \frac{K_aK_vs + K_aK_p}{s^2R_aJ_{\text{eff}} + s(R_af_{\text{eff}} + K_aK_b + K_aK_v) + K_aK_p} \tag{5.3-29}$$

Note that if K_v is equal to zero, Eq. (5.3-29) reduces to Eq. (5.3-26).

Equation (5.3-29) is a second-order system with a finite zero located at $-K_p/K_v$ in the left half plane of the s plane. Depending on the location of this zero, the system could have a large overshoot and a long settling time. From Fig. 5.7, we notice that the manipulator system is also under the influence of disturbances $[D(s)]$ which are due to gravity loading and centrifugal effects of the link. Because of this disturbance, the torque generated at the motor shaft has to compensate for the torques dissipated by the motor, the load, and also the disturbances. Thus, from Eq. (5.3-18),

$$T(s) = [s^2J_{\text{eff}} + sf_{\text{eff}}]\Theta_m(s) + D(s) \tag{5.3-30}$$

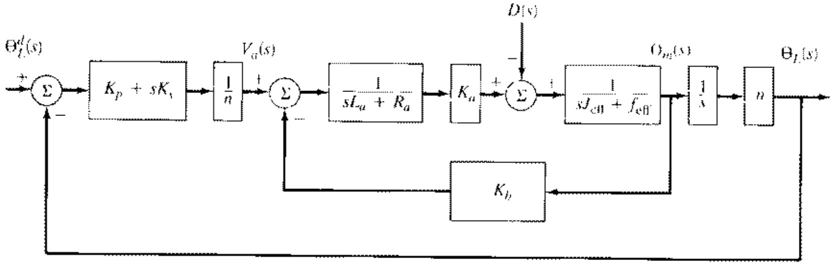

Figure 5.7 Feedback control block diagram of a manipulator with disturbances.

where $D(s)$ is the Laplace transform equivalent of the disturbances. The transfer function relating the disturbance inputs to the actual joint displacement is given by

$$\left.\frac{\Theta_L(s)}{D(s)}\right|_{\Theta_L^d(s)=0} = \frac{-nR_a}{s^2R_aJ_{\text{eff}}+s(R_af_{\text{eff}}+K_aK_b+K_aK_v)+K_aK_p} \qquad (5.3\text{-}31)$$

From Eqs. (5.3-29) and (5.3-31) and using the superposition principle, we can obtain the actual displacement of the joint from these two inputs, as follows:

$$\Theta_L(s) = \frac{K_a(K_p + sK_v)\Theta_L^d(s) - nR_aD(s)}{s^2R_aJ_{\text{eff}} + s(R_af_{\text{eff}} + K_aK_b + K_aK_v) + K_aK_p} \qquad (5.3\text{-}32)$$

We are interested in looking at the performance of the above closed-loop system, with particular emphasis on the steady state error of the system due to step and ramp inputs and the bounds of the position and velocity feedback gains. This is covered in the following section.

5.3.3 Performance and Stability Criteria

The performance of a closed-loop second-order control system is based on several criteria, such as fast rise time, small or zero steady-state error, and fast settling time. We shall first investigate the bounds for the position and velocity feedback gains. Assuming for a moment that the disturbances are zero, we see that from Eqs. (5.3-29) and (5.3-31) that the system is basically a second-order system with a finite zero, as indicated in the previous section. The effect of this finite zero usually causes a second-order system to peak early and to have a larger overshoot (than the second-order system without a finite zero). We shall temporarily ignore the effect of this finite zero and try to determine the values of K_p and K_v to have a critically damped or overdamped system.

The reader will recall that the characteristic equation of a second-order system can be expressed in the following standard form:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \qquad (5.3\text{-}33)$$

where ζ and ω_n are, respectively, the damping ratio and the undamped natural frequency of the system. Relating the closed-loop poles of Eq. (5.3-29) to Eq. (5.3-33), we see that

$$\omega_n^2 = \frac{K_aK_p}{J_{\text{eff}}R_a} \qquad (5.3\text{-}34)$$

and

$$2\zeta\omega_n = \frac{R_af_{\text{eff}} + K_aK_b + K_aK_v}{J_{\text{eff}}R_a} \qquad (5.3\text{-}35)$$

The performance of the second-order system is dictated by its natural undamped frequency ω_n and the damping ratio ζ. For reasons of safety, the manipulator system cannot have an underdamped response for a step input. In order to have good performance (as outlined above), we would like to have a critically damped or an overdamped system, which requires that the system damping ratio be greater than or equal to unity. From Eq. (5.3-34), the position feedback gain is found from the natural frequency of the system:

$$K_p = \frac{\omega_n^2 J_{\text{eff}} R_a}{K_a} > 0 \tag{5.3-36}$$

Substituting ω_n from Eq. (5.3-34) into Eq. (5.3-35), we find that

$$\zeta = \frac{R_a f_{\text{eff}} + K_a K_b + K_a K_v}{2\sqrt{K_a K_p J_{\text{eff}} R_a}} \geqslant 1 \tag{5.3-37}$$

where the equality of the above equation gives a critically damped system response and the inequality gives an overdamped system response. From Eq. (5.3-37), the velocity feedback gain K_v can be found to be

$$K_v \geqslant \frac{2\sqrt{K_a K_p J_{\text{eff}} R_a} - R_a f_{\text{eff}} - K_a K_b}{K_a} \tag{5.3-38}$$

In order not to excite the structural oscillation and resonance of the joint, Paul [1981] suggested that the undamped natural frequency ω_n may be set to no more than one-half of the structural resonant frequency of the joint, that is,

$$\omega_n \leqslant 0.5\omega_r \tag{5.3-39}$$

where ω_r is the structural resonant frequency in radians per second.

The structural resonant frequency is a property of the material used in constructing the manipulator. If the effective stiffness of the joint is k_{stiff}, then the restoring torque $k_{\text{stiff}}\theta_m(t)$ opposes the inertial torque of the motor,

$$J_{\text{eff}}\ddot{\theta}_m(t) + k_{\text{stiff}}\theta_m(t) = 0 \tag{5.3-40}$$

Taking the Laplace transform, the characteristic equation of Eq. (5.3-40) is

$$J_{\text{eff}}s^2 + k_{\text{stiff}} = 0 \tag{5.3-41}$$

and solving the above characteristic equation gives the structural resonant frequency of the system

$$\omega_r = \left[\frac{k_{\text{stiff}}}{J_{\text{eff}}}\right]^{1/2} \tag{5.3-42}$$

Although the stiffness of the joint is fixed, if a load is added to the manipulator's end-effector, the effective moment of inertia will increase which, in effect, reduces the structural resonant frequency. If a structural resonant frequency ω_0 is measured at a known moment of inertia J_0, then the structural resonant frequency at the other moment of inertia J_{eff} is given by

$$\omega_r = \omega_0 \left[\frac{J_0}{J_{\text{eff}}} \right]^{1/2} \tag{5.3-43}$$

Using the condition of Eq. (5.3-39), K_p from Eq. (5.3-36) is bounded by

$$0 < K_p \leqslant \frac{\omega_r^2 J_{\text{eff}} R_a}{4K_a} \tag{5.3-44}$$

which, using Eq. (5.3-43), reduces to

$$0 < K_p \leqslant \frac{\omega_0^2 J_0 R_a}{4K_a} \tag{5.3-45}$$

After finding K_p, the velocity feedback gain K_v can be found from Eq. (5.3-38):

$$K_v \geqslant \frac{R_a \omega_0 \sqrt{J_0 J_{\text{eff}}} - R_a f_{\text{eff}} - K_a K_b}{K_a} \tag{5.3-46}$$

Next we investigate the steady-state errors of the above system for step and ramp inputs. The system error is defined as $e(t) = \theta_L^d(t) - \theta_L(t)$. Using Eq. (5.3-32), the error in the Laplace transform domain can be expressed as

$$\begin{aligned} E(s) &= \Theta_L^d(s) - \Theta_L(s) \\ &= \frac{[s^2 J_{\text{eff}} R_a + s(R_a f_{\text{eff}} + K_a K_b)]\Theta_L^d(s) + nR_a D(s)}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \end{aligned} \tag{5.3-47}$$

For a step input of magnitude A, that is, $\theta_L^d(t) = A$, and if the disturbance input is unknown, then the steady-state error of the system due to a step input can be found from the final value theorem, provided the limits exist; that is,

$$\begin{aligned} e_{\text{ss}}(\text{step}) &\triangleq e_{\text{ssp}} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) \\ &= \lim_{s\to 0} s \frac{[(s^2 J_{\text{eff}} R_a + s(R_a f_{\text{eff}} + K_a K_b)]A/s + nR_a D(s)}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \\ &= \lim_{s\to 0} s \left[\frac{nR_a D(s)}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \right] \end{aligned} \tag{5.3-48}$$

which is a function of the disturbances. Fortunately, we do know some of the disturbances, such as gravity loading and centrifugal torque due to the velocity of the joint. Other disturbances that we generally do not know are the frictional torque due to the gears and the system noise. Thus, we can identify each of these torques separately as

$$\tau_D(t) = \tau_G(t) + \tau_C(t) + \tau_e \tag{5.3-49}$$

where $\tau_G(t)$ and $\tau_C(t)$ are, respectively, torques due to gravity and centrifugal effects of the link, and τ_e are disturbances other than the gravity and centrifugal torques and can be assumed to be a very small constant value. The corresponding Laplace transform of Eq. (5.3-49) is

$$D(s) = T_G(s) + T_C(s) + \frac{T_e}{s} \tag{5.3-50}$$

To compensate for gravity loading and centrifugal effects, we can precompute these torque values and feed the computed torques forward into the controller to minimize their effects as shown in Fig. 5.8. This is called *feedforward compensation.*

Let us denote the computed torques as $\tau_{\text{comp}}(t)$ whose Laplace transform is $T_{\text{comp}}(s)$. With this computed torque and using Eq. (5.3-50), the error equation of Eq. (5.3-47) is modified to

$$E(s) = \frac{[s^2 J_{\text{eff}} R_a + s(R_a f_{\text{eff}} + K_a K_b)]\Theta_L^d(s) + nR_a[T_G(s) + T_C(s) + T_e/s - T_{\text{comp}}(s)]}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \tag{5.3-51}$$

For a step input, $\Theta_L^d(s) = A/s$, the steady-state position error of the system is given by

$$e_{\text{ssp}} = \lim_{s \to 0} s \left[\frac{nR_a[T_G(s) + T_C(s) + T_e/s - T_{\text{comp}}(s)]}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \right] \tag{5.3-52}$$

For the steady-state position error, the contribution from the disturbances due to the centrifugal effect is zero as time approaches infinity. The reason for this is that the centrifugal effect is a function of $\dot{\theta}_L^2(t)$ and, as time approaches infinity, $\dot{\theta}_L(\infty)$ approaches zero. Hence, its contribution to the steady-state position error is zero. If the computed torque $\tau_{\text{comp}}(t)$ is equivalent to the gravity loading of the link, then the steady-state position error reduces to

$$e_{\text{ssp}} = \frac{nR_a T_e}{K_a K_p} \tag{5.3-53}$$

Since K_p is bounded by Eq. (5.3-45), the above steady-state position error reduces to

$$e_{\text{ssp}} = \frac{4nT_e}{\omega_0^2 J_0} \tag{5.3-54}$$

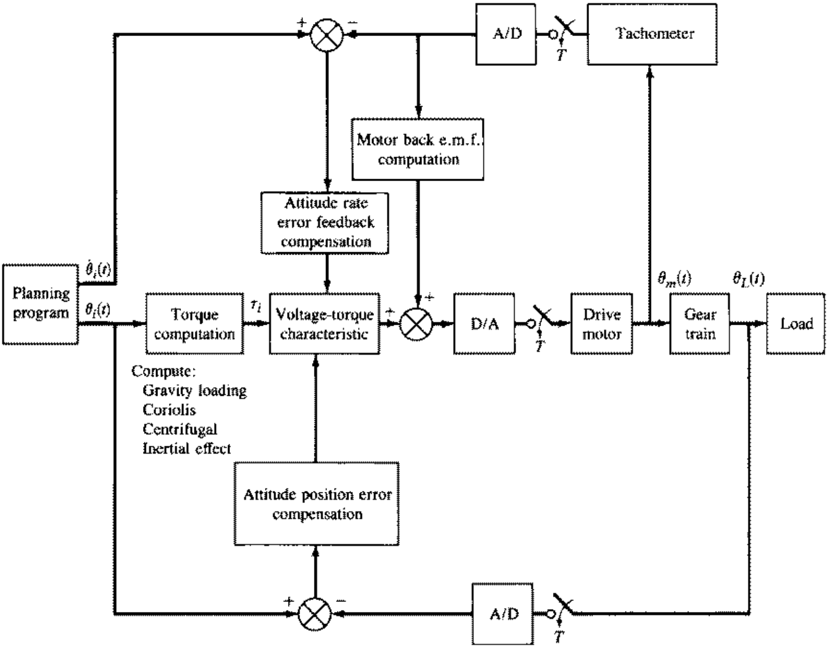

Figure 5.8 Compensation of disturbances.

which is small because τ_e is assumed to be small. The computation of $\tau_G(t)$ will be discussed later using the dynamic model of the manipulator.

If the input to the system is a ramp function, then $\Theta_L^d(s) = A/s^2$, and if we again assume that the disturbances are known as in Eq. (5.3-50), then the steady-state error of the system due to a ramp input is

$$
\begin{aligned}
e_{ss}(\text{ramp}) \triangleq e_{ssv} &= \lim_{s\to 0} s \frac{[s^2 J_{\text{eff}} R_a + s(R_a f_{\text{eff}} + K_a K_b)]A/s^2}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \\
&\quad + \lim_{s\to 0} s \frac{nR_a[T_G(s) + T_C(s) + T_e/s - T_{\text{comp}}(s)]}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \\
&= \frac{(R_a f_{\text{eff}} + K_a K_b)A}{K_a K_p} \\
&\quad + \lim_{s\to 0} s \frac{nR_a[T_G(s) + T_C(s) + T_e/s - T_{\text{comp}}(s)]}{s^2 R_a J_{\text{eff}} + s(R_a f_{\text{eff}} + K_a K_b + K_a K_v) + K_a K_p} \qquad (5.3\text{-}55)
\end{aligned}
$$

Again, in order to reduce the steady-state velocity error, the computed torque $[\tau_{\text{comp}}(t)]$ needs to be equivalent to the gravity and centrifugal effects. Thus, the steady-state velocity error reduces to

$$e_{\text{ssv}} = \frac{(R_a f_{\text{eff}} + K_a K_b)A}{K_a K_p} + e_{\text{ssp}} \tag{5.3-56}$$

which has a finite steady-state error. The computation of $\tau_{\text{comp}}(t)$ depends on the dynamic model of the manipulator. In general, as discussed in Chap. 3, the Lagrange-Euler equations of motion of a six-joint manipulator, excluding the dynamics of the electronic control device, gear friction, and backlash, can be written as [Eq. (3.2-24)]

$$\begin{aligned}\tau_i(t) = & \sum_{k=i}^{6} \sum_{j=1}^{k} \text{Tr}\left[\frac{\partial\, {}^0\mathbf{T}_k}{\partial q_j}\mathbf{J}_k\left(\frac{\partial\, {}^0\mathbf{T}_k}{\partial q_i}\right)^T\right]\ddot{q}_j(t)\\ & + \sum_{r=i}^{6}\sum_{j=1}^{r}\sum_{k=1}^{r} \text{Tr}\left[\frac{\partial^2\, {}^0\mathbf{T}_r}{\partial q_j \partial q_k}\mathbf{J}_r\left(\frac{\partial\, {}^0\mathbf{T}_r}{\partial q_i}\right)^T\right]\dot{q}_j(t)\dot{q}_k(t)\\ & - \sum_{j=i}^{6} m_j \mathbf{g}\left(\frac{\partial\, {}^0\mathbf{T}_j}{\partial q_i}\right)\bar{\mathbf{r}}_j \qquad \text{for } i = 1, 2, \ldots, 6\end{aligned} \tag{5.3-57}$$

where $\tau_i(t)$ is the generalized applied torque for joint i to drive the ith link, $\dot{q}_i(t)$ and $\ddot{q}_i(t)$ are the angular velocity and angular acceleration of joint i, respectively, and q_i is the generalized coordinate of the manipulator and indicates its angular position. ${}^0\mathbf{T}_i$ is a 4 × 4 homogeneous link transformation matrix which relates the spatial relationship between two coordinate frames (the ith and the base coordinate frames), $\bar{\mathbf{r}}_i$ is the position of the center of mass of link i with respect to the ith coordinate system, $\mathbf{g} = (g_x, g_y, g_z, 0)$ is the gravity row vector and $|\mathbf{g}| = 9.8062$ m/s^2, and $\mathbf{J}_i$ is the pseudo-inertia matrix of link i about the ith coordinate frame and can be written as in Eq. (3.2-18).

Equation (5.3-57) can be expressed in matrix form explicitly as

$$\sum_{k=1}^{6} D_{ik}\ddot{q}_k(t) + \sum_{k=1}^{6}\sum_{m=1}^{6} h_{ikm}\dot{q}_k(t)\dot{q}_m(t) + c_i = \tau_i(t) \qquad i = 1, 2, \ldots, 6 \tag{5.3-58}$$

where

$$D_{ik} = \sum_{j=\max(i,k)}^{6} \text{Tr}\left[\frac{\partial\, {}^0\mathbf{T}_j}{\partial q_k}\mathbf{J}_j\left(\frac{\partial\, {}^0\mathbf{T}_j}{\partial q_i}\right)^T\right] \qquad i,k = 1, 2, \ldots, 6 \tag{5.3-59}$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{6} \text{Tr}\left[\frac{\partial^2\, {}^0\mathbf{T}_j}{\partial q_k \partial q_m}\mathbf{J}_j\left(\frac{\partial\, {}^0\mathbf{T}_j}{\partial q_i}\right)^T\right] \qquad i,k,m = 1, 2, \ldots, 6 \tag{5.3-60}$$

$$c_i = \sum_{j=i}^{6} \left[-m_j \mathbf{g} \left(\frac{\partial\, {}^0\mathbf{T}_j}{\partial q_i} \right) \bar{\mathbf{r}}_j \right] \qquad i = 1, 2, \ldots, 6 \tag{5.3-61}$$

Eq. (5.3-57) can be rewritten in a matrix notation as

$$\tau_i(t) = [D_{i1}, D_{i2}, D_{i3}, D_{i4}, D_{i5}, D_{i6}] \begin{bmatrix} \ddot{q}_1(t) \\ \ddot{q}_2(t) \\ \ddot{q}_3(t) \\ \ddot{q}_4(t) \\ \ddot{q}_5(t) \\ \ddot{q}_6(t) \end{bmatrix}$$

$$+ \; [\dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)] \tag{5.3-62}$$

$$\times \begin{bmatrix} h_{i11} & h_{i12} & h_{i13} & h_{i14} & h_{i15} & h_{i16} \\ h_{i21} & h_{i22} & h_{i23} & h_{i24} & h_{i25} & h_{i26} \\ h_{i31} & h_{i32} & h_{i33} & h_{i34} & h_{i35} & h_{i36} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ h_{i61} & h_{i62} & h_{i63} & h_{i64} & h_{i65} & h_{i66} \end{bmatrix} \begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \\ \dot{q}_4(t) \\ \dot{q}_5(t) \\ \dot{q}_6(t) \end{bmatrix} + c_i$$

Using the Lagrange-Euler equations of motion as formulated above, the computed torque for the gravity loading and centrifugal and Coriolis effects for joint i can be found, respectively, as

$$\tau_G(t) = c_i \qquad i = 1, 2, \ldots, 6 \tag{5.3-63}$$

and

$$\tau_C(t) = [\dot{q}_1(t), \dot{q}_2(t), \dot{q}_3(t), \dot{q}_4(t), \dot{q}_5(t), \dot{q}_6(t)]$$

$$\times \begin{bmatrix} h_{i11} & h_{i12} & h_{i13} & h_{i14} & h_{i15} & h_{i16} \\ h_{i21} & h_{i22} & h_{i23} & h_{i24} & h_{i25} & h_{i26} \\ h_{i31} & h_{i32} & h_{i33} & h_{i34} & h_{i35} & h_{i36} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ h_{i61} & h_{i62} & h_{i63} & h_{i64} & h_{i65} & h_{i66} \end{bmatrix} \begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \\ \dot{q}_4(t) \\ \dot{q}_5(t) \\ \dot{q}_6(t) \end{bmatrix} \quad i = 1, 2, \ldots, 6 \tag{5.3-64}$$

This compensation leads to what is usually known as the "inverse dynamics problem" or "computed torque" technique. This is covered in the next section.

5.3.4 Controller for Multijoint Robots

For a manipulator with multiple joints, one of the basic control schemes is the computed torque technique based on the L-E or the N-E equations of motion. Basically the computed torque technique is a feedforward control and has feedforward and feedback components. The control components compensate for the interaction forces among all the various joints and the feedback component computes the necessary correction torques to compensate for any deviations from the desired trajectory. It assumes that one can accurately compute the counterparts of $\mathbf{D}(\mathbf{q})$, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{c}(\mathbf{q})$ in the L-E equations of motion [Eq. (3.2-26)] to minimize their nonlinear effects, and use a proportional plus derivative control to servo the joint motors. Thus, the structure of the control law has the form of

$$\tau(t) = \mathbf{D}_a(\mathbf{q})\{\ddot{\mathbf{q}}^d(t) + \mathbf{K}_v[\dot{\mathbf{q}}^d(t) - \dot{\mathbf{q}}(t)] + \mathbf{K}_p[\mathbf{q}^d(t) - \mathbf{q}(t)]\} + \mathbf{h}_a(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{c}_a(\mathbf{q}) \tag{5.3-65}$$

where $\mathbf{K}_v$ and $\mathbf{K}_p$ are 6×6 derivative and position feedback gain matrices, respectively, and the manipulator has 6 degrees of freedom.

Substituting $\tau(t)$ from Eq. (5.3-65) into Eq. (3.2-26), we have

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}}(t) + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{c}(\mathbf{q}) = \mathbf{D}_a(\mathbf{q})\{\ddot{\mathbf{q}}^d(t) + \mathbf{K}_v[\dot{\mathbf{q}}^d(t) - \dot{\mathbf{q}}(t)] + \mathbf{K}_p[\mathbf{q}^d(t) - \mathbf{q}(t)]\}$$
$$+ \ \mathbf{h}_a(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{c}_a(\mathbf{q}) \tag{5.3-66}$$

If $\mathbf{D}_a(\mathbf{q})$, $\mathbf{h}_a(\mathbf{q}, \dot{\mathbf{q}})$, $\mathbf{c}_a(\mathbf{q})$ are equal to $\mathbf{D}(\mathbf{q})$, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{c}(\mathbf{q})$, respectively, then Eq. (5.3-66) reduces to

$$\mathbf{D}(\mathbf{q})[\ddot{\mathbf{e}}(t) + \mathbf{K}_v\dot{\mathbf{e}}(t) + \mathbf{K}_p\mathbf{e}(t)] = 0 \tag{5.3-67}$$

where $\mathbf{e}(t) \triangleq \mathbf{q}^d(t) - \mathbf{q}(t)$ and $\dot{\mathbf{e}}(t) \triangleq \dot{\mathbf{q}}^d(t) - \dot{\mathbf{q}}(t)$.

Since $\mathbf{D}(\mathbf{q})$ is always nonsingular, $\mathbf{K}_p$ and $\mathbf{K}_v$ can be chosen appropriately so the characteristic roots of Eq. (5.3-67) have negative real parts, then the position error vector $\mathbf{e}(t)$ approaches zero asymptotically.

The computation of the joint torques based on the complete L-E equations of motion [Eq. (5.3-65)] is very inefficient. As a result, Paul [1972] concluded that real-time closed-loop digital control is impossible or very difficult. Because of this reason, it is customary to simplify Eq. (5.3-65) by neglecting the velocity-related coupling term $\mathbf{h}_a(\mathbf{q}, \dot{\mathbf{q}})$ and the off-diagonal elements of the acceleration-related matrix $\mathbf{D}_a(\mathbf{q})$. In this case, the structure of the control law has the form

$$\tau(t) = \operatorname{diag}[\mathbf{D}_a(\mathbf{q})]\{\ddot{\mathbf{q}}^d(t) + \mathbf{K}_v[\dot{\mathbf{q}}^d(t) - \dot{\mathbf{q}}(t)] + \mathbf{K}_p[\mathbf{q}^d(t) - \mathbf{q}(t)]\}$$
$$+ \ \mathbf{c}_a(\mathbf{q}) \tag{5.3-68}$$

A computer simulation study had been conducted to which showed that these terms cannot be neglected when the robot arm is moving at high speeds (Paul [1972]).

An analogous control law in the joint-variable space can be derived from the N-E equations of motion to servo a robot arm. The control law is computed recursively using the N-E equations of motion. The recursive control law can be obtained by substituting $\ddot{q}_i(t)$ into the N-E equations of motion to obtain the necessary joint torque for each actuator:

$$\ddot{q}_i(t) = \ddot{q}_i^d(t) + \sum_{j=1}^{n} K_v^{ij}[\dot{q}_j^d(t) - \dot{q}_j(t)] + \sum_{j=1}^{n} K_p^{ij}[q_j^d(t) - q_j(t)] \quad (5.3\text{-}69)$$

where K_v^{ij} and K_p^{ij} are the derivative and position feedback gains for joint i respectively and $e_j(t) = q_j^d(t) - q_j(t)$ is the position error for joint j. The physical interpretation of putting Eq. (5.3-69) into the N-E recursive equations can be viewed as follows:

1. The first term will generate the desired torque for each joint if there is no modeling error and the physical system parameters are known. However, there are errors due to backlash, gear friction, uncertainty about the inertia parameters, and time delay in the servo loop so that deviation from the desired joint trajectory will be inevitable.
2. The remaining terms in the N-E equations of motion will generate the correction torque to compensate for small deviations from the desired joint trajectory.

The above recursive control law is a proportional plus derivative control and has the effect of compensating for inertial loading, coupling effects, and gravity loading of the links. In order to achieve a critically damped system for each joint subsystem (which in turn loosely implies that the whole system behaves as a critically damped system), the feedback gain matrices $\mathbf{K}_p$ and $\mathbf{K}_v$ (diagonal matrices) can be chosen as discussed in Sec. 5.3.3, or as in Paul [1981] or Luh [1983*b*].

In summary, the computed torque technique is a feedforward compensation control. Based on complete L-E equations of motion, the joint torques can be computed in $O(n^4)$ time. The analogous control law derived from the N-E equations of motion can be computed in $O(n)$ time. One of the main drawbacks of this control technique is that the convergence of the position error vector depends on the dynamic coefficients of $\mathbf{D}(\mathbf{q})$, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{c}(\mathbf{q})$ in the equations of motion.

5.3.5 Compensation of Digitally Controlled Systems

In a sampled-data control system, time is normalized to the sampling period Δt; i.e., velocity is expressed as radians per Δt rather than radians per second. This has the effect of scaling the link equivalent inertia up by f_s^2, where f_s is the sampling frequency ($f_s = 1/\Delta t$).

It is typical to use 60-Hz sampling frequency (16-msec sampling period) because of its general availability and because the mechanical resonant frequency of most manipulators is around 5 to 10 Hz. Although the Nyquist sampling theorem indicates that, if the sampling rate is at least twice the cutoff frequency of the system, one should be able to recover the signal, the sampling rate for a continuous time system is more stringent than that. To minimize any deterioration of the controller due to sampling, the rate of sampling must be much greater than the natural frequency of the arm (inversely, the sampling period must be much less than the smallest time constant of the arm). Thus, to minimize the effect of sampling, usually 20 times the cutoff frequency is chosen. That is,

$$\Delta t = \frac{1}{20\,\omega_n/2\pi} = \frac{1}{20 f_n} \tag{5.3-70}$$

5.3.6 Voltage-Torque Conversion

Torque in an armature-controlled dc motor is theoretically a linear function of the armature voltage. However, due to bearing friction at low torques and saturation characteristics at high torques, the actual voltage-torque curves are not linear. For these reasons, a computer conversion of computed torque to required input voltage is usually accomplished via lookup tables or calculation from piecewise linear approximation formulas. The output voltage is usually a constant value and the voltage pulse width varies. A typical voltage-torque curve is shown in Fig. 5.9, where V_ϕ is the motor drive at which the joint will move at constant velocity exerting zero force in the direction of motion, and F_ϕ is the force/torque that the joint will exert at drive level V_ϕ with a negative velocity. The slopes and slope differences are obtained from the experimental curves.

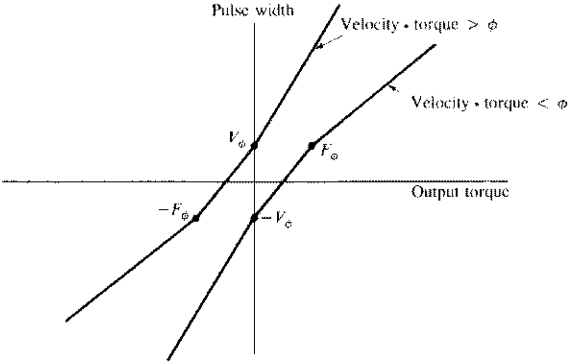

Figure 5.9 Voltage-torque conversion curve.

5.4 NEAR-MINIMUM-TIME CONTROL

For most manufacturing tasks, it is desirable to move a manipulator at its highest speed to minimize the task cycle time. This prompted Kahn and Roth [1971] to investigate the time-optimal control problem for mechanical manipulators. The objective of minimum-time control is to transfer the end-effector of a manipulator from an initial position to a specified desired position in minimum time.

Let us briefly discuss the basics of time-optimal control for a six-link manipulator. The state space representation of the equations of motion of a six-link robot can be formulated from the L-E equations of motion. Let us define a $2n$-dimensional state vector of a manipulator as

$$\mathbf{x}^T(t) = [\mathbf{q}^T(t), \dot{\mathbf{q}}^T(t)] = [q_1(t), \ldots, q_n(t), \dot{q}_1(t), \ldots, \dot{q}_n(t)]$$
$$\triangleq [\mathbf{x}_1^T(t), \mathbf{x}_2^T(t)] \triangleq [x_1(t), x_2(t), \ldots, x_{2n}(t)] \quad (5.4\text{-}1)$$

and an n-dimensional input vector as

$$\mathbf{u}^T(t) = [\tau_1(t), \tau_2(t), \ldots, \tau_n(t)] \quad (5.4\text{-}2)$$

The L-E equations of motion can be expressed in state space representation as

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t)] \quad (5.4\text{-}3)$$

where $\mathbf{f}(\cdot)$ is a $2n \times 1$ continuously differentiable vector-valued function. Since $\mathbf{D}(\mathbf{q})$ is always nonsingular, the above equation can be expressed as

$$\dot{\mathbf{x}}_1(t) = \mathbf{x}_2(t)$$

and
$$\dot{\mathbf{x}}_2(t) = \mathbf{f}_2[\mathbf{x}(t)] + \mathbf{b}[\mathbf{x}_1(t)]\mathbf{u}(t) \quad (5.4\text{-}4)$$

where $\mathbf{f}_2(\mathbf{x})$ is an $n \times 1$ vector-valued function,

$$\mathbf{f}_2(\mathbf{x}) \equiv -\mathbf{D}^{-1}(\mathbf{x}_1)[\mathbf{h}(\mathbf{x}_1, \mathbf{x}_2) + \mathbf{c}(\mathbf{x}_1)] \quad (5.4\text{-}5)$$

and it can be shown that $\mathbf{b}(\mathbf{x}_1)$ is equivalent to the matrix $\mathbf{D}^{-1}(\mathbf{x}_1)$.

At the initial time $t = t_0$, the system is assumed to be in the initial state $\mathbf{x}(t_0) = \mathbf{x}_0$, and at the final minimum time $t = t_f$ the system is required to be in the desired final state $\mathbf{x}(t_f) = \mathbf{x}_f$. Furthermore, the admissible controls of the system are assumed to be bounded and satisfy the constraints,

$$|u_i| \leqslant (u_i)_{\max} \qquad \text{for all } t \quad (5.4\text{-}6)$$

Then the time-optimal control problem is to find an admissible control which transfers the system from the initial state $\mathbf{x}_0$ to the final state $\mathbf{x}_f$, while minimizing the performance index in Eq. (5.4-7) and subject to the constraints of Eq. (5.4-3),

$$J = \int_{t_0}^{t_f} dt = t_f - t_0 \quad (5.4\text{-}7)$$

Using the Pontryagin minimum principle (Kirk [1970]), an optimal control which minimizes the above functional J must minimize the hamiltonian. In terms of the optimal state vector $\mathbf{x}^*(t)$, the optimal control vector $\mathbf{v}^*(t)$, the optimal adjoint variables $\mathbf{p}^*(t)$, and the hamiltonian function,

$$H(\mathbf{x},\ \mathbf{p},\ \mathbf{v}) = \mathbf{p}^T\mathbf{f}(\mathbf{x},\ \mathbf{v}) + 1 \tag{5.4-8}$$

the necessary conditions for $\mathbf{v}^*(t)$ to be an optimal control are

$$\dot{\mathbf{x}}^*(t) = \frac{\partial H(\mathbf{x}^*,\ \mathbf{p}^*,\ \mathbf{u}^*)}{\partial \mathbf{p}} \qquad \text{for all } t \in [t_0,\ t_f] \tag{5.4-9}$$

$$\dot{\mathbf{p}}^*(t) = -\frac{\partial H(\mathbf{x}^*,\ \mathbf{p}^*,\ \mathbf{u}^*)}{\partial \mathbf{x}} \qquad \text{for all } t \in [t_0,\ t_f] \tag{5.4-10}$$

and $$H(\mathbf{x}^*,\ \mathbf{p}^*,\ \mathbf{u}^*) \leqslant H(\mathbf{x}^*,\ \mathbf{p}^*,\ \mathbf{u}) \qquad \text{for all } t \in [t_0,\ t_f] \tag{5.4-11}$$

and for all admissible controls. Obtaining $\mathbf{v}^*(t)$ from Eqs. (5.4-8) to (5.4-11), the optimization problem reduces to a two point boundary value problem with boundary conditions on the state $\mathbf{x}(t)$ at the initial and final times. Due to the nonlinearity of the equations of motion, a numerical solution is usually the only approach to this problem. However, the numerical solution only computes the control function (open-loop control) and does not accommodate any system disturbances. In addition, the solution is optimal for the special initial and final conditions. Hence, the computations of the optimal control have to be performed for each manipulator motion. Furthermore, in practice, the numerical procedures do not provide an acceptable solution for the control of mechanical manipulators. Therefore, as an alternative to the numerical solution, Kahn and Roth [1971] proposed an approximation to the optimal control which results in a near-minimum-time control.

The suboptimal feedback control is obtained by approximating the nonlinear system [Eq. (5.4-4)] by a linear system and analytically finding an optimal control for the linear system. The linear system is obtained by a change of variables followed by linearization of the equations of motion. A transformation is used to decouple the controls in the linearized system. Defining a new set of dependent variables, $\xi_i(t)$, $i = 1, 2, \ldots, 2n$, the equations of motion can be transformed, using the new state variables, to

$$\xi_i(t) = x_i(t) - x_i(t_f) \qquad i = 1, 2, \ldots, n$$

and $$\xi_i(t) = x_i(t) \qquad i = n+1, \ldots, 2n \tag{5.4-12}$$

The first $n\xi_i(t)$ is the error of the angular position, and the second $n\xi_i(t)$ is the error of the rate of the angular position. Because of this change of variables, the control problem becomes one of moving the system from an initial state $\xi(t_0)$ to the origin of the ξ space.

In order to obtain the linearized system, Eq. (5.4-12) is substituted into Eq. (5.4-4) and a Taylor series expansion is used to linearize the system about the origin of the ξ space. In addition, all sine and cosine functions of ξ_i are replaced by their series representations. As a result, the linearized equations of motion are

$$\dot{\xi}(t) = \mathbf{A}\xi(t) + \mathbf{B}\mathbf{v}(t) \tag{5.4-13}$$

where $\xi^T(t) = (\xi_1, \xi_2, \ldots, \xi_n)$ and $\mathbf{v}(t)$ is related to $\mathbf{u}(t)$ by $\mathbf{v}(t) = \mathbf{u}(t) + \mathbf{c}$, where the vector $\mathbf{c}$ contains the steady-state torques due to gravity at the final state. Although Eq. (5.4-13) is linear, the control functions $\mathbf{v}(t)$ are coupled. By properly selecting a set of basis vectors from the linearly independent columns of the controllability matrices of $\mathbf{A}$ and $\mathbf{B}$ to decouple the control function, a new set of equations with no coupling in control variables can be obtained:

$$\dot{\zeta}(t) = \overline{\mathbf{A}}\zeta(t) + \overline{\mathbf{B}}\mathbf{v}(t) \tag{5.4-14}$$

Using a three-link manipulator as an example and applying the above equations to it, we can obtain a three double-integrator system with unsymmetric bounds on controls:

$$\begin{aligned} \dot{\zeta}_{2i-1}(t) &= v_i \\ \dot{\zeta}_{2i}(t) &= \zeta_{2i-1} \qquad i = 1, 2, 3 \end{aligned} \tag{5.4-15}$$

where $v_i^- \leqslant v_i \leqslant v_i^+$ and

$$\begin{aligned} v_i^+ &= (u_i)_{\max} + c_i \\ v_i^- &= -(u_i)_{\max} + c_i \end{aligned} \tag{5.4-16}$$

where c_i is the ith element of vector $\mathbf{c}$.

From this point on, a solution to the time-optimal control and switching surfaces† problem can be obtained by the usual procedures. The linearized and decoupled suboptimal control [Eqs. (5.4-15) and (5.4-16)] generally results in response times and trajectories which are reasonably close to the time-optimal solutions. However, this control method is usually too complex to be used for manipulators with 4 or more degrees of freedom and it neglects the effect of unknown external loads.

† Recall that time-optimal controls are piecewise constant functions of time, and thus we are interested in regions of the state space over which the control is constant. These regions are separated by curves in two-dimensional space, by surfaces in three-dimensional space, and by hypersurfaces in n-dimensional space. These separating surfaces are called switching curves, switching surfaces, and switching hypersurfaces, respectively.

5.5 VARIABLE STRUCTURE CONTROL

In 1978, Young [1978] proposed to use the theory of variable structure systems for the control of manipulators. Variable structure systems (VSS) are a class of systems with discontinuous feedback control. For the last 20 years, the theory of variable structure systems has found numerous applications in control of various processes in the steel, chemical, and aerospace industries. The main feature of VSS is that it has the so-called *sliding mode* on the switching surface. Within the sliding mode, the system remains insensitive to parameter variations and disturbances and its trajectories lie in the switching surface. It is this insensitivity property of VSS that enables us to eliminate the interactions among the joints of a manipulator. The sliding phenomena do not depend on the system parameters and have a stable property. Hence, the theory of VSS can be used to design a variable structure controller (VSC) which induces the sliding mode and in which lie the robot arm's trajectories. Such design of the variable structure controller does not require accurate dynamic modeling of the manipulator; the bounds of the model parameters are sufficient to construct the controller. Variable structure control differs from time-optimal control in the sense that the variable structure controller induces the sliding mode in which the trajectories of the system lie. Furthermore, the system is insensitive to system parameter variations in the sliding mode.

Let us consider the variable structure control for a six-link manipulator. From Eq. (5.4-1), defining the state vector $\mathbf{x}^T(t)$ as

$$\begin{aligned}\mathbf{x}^T &= (q_1, \ldots, q_6, \dot{q}_1, \ldots, \dot{q}_6)\\ &= (p_1, \ldots, p_6, v_1, \ldots, v_6)\\ &= (\mathbf{p}^T, \mathbf{v}^T) \qquad (5.5\text{-}1)\end{aligned}$$

and introducing the position error vector $\mathbf{e}_1(t) = \mathbf{p}(t) - \mathbf{p}^d$ and the velocity error vector $\mathbf{e}_2(t) = \mathbf{v}(t)$ (with $\mathbf{v}^d = 0$), we have changed the tracking problem to a regulator problem. The error equations of the system become

$$\dot{\mathbf{e}}_1(t) = \mathbf{v}(t)$$

and $$\dot{\mathbf{v}}(t) = \mathbf{f}_2(\mathbf{e}_1 + \mathbf{p}^d, \mathbf{v}) + \mathbf{b}(\mathbf{e}_1 + \mathbf{p}^d)\mathbf{u}(t) \qquad (5.5\text{-}2)$$

where $\mathbf{f}_2(\cdot)$ and $\mathbf{b}(\cdot)$ are defined in Eq. (5.4-5). For the regulator system problem in Eq. (5.5-2), a variable structure control $\mathbf{u}(\mathbf{p}, \mathbf{v})$ can be constructed as

$$u_i(\mathbf{p}, \mathbf{v}) = \begin{cases} u_i^+ (\mathbf{p}, \mathbf{v}) & \text{if } s_i(e_i, v_i) > 0 \\ u_i^- (\mathbf{p}, \mathbf{v}) & \text{if } s_i(e_i, v_i) < 0 \end{cases} \qquad i = 1, \ldots, 6 \qquad (5.5\text{-}3)$$

where $s_i(e_i, v_i)$ are the switching surfaces found to be

$$s_i(e_i, v_i) = c_i e_i + v_i \qquad c_i > 0 \qquad i = 1, \ldots, 6 \tag{5.5-4}$$

and the synthesis of the control reduces to choosing the feedback controls as in Eq. (5.5-3) so that the sliding mode occurs on the intersection of the switching planes. By solving the algebraic equations of the switching planes,

$$\dot{s}_i(e_i, v_i) = 0 \qquad i=1, \ldots, 6 \tag{5.5-5}$$

a unique control exists and is found to be

$$\mathbf{u}_{eq} = -\mathbf{D}(\mathbf{p})(\mathbf{f}(\mathbf{p}, \mathbf{v}) + \mathbf{C}\mathbf{v}) \tag{5.5-6}$$

where $\mathbf{C} \equiv \text{diag}\,[c_1, c_2, \ldots, c_6]$. Then, the sliding mode is obtained from Eq. (5.5-4) as

$$\dot{e}_i = -c_i e_i \qquad i = 1, \ldots, 6 \tag{5.5-7}$$

The above equation represents six uncoupled first-order linear systems, each representing 1 degree of freedom of the manipulator when the system is in the sliding mode. As we can see, the controller [Eq. (5.5-3)] forces the manipulator into the sliding mode and the interactions among the joints are completely eliminated. When in the sliding mode, the controller [Eq. (5.5-6)] is used to control the manipulator. The dynamics of the manipulator in the sliding mode depend only on the design parameters c_i. With the choice of $c_i > 0$, we can obtain the asymptotic stability of the system in the sliding mode and make a speed adjustment of the motion in sliding mode by varying the parameters c_i.

In summary, the variable structure control eliminates the nonlinear interactions among the joints by forcing the system into the sliding mode. However, the controller produces a discontinuous feedback control signal that change signs rapidly. The effects of such control signals on the physical control device of the manipulator (i.e., chattering) should be taken into consideration for any applications to robot arm control. A more detailed discussion of designing a multi-input controller for a VSS can be found in Young [1978].

5.6 NONLINEAR DECOUPLED FEEDBACK CONTROL

There is a substantial body of nonlinear control theory which allows one to design a near-optimal control strategy for mechanical manipulators. Most of the existing robot control algorithms emphasize nonlinear compensations of the interactions among the links, e.g., the computed torque technique. Hemami and Camana [1976] applied the nonlinear feedback control technique to a simple locomotion system which has a particular class of nonlinearity (sine, cosine, and polynomial) and obtained decoupled subsystems, postural stability, and desired periodic trajectories. Their approach is different from the method of linear system decoupling where the system to be decoupled must be linear. Saridis and Lee [1979] proposed an iterative algorithm for sequential improvement of a nonlinear suboptimal control law. It provides an approximate optimal control for a manipulator. To achieve such a high quality of control, this method also requires a considerable amount of computational time. In this section, we shall briefly describe the general nonlinear decoupling theory (Falb and Wolovich [1967], Freund [1982]) which will be utilized together with the Newton-Euler equations of motion to compute a nonlinear decoupled controller for robot manipulators.

Given a general nonlinear system as in

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u}(t)$$

and

$$\mathbf{y}(t) = \mathbf{C}(\mathbf{x}) \tag{5.6-1}$$

where $\mathbf{x}(t)$ is an n-dimensional vector, $\mathbf{u}(t)$ and $\mathbf{y}(t)$ are m-dimensional vectors, and $\mathbf{A}(\mathbf{x})$, $\mathbf{B}(\mathbf{x})$, and $\mathbf{C}(\mathbf{x})$ are matrices of compatible order. Let us define a nonlinear operator N_A^K as

$$N_A^K C_i(\mathbf{x}) = \left[\frac{\partial}{\partial \mathbf{x}} N_A^{K-1} C_i(\mathbf{x}) \right] \mathbf{A}(\mathbf{x}) \qquad \begin{array}{l} K = 1, 2, \ldots, n \\ i = 1, 2, \ldots, m \end{array} \tag{5.6-2}$$

where $C_i(\mathbf{x})$ is the ith component of $\mathbf{C}(\mathbf{x})$ and $N_A^0 C_i(\mathbf{x}) = C_i(\mathbf{x})$. Also, let us define the differential order d_i of the nonlinear system as

$$d_i = \min \left\{ j: \left[\frac{\partial}{\partial \mathbf{x}} N_A^{j-1} C_i(\mathbf{x}) \right] \mathbf{B}(\mathbf{x}) \neq 0,\ j = 1, 2, \ldots, n \right\} \tag{5.6-3}$$

Then, the control objective is to find a feedback decoupled controller $\mathbf{u}(t)$:

$$\mathbf{u}(t) = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\ \mathbf{w}(t) \tag{5.6-4}$$

where $\mathbf{w}(t)$ is an m-dimensional reference input vector, $\mathbf{F}(\mathbf{x})$ is an $m \times 1$ feedback vector for decoupling and pole assignment, and $\mathbf{G}(\mathbf{x})$ is an $m \times m$ input gain matrix so that the overall system has a decoupled input-output relationship.

Substituting $\mathbf{u}(t)$ from Eq. (5.6-4) into the system equation of Eq. (5.6-1) results in the following expressions:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{F}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{G}(\mathbf{x})\mathbf{w}(t)$$

and

$$\mathbf{y}(t) = \mathbf{C}(\mathbf{x}) \tag{5.6-5}$$

In order to obtain the decoupled input-output relationships in the above system, $\mathbf{F}(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ are chosen, respectively, as follows:

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_1^*(\mathbf{x}) + \mathbf{F}_2^*(\mathbf{x}) \tag{5.6-6}$$

where

$$\mathbf{F}_1^*(\mathbf{x}) = -\mathbf{D}^{*-1}(\mathbf{x})\mathbf{C}^*(\mathbf{x})$$

$$\mathbf{F}_2^*(\mathbf{x}) = -\mathbf{D}^{*-1}(\mathbf{x})\mathbf{M}^*(\mathbf{x})$$

and

$$\mathbf{G}(\mathbf{x}) = \mathbf{D}^{*-1}(\mathbf{x})\,\mathbf{\Lambda}$$

$\mathbf{F}_1^*(\mathbf{x})$ represents the state feedback that yields decoupling, while $\mathbf{F}_2^*(\mathbf{x})$ performs the control part with arbitrary pole assignment. The input gain of the decoupled part can be chosen by $\mathbf{G}(\mathbf{x})$, and $\mathbf{D}^*(\mathbf{x})$ is an $m \times m$ matrix whose ith row is given by

$$\mathbf{D}_i^*(\mathbf{x}) = \left[\frac{\partial}{\partial \mathbf{x}} N_A^{d_i-1} C_i(\mathbf{x})\right] \mathbf{B}(\mathbf{x}) \qquad \text{for } d_i \neq 0 \tag{5.6-7}$$

$\mathbf{C}^*(\mathbf{x})$ is an m-dimensional vector whose ith component is given by

$$C_i^*(\mathbf{x}) = N_A^{d_i} C_i(\mathbf{x}) \tag{5.6-8}$$

$\mathbf{M}^*(\mathbf{x})$ is an m-dimensional vector whose ith component is given by

$$M_i^*(\mathbf{x}) = \sum_{K=0}^{d_i-1} \alpha_{K,i} N_A^K C_i(\mathbf{x}) \qquad \text{for } d_i \neq 0 \tag{5.6-9}$$

and $\mathbf{\Lambda}$ is a diagonal matrix whose elements are constant values λ_i for $i = 1, 2, \ldots, m$. Then, the system in Eq. (5.6-1) can be represented in terms of

$$\mathbf{y}^*(t) = \mathbf{C}^*(\mathbf{x}) + \mathbf{D}^*(\mathbf{x})\,\mathbf{u}(t) \tag{5.6-10}$$

where $\mathbf{y}^*(t)$ is an output vector whose ith component is $y_i^{(d_i)}(t)$. That is,

$$y_i^{(d_i)}(t) = C_i^*(\mathbf{x}) + \mathbf{D}_i^*\mathbf{u}(t) \tag{5.6-11}$$

Utilizing Eq. (5.6-4) and Eqs. (5.6-6) to (5.6-11), we obtain

$$y_i^{(d_i)}(t) + \alpha_{d_i-1,i}\, y_i^{(d_i-1)}(t) + \cdots + \alpha_{0,i}\, y_i(t) = \lambda_i \omega_i(t) \tag{5.6-12}$$

where $\alpha_{K,i}$ and λ_i are arbitrary scalars.

To show that the ith component of $\mathbf{y}^*(t)$ has the form of Eq. (5.6-11), let us assume that $d_i = 1$. Then, $y_i(t) = C_i(\mathbf{x})$ and, by differentiating it successively, we have

$$\begin{aligned}
y_i^{(1)}(t) &= \dot{y}_i(t) = \frac{\partial C_i(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}}(t) \\
&= \frac{\partial C_i(\mathbf{x})}{\partial \mathbf{x}} [\mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{F}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{G}(\mathbf{x})\mathbf{w}(t)] \\
&= N_{A+BF}^1 C_i(\mathbf{x}) + \frac{\partial C_i(\mathbf{x})}{\partial \mathbf{x}} [\mathbf{B}(\mathbf{x})\mathbf{G}(\mathbf{x})\mathbf{w}(t)]
\end{aligned}$$

Using the identity, $N_{A+BF}^{d_i} C_i(\mathbf{x}) = N_A^{d_i} C_i(\mathbf{x}) + [\partial/\partial\mathbf{x} N_A^{d_i-1} C_i(\mathbf{x})]\mathbf{B}(\mathbf{x})\mathbf{F}(\mathbf{x})$, $y_i^{(1)}(t)$ can be written as

$$y_i^{(1)}(t) = N_A^1 C_i(\mathbf{x}) + \left[\frac{\partial}{\partial \mathbf{x}} C_i(\mathbf{x})\right] \mathbf{B}(\mathbf{x})\mathbf{F}(\mathbf{x}) + \frac{\partial C_i(\mathbf{x})}{\partial \mathbf{x}} [\mathbf{B}(\mathbf{x})\mathbf{G}(\mathbf{x})\mathbf{w}(t)]$$

Using Eqs. (5.6-4) and (5.6-7), it becomes

$$y_i^{(1)}(t) = C_i^*(\mathbf{x}) + \mathbf{D}_i^*(\mathbf{x})\mathbf{u}(t)$$

Similar comments hold for $d_i = 2, 3, \ldots$ to yield Eq. (5.6-11). Thus, the resultant system has decoupled input-output relations and becomes a time-invariant second-order system which can be used to model each joint of the robot arm.

As discussed in Chap. 3, the Lagrange-Euler equations of motion of a six-link robot can be written as

$$\begin{bmatrix} D_{11} & \cdots & D_{16} \\ \vdots & & \vdots \\ D_{16} & \cdots & D_{66} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1(t) \\ \vdots \\ \ddot{\theta}_6(t) \end{bmatrix} + \begin{bmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ \vdots \\ h_6(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{bmatrix} + \begin{bmatrix} c_1(\boldsymbol{\theta}) \\ \vdots \\ c_6(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} u_1(t) \\ \vdots \\ u_6(t) \end{bmatrix} \tag{5.6-13}$$

which can be rewritten in vector-matrix notation as

$$\mathbf{D}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) = \mathbf{u}(t) \tag{5.6-14}$$

where $\mathbf{u}(t)$ is a 6×1 applied torque vector for joint actuators, $\boldsymbol{\theta}(t)$ is the angular positions, $\dot{\boldsymbol{\theta}}(t)$ is the angular velocities, $\ddot{\boldsymbol{\theta}}(t)$ is a 6×1 acceleration vector, $\mathbf{c}(\boldsymbol{\theta})$ is a 6×1 gravitational force vector, $\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is a 6×1 Coriolis and centrifugal force vector, and $\mathbf{D}(\boldsymbol{\theta})$ is a 6×6 acceleration-related matrix. Since $\mathbf{D}(\boldsymbol{\theta})$ is always nonsingular, the above equation can be rewritten as

$$\ddot{\boldsymbol{\theta}}(t) = -\mathbf{D}^{-1}(\boldsymbol{\theta})[\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta})] + \mathbf{D}^{-1}(\boldsymbol{\theta})\mathbf{u}(t) \tag{5.6-15}$$

or, explicitly,

$$\ddot{\boldsymbol{\theta}}(t) = -\begin{bmatrix} D_{11} & \cdots & D_{16} \\ \vdots & & \vdots \\ D_{16} & \cdots & D_{66} \end{bmatrix}^{-1} \begin{bmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + c_1(\boldsymbol{\theta}) \\ \vdots \\ h_6(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + c_6(\boldsymbol{\theta}) \end{bmatrix} \tag{5.6-16}$$

$$+ \begin{bmatrix} D_{11} & & D_{16} \\ \cdot \cdots \cdots \cdots & & \cdot \\ \cdot & & \cdot \\ \cdot \cdots \cdots \cdots & & \cdot \\ D_{16} & & D_{66} \end{bmatrix}^{-1} \begin{bmatrix} u_1(t) \\ \vdots \\ u_6(t) \end{bmatrix}$$

The above dynamic model consists of second-order differential equations for each joint variable; hence, $d_i = 2$. Treating each joint variable $\theta_i(t)$ as an output variable, the above equation can be related to Eq. (5.6-11) as

$$\begin{aligned} y_i^{(2)}(t) = \ddot{y}_i(t) &= -[\mathbf{D}^{-1}(\boldsymbol{\theta})]_i[\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta})] + [\mathbf{D}^{-1}(\boldsymbol{\theta})]_i\mathbf{u}(t) \\ &= C_i^*(\mathbf{x}) + \mathbf{D}_i^*(\mathbf{x})\mathbf{u}(t) \end{aligned} \tag{5.6-17}$$

where

$$C_i^*(\mathbf{x}) = -[\mathbf{D}^{-1}(\boldsymbol{\theta})]_i[\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta})] \tag{5.6-18}$$

$$\mathbf{x}^T(t) = [\boldsymbol{\theta}^T(t), \dot{\boldsymbol{\theta}}^T(t)]$$

and

$$\mathbf{D}_i^*(\mathbf{x}) = [\mathbf{D}^{-1}(\boldsymbol{\theta})]_i \tag{5.6-19}$$

and $[\mathbf{D}^{-1}(\boldsymbol{\theta})]_i$ is the ith row of the $\mathbf{D}^{-1}(\boldsymbol{\theta})$ matrix. Thus, the controller $\mathbf{u}(t)$ for the decoupled system [Eq. (5.6-5)] must be

$$\begin{aligned} \mathbf{u}(t) &= -\mathbf{D}^{*-1}(\mathbf{x})[\mathbf{C}^*(\mathbf{x}) + \mathbf{M}^*(\mathbf{x}) - \boldsymbol{\Lambda}\,\mathbf{w}(t)] \\ &= -\mathbf{D}(\boldsymbol{\theta})\,\{-\mathbf{D}^{-1}(\boldsymbol{\theta})[\mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta})] + \mathbf{M}^*(\mathbf{x}) - \boldsymbol{\Lambda}\,\mathbf{w}(t)\} \\ &= \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) - \mathbf{D}(\boldsymbol{\theta})[\mathbf{M}^*(\mathbf{x}) - \boldsymbol{\Lambda}\,\mathbf{w}(t)] \end{aligned} \tag{5.6-20}$$

Explicitly, for joint i,

$$u_i(t) = h_i(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + c_i(\boldsymbol{\theta}) - [D_{i1} \cdots D_{i6}] \begin{bmatrix} \alpha_{11}\dot{\theta}_1(t) + \alpha_{01}\theta_1(t) - \lambda_1 w_1(t) \\ \vdots \\ \alpha_{16}\dot{\theta}_6(t) + \alpha_{06}\theta_6(t) - \lambda_6 w_6(t) \end{bmatrix} \tag{5.6-21}$$

From the above equation, we note that the controller $u_i(t)$ for joint i depends only on the current dynamic variables and the input $\mathbf{w}(t)$. Substituting $\mathbf{u}(t)$ from Eq. (5.6-20) into Eq. (5.6-14), we have

$$\mathbf{D}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}(t) + \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta})$$

$$= \mathbf{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{c}(\boldsymbol{\theta}) - \mathbf{D}(\boldsymbol{\theta}) \begin{bmatrix} \alpha_{11}\dot{\theta}_1(t) + \alpha_{01}\theta_1(t) - \lambda_1 w_1(t) \\ \vdots \\ \alpha_{16}\dot{\theta}_6(t) + \alpha_{06}\theta_6(t) - \lambda_6 w_6(t) \end{bmatrix} \qquad (5.6\text{-}22)$$

which leads to

$$\mathbf{D}(\boldsymbol{\theta}) \begin{bmatrix} \ddot{\theta}_1(t) + \alpha_{11}\dot{\theta}_1(t) + \alpha_{01}\theta_1(t) - \lambda_1 w_1(t) \\ \vdots \\ \ddot{\theta}_6(t) + \alpha_{16}\dot{\theta}_6(t) + \alpha_{06}\theta_6(t) - \lambda_6 w_6(t) \end{bmatrix} = 0 \qquad (5.6\text{-}23)$$

Since $\mathbf{D}(\boldsymbol{\theta})$ is always nonsingular, the above equation becomes

$$\ddot{\theta}_i(t) + \alpha_{1i}\dot{\theta}_i(t) + \alpha_{0i}\theta_i(t) = \lambda_i \omega_i(t) \qquad i = 1, 2, \ldots, 6 \qquad (5.6\text{-}24)$$

which indicates the final decoupled input-output relationships of the system. It is interesting to note that the parameters α_{1i}, α_{0i}, and λ_i can be selected arbitrarily, provided that the stability criterion is maintained. Hence, the manipulator can be considered as six independent, decoupled, second-order, time-invariant systems and the controller $\mathbf{u}(t)$ [Eq. (5.6-20)] can be computed efficiently based on the manipulator dynamics. An efficient way of computing the controller $\mathbf{u}(t)$ is through the use of the Newton-Euler equations of motion. Hence, to compute the controller $u_i(t)$ for joint i, $\ddot{\theta}_i(t)$ is substituted with $\lambda_i w_i(t) - \alpha_{1i}\dot{\theta}_i(t) - \alpha_{0i}\theta_i(t)$ in the Newton-Euler equations of motion.

5.7 RESOLVED MOTION CONTROL

In the last section, several methods were discussed for controlling a mechanical manipulator in the joint-variable space to follow a joint-interpolated trajectory. In many applications, resolved motion control, which commands the manipulator hand to move in a desired cartesian direction in a coordinated position and rate control, is more appropriate. *Resolved motion* means that the motions of the various joint motors are combined and resolved into separately controllable hand motions along the world coordinate axes. This implies that several joint motors must run simultaneously at different time-varying rates in order to achieve desired coordinated hand motion along any world coordinate axis. This enables the user to specify the direction and speed along any arbitrarily oriented path for the manipulator to follow. This motion control greatly simplifies the specification of the sequence of motions for completing a task because users are usually more adapted to the cartesian coordinate system than the manipulator's joint angle coordinates.

In general, the desired motion of a manipulator is specified in terms of a time-based hand trajectory in cartesian coordinates, while the servo control system requires that the reference inputs be specified in joint coordinates. The mathematical relationship between these two coordinate systems is important in designing efficient control in the cartesian space. We shall briefly describe the basic kinematics theory relating these two coordinate systems for a six-link robot arm that will lead us to understand various important resolved motion control methods.

The location of the manipulator hand with respect to a fixed reference coordinate system can be realized by establishing an orthonormal coordinate frame at the hand (the hand coordinate frame), as shown in Fig. 5.10. The problem of finding the location of the hand is reduced to finding the position and orientation of the hand coordinate frame with respect to the inertial frame of the manipulator. This can be conveniently achieved by a 4 × 4 homogeneous transformation matrix:

$$^{\text{base}}\mathbf{T}_{\text{hand}}(t) = \begin{bmatrix} n_x(t) & s_x(t) & a_x(t) & p_x(t) \\ n_y(t) & s_y(t) & a_y(t) & p_y(t) \\ n_z(t) & s_z(t) & a_z(t) & p_z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n}(t) & \mathbf{s}(t) & \mathbf{a}(t) & \mathbf{p}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7-1}$$

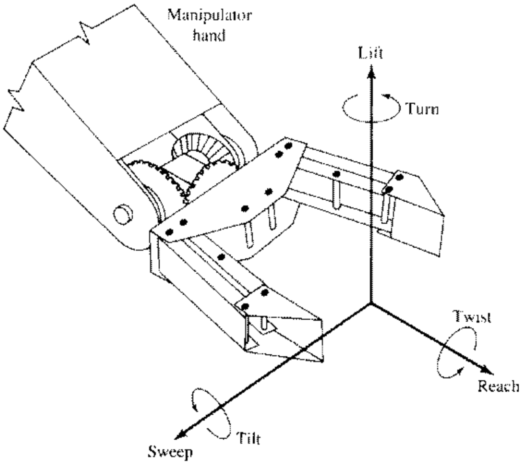

Figure 5.10 The hand coordinate system.

where **p** is the position vector of the hand, and **n**, **s**, **a** are the unit vectors along the principal axes of the coordinate frame describing the orientation of the hand. Instead of using the rotation submatrix [**n**, **s**, **a**] to describe the orientation, we can use three Euler angles, yaw $\alpha(t)$, pitch $\beta(t)$, and roll $\gamma(t)$, which are defined as rotations of the hand coordinate frame about the $\mathbf{x}_0$, $\mathbf{y}_0$, and $\mathbf{z}_0$ of the reference frame, respectively. One can obtain the elements of [**n**, **s**, **a**] from the Euler rotation matrix resulting from a rotation of the α angle about the $\mathbf{x}_0$ axis, then a rotation of the β angle about the $\mathbf{y}_0$ axis, and a rotation of the γ angle about the $\mathbf{z}_0$ axis of the reference frame [Eq. (2.2-19)]. Thus:

$$
{}^{\text{base}}\mathbf{R}_{\text{hand}}(t) = \begin{bmatrix} n_x(t) & s_x(t) & a_x(t) \\ n_y(t) & s_y(t) & a_y(t) \\ n_z(t) & s_z(t) & a_z(t) \end{bmatrix}
$$

$$
= \begin{bmatrix} C\gamma & -S\gamma & 0 \\ S\gamma & C\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\beta & 0 & S\beta \\ 0 & 1 & 0 \\ -S\beta & 0 & C\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix}
$$

$$
= \begin{bmatrix} C\gamma C\beta & -S\gamma C\alpha + C\gamma S\beta S\alpha & S\gamma S\alpha + C\gamma S\beta C\alpha \\ S\gamma C\beta & C\gamma C\alpha + S\gamma S\beta S\alpha & -C\gamma S\alpha + S\gamma S\beta C\alpha \\ -S\beta & C\beta S\alpha & C\beta C\alpha \end{bmatrix} \qquad (5.7\text{-}2)
$$

where $\sin\alpha \equiv S\alpha$, $\cos\alpha \equiv C\alpha$, $\sin\beta \equiv S\beta$, $\cos\beta \equiv C\beta$, $\sin\gamma \equiv S\gamma$, and $\cos\gamma \equiv C\gamma$.

Let us define the position $\mathbf{p}(t)$, Euler angles $\boldsymbol{\Phi}(t)$, linear velocity $\mathbf{v}(t)$, and angular velocity $\boldsymbol{\Omega}(t)$ vectors of the manipulator hand with respect to the reference frame, respectively:

$$
\mathbf{p}(t) \triangleq [p_x(t), p_y(t), p_z(t)]^T \qquad \boldsymbol{\Phi}(t) \triangleq [\alpha(t), \beta(t), \gamma(t)]^T
$$

$$
\mathbf{v}(t) \triangleq [v_x(t), v_y(t), v_z(t)]^T \qquad \boldsymbol{\Omega}(t) \triangleq [\omega_x(t), \omega_y(t), \omega_z(t)]^T \qquad (5.7\text{-}3)
$$

The linear velocity of the hand with respect to the reference frame is equal to the time derivative of the position of the hand:

$$
\mathbf{v}(t) = \frac{d\mathbf{p}(t)}{dt} = \dot{\mathbf{p}}(t) \qquad (5.7\text{-}4)
$$

Since the inverse of a direction cosine matrix is equivalent to its transpose, the instantaneous angular velocities of the hand coordinate frame about the principal axes of the reference frame can be obtained from Eq. (5.7-2):

$$\mathbf{R}\frac{d\mathbf{R}^T}{dt} = -\frac{d\mathbf{R}}{dt}\mathbf{R}^T = -\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -S\beta\dot{\alpha} + \dot{\gamma} & -S\gamma C\beta\dot{\alpha} - C\gamma\dot{\beta} \\ S\beta\dot{\alpha} - \dot{\gamma} & 0 & C\gamma C\beta\dot{\alpha} - S\gamma\dot{\beta} \\ S\gamma C\beta\dot{\alpha} + C\gamma\dot{\beta} & -C\gamma C\beta\dot{\alpha} + S\gamma\dot{\beta} & 0 \end{bmatrix} \qquad (5.7\text{-}5)$$

From the above equation, the relation between the $[\omega_x(t), \omega_y(t), \omega_z(t)]^T$ and $[\dot{\alpha}(t), \dot{\beta}(t), \dot{\gamma}(t)]^T$ can be found by equating the nonzero elements in the matrices:

$$\begin{bmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{bmatrix} = \begin{bmatrix} C\gamma C\beta & -S\gamma & 0 \\ S\gamma C\beta & C\gamma & 0 \\ -S\beta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \\ \dot{\gamma}(t) \end{bmatrix} \qquad (5.7\text{-}6)$$

Its inverse relation can be found easily:

$$\begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \\ \dot{\gamma}(t) \end{bmatrix} = \sec\beta \begin{bmatrix} C\gamma & S\gamma & 0 \\ -S\gamma C\beta & C\gamma C\beta & 0 \\ C\gamma S\beta & S\gamma S\beta & C\beta \end{bmatrix} \begin{bmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{bmatrix} \qquad (5.7\text{-}7)$$

or expressed in matrix-vector form,

$$\dot{\mathbf{\Phi}}(t) \triangleq [\mathbf{S}(\mathbf{\Phi})]\,\mathbf{\Omega}(t) \qquad (5.7\text{-}8)$$

Based on the moving coordinate frame concept, the linear and angular velocities of the hand can be obtained from the velocities of the lower joints:

$$\begin{bmatrix} \mathbf{v}(t) \\ \mathbf{\Omega}(t) \end{bmatrix} = [\mathbf{N}(\mathbf{q})]\dot{\mathbf{q}}(t) = [\mathbf{N}_1(\mathbf{q}), \mathbf{N}_2(\mathbf{q}), \ldots, \mathbf{N}_6(\mathbf{q})]\,\dot{\mathbf{q}}(t) \quad (5.7\text{-}9)$$

where $\dot{\mathbf{q}}(t) = (\dot{q}_1, \ldots, \dot{q}_6)^T$ is the joint velocity vector of the manipulator, and $\mathbf{N}(\mathbf{q})$ is a 6×6 jacobian matrix whose ith column vector $\mathbf{N}_i(\mathbf{q})$ can be found to be (Whitney [1972]):

$$\mathbf{N}_i(\mathbf{q}) = \begin{cases} \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p} - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} & \text{if joint } i \text{ is rotational} \\ \\ \begin{bmatrix} \mathbf{z}_{i-1} \\ 0 \end{bmatrix} & \text{if joint i is translational} \end{cases} \tag{5.7-10}$$

where × indicates the vector cross product, $\mathbf{p}_{i-1}$ is the position of the origin of the $(i-1)$th coordinate frame with respect to the reference frame, $\mathbf{z}_{i-1}$ is the unit vector along the axis of motion of joint i, and $\mathbf{p}$ is the position of the hand with respect to the reference coordinate frame.

If the inverse jacobian matrix exists at $\mathbf{q}(t)$, then the joint velocities $\dot{\mathbf{q}}(t)$ of the manipulator can be computed from the hand velocities using Eq. (5.7-9):

$$\dot{\mathbf{q}}(t) = \mathbf{N}^{-1}(\mathbf{q}) \begin{bmatrix} \mathbf{v}(t) \\ \boldsymbol{\Omega}(t) \end{bmatrix} \tag{5.7-11}$$

Given the desired linear and angular velocities of the hand, this equation computes the joint velocities and indicates the rates at which the joint motors must be maintained in order to achieve a steady hand motion along the desired cartesian direction.

The accelerations of the hand can be obtained by taking the time derivative of the velocity vector in Eq. (5.7-9):

$$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\boldsymbol{\Omega}}(t) \end{bmatrix} = \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}(t) + \mathbf{N}(\mathbf{q})\ddot{\mathbf{q}}(t) \tag{5.7-12}$$

where $\ddot{\mathbf{q}}(t) = [\ddot{q}_1(t), \ldots, \ddot{q}_6(t)]^T$ is the joint acceleration vector of the manipulator. Substituting $\dot{\mathbf{q}}(t)$ from Eq. (5.7-11) into Eq. (5.7-12) gives

$$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\boldsymbol{\Omega}}(t) \end{bmatrix} = \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{N}^{-1}(\mathbf{q}) \begin{bmatrix} \mathbf{v}(t) \\ \boldsymbol{\Omega}(t) \end{bmatrix} + \mathbf{N}(\mathbf{q})\ddot{\mathbf{q}}(t) \tag{5.7-13}$$

and the joint accelerations $\ddot{\mathbf{q}}(t)$ can be computed from the hand velocities and accelerations as

$$\ddot{\mathbf{q}}(t) = \mathbf{N}^{-1}(\mathbf{q}) \begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\boldsymbol{\Omega}}(t) \end{bmatrix} - \mathbf{N}^{-1}(\mathbf{q})\dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{N}^{-1}(\mathbf{q}) \begin{bmatrix} \mathbf{v}(t) \\ \boldsymbol{\Omega}(t) \end{bmatrix} \tag{5.7-14}$$

The above kinematic relations between the joint coordinates and the cartesian coordinates will be used in Sec. 5.7.1 for various resolved motion control methods and in deriving the resolved motion equations of motion of the manipulator hand in cartesian coordinates.

5.7.1 Resolved Motion Rate Control

Resolved motion rate control (RMRC) means that the motions of the various joint motors are combined and run simultaneously at different time-varying rates in order to achieve steady hand motion along any world coordinate axis. The mathematics that relate the world coordinates, such as lift p_x, sweep p_y, reach p_z, yaw α, pitch β, and roll γ to the joint angle coordinate of a six-link manipulator is inherently nonlinear and can be expressed by a nonlinear vector-valued function as

$$\mathbf{x}(t) = \mathbf{f}[\mathbf{q}(t)] \tag{5.7-15}$$

where $\mathbf{f}(\mathbf{q})$ is a 6×1 vector-valued function, and

$$\mathbf{x}(t) = \text{world coordinates} = (p_x, p_y, p_z, \alpha, \beta, \gamma)^T$$

and $$\mathbf{q}(t) = \text{generalized coordinates} = (q_1, q_2, \ldots, q_n)^T$$

The relationship between the linear and angular velocities and the joint velocities of a six-link manipulator is given by Eq. (5.7-9).

For a more general discussion, if we assume that the manipulator has m degrees of freedom while the world coordinates of interest are of dimension n, then the joint angles and the world coordinates are related by a nonlinear function, as in Eq. (5.7-15).

If we differentiate Eq. (5.7-15) with respect to time, we have

$$\frac{d\mathbf{x}(t)}{dt} = \dot{\mathbf{x}}(t) = \mathbf{N}(\mathbf{q})\dot{\mathbf{q}}(t) \tag{5.7-16}$$

where $\mathbf{N}(\mathbf{q})$ is the jacobian matrix with respect to $\mathbf{q}(t)$, that is,

$$N_{ij} = \frac{\partial f_i}{\partial q_j} \qquad 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m \tag{5.7-17}$$

We see that if we work with rate control, the relationship is linear, as indicated by Eq. (5.7-16). When $\mathbf{x}(t)$ and $\mathbf{q}(t)$ are of the same dimension, that is, $m = n$, then the manipulator is nonredundant and the jacobian matrix can be inverted at a particular nonsingular position $\mathbf{q}(t)$:

$$\dot{\mathbf{q}}(t) = \mathbf{N}^{-1}(\mathbf{q})\dot{\mathbf{x}}(t) \tag{5.7-18}$$

From Eq. (5.7-18), given the desired rate along the world coordinates, one can easily find the combination of joint motor rates to achieve the desired hand motion. Various methods of computing the inverse jacobian matrix can be used. A resolved motion rate control block diagram is shown in Fig. 5.11.

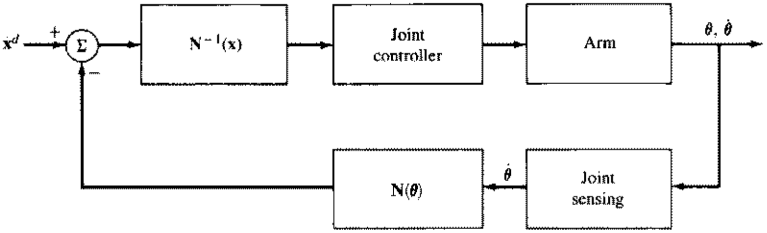

Figure 5.11 The resolved motion rate control block diagram.

If $m > n$, then the manipulator is redundant and the inverse jacobian matrix does not exist. This reduces the problem to finding the generalized inverse of the jacobian matrix. In this case, if the rank of $\mathbf{N}(\mathbf{q})$ is n, then $\dot{\mathbf{q}}(t)$ can be found by minimizing an error criterion formed by adjoining Eq. (5.7-16) with a Lagrange multiplier to a cost criterion, that is,

$$C = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{A}\dot{\mathbf{q}} + \lambda^T[\dot{\mathbf{x}} - \mathbf{N}(\mathbf{q})\dot{\mathbf{q}}] \tag{5.7-19}$$

where λ is a Lagrange multiplier vector, and $\mathbf{A}$ is an $m \times m$ symmetric, positive definite matrix.

Minimizing the cost criterion C with respect to $\dot{\mathbf{q}}(t)$ and λ, we have

$$\dot{\mathbf{q}}(t) = \mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})\lambda \tag{5.7-20}$$

and

$$\dot{\mathbf{x}}(t) = \mathbf{N}(\mathbf{q})\dot{\mathbf{q}}(t) \tag{5.7-21}$$

Substituting $\dot{\mathbf{q}}(t)$ from Eq. (5.7-20) into Eq. (5.7-21), and solving for λ, yields

$$\lambda = [\mathbf{N}(\mathbf{q})\mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})]^{-1}\dot{\mathbf{x}}(t) \tag{5.7-22}$$

Substituting λ into Eq. (5.7-20), we obtain

$$\dot{\mathbf{q}}(t) = \mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})[\mathbf{N}(\mathbf{q})\mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})]^{-1}\dot{\mathbf{x}}(t) \tag{5.7-23}$$

If the matrix $\mathbf{A}$ is an identity matrix, then Eq. (5.7-23) reduces to Eq. (5.7-18).

Quite often, it is of interest to command the hand motion along the hand coordinate system rather than the world coordinate system (see Fig. 5.10). In this case, the desired hand rate motion $\dot{\mathbf{h}}(t)$ along the hand coordinate system is related to the world coordinate motion by

$$\dot{\mathbf{x}}(t) = {}^0\mathbf{R}_h\dot{\mathbf{h}}(t) \tag{5.7-24}$$

where ${}^0\mathbf{R}_h$ is an $n \times 6$ matrix that relates the orientation of the hand coordinate system to the world coordinate system. Given the desired hand rate motion $\dot{\mathbf{h}}(t)$

with respect to the hand coordinate system, and using Eqs. (5.7-23) to (5.7-24), the joint rate $\dot{\mathbf{q}}(t)$ can be computed by:

$$\dot{\mathbf{q}}(t) = \mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})[\mathbf{N}(\mathbf{q})\mathbf{A}^{-1}\mathbf{N}^T(\mathbf{q})]^{-1}\,{}^0\mathbf{R}_h\dot{\mathbf{h}}(t) \tag{5.7-25}$$

In Eqs. (5.7-23) and (5.7-25), the angular position $\mathbf{q}(t)$ depends on time t, so we need to evaluate $\mathbf{N}^{-1}(\mathbf{q})$ at each sampling time t for the calculation of $\dot{\mathbf{q}}(t)$. The added computation in obtaining the inverse jacobian matrix at each sampling time and the singularity problem associated with the matrix inversion are important issues in using this control method.

5.7.2 Resolved Motion Acceleration Control

The resolved motion acceleration control (RMAC) (Luh et al. [1980*b*]) extends the concept of resolved motion rate control to include acceleration control. It presents an alternative position control which deals directly with the position and orientation of the hand of a manipulator. All the feedback control is done at the hand level, and it assumes that the desired accelerations of a preplanned hand motion are specified by the user.

The actual and desired position and orientation of the hand of a manipulator can be represented by 4 × 4 homogeneous transformation matrices, respectively, as

$$\mathbf{H}(t) = \begin{bmatrix} \mathbf{n}(t) & \mathbf{s}(t) & \mathbf{a}(t) & \mathbf{p}(t) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{H}^d(t) = \begin{bmatrix} \mathbf{n}^d(t) & \mathbf{s}^d(t) & \mathbf{a}^d(t) & \mathbf{p}^d(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.7-26}$$

where $\mathbf{n}$, $\mathbf{s}$, $\mathbf{a}$ are the unit vectors along the principal axes $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ of the hand coordinate system, respectively, and $\mathbf{p}(t)$ is the position vector of the hand with respect to the base coordinate system. The orientation submatrix $[\mathbf{n}, \mathbf{s}, \mathbf{a}]$ can be defined in terms of Euler angles of rotation (α, β, γ) with respect to the base coordinate system as in Eq. (5.7-2).

The position error of the hand is defined as the difference between the desired and the actual position of the hand and can be expressed as

$$\mathbf{e}_p(t) = \mathbf{p}^d(t) - \mathbf{p}(t) = \begin{bmatrix} p_x^d(t) - p_x(t) \\ p_y^d(t) - p_y(t) \\ p_z^d(t) - p_z(t) \end{bmatrix} \tag{5.7-27}$$

Similarly, the orientation error is defined by the discrepancies between the desired and actual orientation axes of the hand and can be represented by

$$\mathbf{e}_0(t) = \tfrac{1}{2}[\mathbf{n}(t) \times \mathbf{n}^d + \mathbf{s}(t) \times \mathbf{s}^d + \mathbf{a}(t) \times \mathbf{a}^d] \tag{5.7-28}$$

Thus, control of the manipulator is achieved by reducing these errors of the hand to zero.

Considering a six-link manipulator, we can combine the linear velocities $\mathbf{v}(t)$ and the angular velocities $\boldsymbol{\omega}(t)$ of the hand into a six-dimensional vector as $\dot{\mathbf{x}}(t)$,

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{v}(t) \\ \boldsymbol{\omega}(t) \end{bmatrix} = \mathbf{N}(\mathbf{q})\dot{\mathbf{q}}(t) \qquad (5.7\text{-}29)$$

where $\mathbf{N}(\mathbf{q})$ is a 6×6 matrix as given in Eq. (5.7-10). Equation (5.7-29) is the basis for resolved motion rate control where joint velocities are solved from the hand velocities. If this idea is extended further to solve for the joint accelerations from the hand acceleration $\ddot{\mathbf{x}}(t)$, then the time derivative of $\dot{\mathbf{x}}(t)$ is the hand acceleration

$$\ddot{\mathbf{x}}(t) = \mathbf{N}(\mathbf{q})\ddot{\mathbf{q}}(t) + \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}(t) \qquad (5.7\text{-}30)$$

The closed-loop resolved motion acceleration control is based on the idea of reducing the position and orientation errors of the hand to zero. If the cartesian path for a manipulator is preplanned, then the desired position $\mathbf{p}^d(t)$, the desired velocity $\mathbf{v}^d(t)$, and the desired acceleration $\dot{\mathbf{v}}^d(t)$ of the hand are known with respect to the base coordinate system. In order to reduce the position error, one may apply joint torques and forces to each joint actuator of the manipulator. This essentially makes the actual linear acceleration of the hand, $\dot{\mathbf{v}}(t)$, satisfy the equation

$$\dot{\mathbf{v}}(t) = \dot{\mathbf{v}}^d(t) + k_1[\mathbf{v}^d(t) - \mathbf{v}(t)] + k_2[\mathbf{p}^d(t) - \mathbf{p}(t)] \qquad (5.7\text{-}31)$$

where k_1 and k_2 are scalar constants. Equation (5.7-31) can be rewritten as

$$\ddot{\mathbf{e}}_p(t) + k_1\dot{\mathbf{e}}_p(t) + k_2\mathbf{e}_p(t) = 0 \qquad (5.7\text{-}32)$$

where $\mathbf{e}_p(t) = \mathbf{p}^d(t) - \mathbf{p}(t)$. The input torques and forces must be chosen so as to guarantee the asymptotic convergence of the position error of the hand. This requires that k_1 and k_2 be chosen such that the characteristic roots of Eq. (5.7-32) have negative real parts.

Similarly, to reduce the orientation error of the hand, one has to choose the input torques and forces to the manipulator so that the angular acceleration of the hand satisfies the expression

$$\dot{\boldsymbol{\omega}}(t) = \dot{\boldsymbol{\omega}}^d(t) + k_1[\boldsymbol{\omega}^d(t) - \boldsymbol{\omega}(t)] + k_2\mathbf{e}_0 \qquad (5.7\text{-}33)$$

Let us group $\mathbf{v}^d$ and $\boldsymbol{\omega}^d$ into a six-dimensional vector and the position and orientation errors into an error vector:

$$\dot{\mathbf{x}}^d(t) = \begin{bmatrix} \mathbf{v}^d(t) \\ \boldsymbol{\omega}^d(t) \end{bmatrix} \quad \text{and} \quad \mathbf{e}(t) = \begin{bmatrix} \mathbf{e}_p(t) \\ \mathbf{e}_0(t) \end{bmatrix} \qquad (5.7\text{-}34)$$

Combining Eqs. (5.7-31) and (5.7-33), we have

$$\ddot{\mathbf{x}}(t) = \ddot{\mathbf{x}}^d(t) + k_1[\dot{\mathbf{x}}^d(t) - \dot{\mathbf{x}}(t)] + k_2\mathbf{e}(t) \tag{5.7-35}$$

Substituting Eqs. (5.7-29) and (5.7-30) into Eq. (5.7-35) and solving for $\ddot{\mathbf{q}}(t)$ gives

$$\begin{aligned}\ddot{\mathbf{q}}(t) &= \mathbf{N}^{-1}(\mathbf{q})[\ddot{\mathbf{x}}^d(t) + k_1(\dot{\mathbf{x}}^d(t) - \dot{\mathbf{x}}(t)) + k_2\mathbf{e}(t) - \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}(t)] \\ &= -k_1\dot{\mathbf{q}}(t) + \mathbf{N}^{-1}(\mathbf{q})[\ddot{\mathbf{x}}^d(t) + k_1\dot{\mathbf{x}}^d(t) + k_2\mathbf{e}(t) - \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}(t)]\end{aligned} \tag{5.7-36}$$

Equation (5.7-36) is the basis for the closed-loop resolved acceleration control for manipulators. In order to compute the applied joint torques and forces to each joint actuator of the manipulator, the recursive Newton-Euler equations of motion are used. The joint position $\mathbf{q}(t)$, and joint velocity $\dot{\mathbf{q}}(t)$ are measured from the potentiometers, or optical encoders, of the manipulator. The quantities $\mathbf{v}$, $\boldsymbol{\omega}$, $\mathbf{N}$, $\mathbf{N}^{-1}$, $\dot{\mathbf{N}}$, and $\mathbf{H}(t)$ can be computed from the above equations. These values together with the desired position $\mathbf{p}^d(t)$, desired velocity $\mathbf{v}^d(t)$, and the desired acceleration $\mathbf{v}^d(t)$ of the hand obtained from a planned trajectory can be used to compute the joint acceleration using Eq. (5.7-36). Finally the applied joint torques and forces can be computed recursively from the Newton-Euler equations of motion. As in the case of RMRC, this control method is characterized by extensive computational requirements, singularities associated with the jacobian matrix, and the need to plan a manipulator hand trajectory with acceleration information.

5.7.3 Resolved Motion Force Control

The basic concept of resolved motion force control (RMFC) is to determine the applied torques to the joint actuators in order to perform the cartesian position control of the robot arm. An advantage of RMFC is that the control is not based on the complicated dynamic equations of motion of the manipulator and still has the ability to compensate for changing arm configurations, gravity loading forces on the links, and internal friction. Similar to RMAC, all the control of RMFC is done at the hand level.

The RMFC is based on the relationship between the resolved force vector **F** obtained from a wrist force sensor and the joint torques at the joint actuators. The control technique consists of the cartesian position control and the force convergent control. The position control calculates the desired forces and moments to be applied to the end-effector in order to track a desired cartesian trajectory. The force convergent control determines the necessary joint torques to each actuator so that the end-effector can maintain the desired forces and moments obtained from the position control. A control block diagram of the RMFC is shown in Fig. 5.12.

We shall briefly discuss the mathematics that governs this control technique. A more detailed discussion can be found in Wu and Paul [1982]. The basic control concept of the RMFC is based on the relationship between the resolved

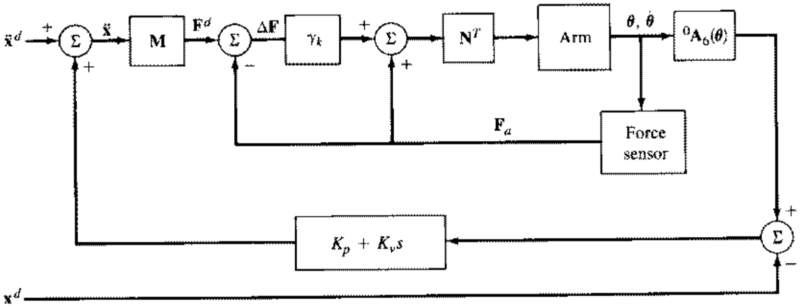

Figure 5.12 Resolved motion force control.

force vector, $\mathbf{F} = (F_x, F_y, F_z, M_x, M_y, M_z)^T$, and the joint torques, $\tau = (\tau_1, \tau_2, \ldots, \tau_n)^T$, which are applied to each joint actuator in order to counterbalance the forces felt at the hand, where $(F_x, F_y, F_z)^T$ and $(M_x, M_y, M_z)^T$ are the cartesian forces and moments in the hand coordinate system, respectively. The underlying relationship between these quantities is

$$\tau(t) = \mathbf{N}^T(\mathbf{q})\mathbf{F}(t) \tag{5.7-37}$$

where $\mathbf{N}$ is the jacobian matrix, as in Eq. (5.7-10).

Since the objective of RMFC is to track the cartesian position of the end-effector, an appropriate time-based position trajectory has to be specified as functions of the arm transformation matrix ${}^0\mathbf{A}_6(t)$, the velocity $(v_x, v_y, v_z)^T$, and the angular velocity $(\omega_x, \omega_y, \omega_z)^T$ about the hand coordinate system. That is, the desired time-varying arm transformation matrix, ${}^0\mathbf{A}_6(t + \Delta t)$, can be represented as

$${}^0\mathbf{A}_6(t + \Delta t) = {}^0\mathbf{A}_6(t) \begin{bmatrix} 1 & -\omega_z(t) & \omega_y(t) & v_x(t) \\ \omega_z(t) & 1 & -\omega_x(t) & v_y(t) \\ -\omega_y(t) & \omega_x(t) & 1 & v_z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \Delta t \tag{5.7-38}$$

then, the desired cartesian velocity $\dot{\mathbf{x}}^d(t) = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T$ can be obtained from the element of the following equation

$$\begin{bmatrix} 1 & -\omega_z(t) & \omega_y(t) & v_x(t) \\ \omega_z(t) & 1 & -\omega_x(t) & v_y(t) \\ -\omega_y(t) & \omega_x(t) & 1 & v_z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{\Delta t}[({}^0\mathbf{A}_6)^{-1}(t)\,{}^0\mathbf{A}_6(t + \Delta t)] \tag{5.7-39}$$

The cartesian velocity error $\dot{\mathbf{x}}^d - \dot{\mathbf{x}}$ can be obtained using the above equation. The velocity error $\dot{\mathbf{x}}^d - \dot{\mathbf{x}}$ used in Eq. (5.7-31) is different from the above velocity error because the above error equation uses the homogeneous transformation matrix method. In Eq. (5.7-31), the velocity error is obtained simply by differentiating $\mathbf{p}^d(t) - \mathbf{p}(t)$.

Similarly, the desired cartesian acceleration $\ddot{\mathbf{x}}^d(t)$ can be obtained as:

$$\ddot{\mathbf{x}}^d(t) = \frac{\dot{\mathbf{x}}^d(t + \Delta t) - \dot{\mathbf{x}}^d(t)}{\Delta t} \tag{5.7-40}$$

Based on the proportional plus derivative control approach, if there is no error in position and velocity of the hand, then we want the actual cartesian acceleration $\ddot{\mathbf{x}}(t)$ to track the desired cartesian acceleration as closely as possible. This can be done by setting the actual cartesian acceleration as

$$\ddot{\mathbf{x}}(t) = \ddot{\mathbf{x}}^d(t) + K_v[\dot{\mathbf{x}}^d(t) - \dot{\mathbf{x}}(t)] + K_p[\mathbf{x}^d(t) - \mathbf{x}(t)] \tag{5.7-41}$$

or

$$\ddot{\mathbf{x}}_e(t) + K_v\dot{\mathbf{x}}_e(t) + K_p\mathbf{x}_e(t) = 0 \tag{5.7-42}$$

By choosing the values of K_v and K_p so that the characteristic roots of Eq. (5.7-42) have negative real parts, $\mathbf{x}(t)$ will converge to $\mathbf{x}^d(t)$ asymptotically.

Based on the above control technique, the desired cartesian forces and moments to correct the position errors can be obtained using Newton's second law:

$$\mathbf{F}^d(t) = \mathbf{M}\ddot{\mathbf{x}}(t) \tag{5.7-43}$$

where $\mathbf{M}$ is the mass matrix with diagonal elements of total mass of the load m and the moments of inertia I_{xx}, I_{yy}, I_{zz} at the principal axes of the load. Then, using the Eq. (5.7-37), the desired cartesian forces $\mathbf{F}^d$ can be resolved into the joint torques:

$$\tau(t) = \mathbf{N}^T(\mathbf{q})\mathbf{F}^d = \mathbf{N}^T(\mathbf{q})\,\mathbf{M}\ddot{\mathbf{x}}(t) \tag{5.7-44}$$

In general, the above RMFC works well when the mass and the load are negligible, as compared with the mass of the manipulator. But, if the mass and the load approaches the mass of the manipulator, the position of the hand usually does not converge to the desired position. This is due to the fact that some of the joint torques are spent to accelerate the links. In order to compensate for these loading and acceleration effects, a force convergence control is incorporated as a second part of the RMFC.

The force convergent control method is based on the Robbins-Monro stochastic approximation method to determine the actual cartesian force $\mathbf{F}_a$ so that the observed cartesian force $\mathbf{F}_0$ (measured by a wrist force sensor) at the hand will converge to the desired cartesian force $\mathbf{F}^d$ obtained from the above position control technique. If the error between the measured force vector $\mathbf{F}_0$ and the desired cartesian force is greater than a user-designed threshold $\Delta\mathbf{F}(k) = \mathbf{F}^d(k) - \mathbf{F}_0(k)$, then the actual cartesian force is updated by

$$\mathbf{F}_a(k + 1) = \mathbf{F}_a(k) + \gamma_k\,\Delta\mathbf{F}(k) \tag{5.7-45}$$

where $\gamma_k = 1/(k + 1)$ for $k = 0, 1, \ldots, N$. Theoretically, the value of N must be large. However, in practice, the value of N can be chosen based on the force convergence. Based on a computer simulation study (Wu and Paul [1982]), a value of $N = 1$ or 2 gives a fairly good convergence of the force vector.

In summary, the RMFC with force convergent control has the advantage that the control method can be extended to various loading conditions and to a manipulator with any number of degrees of freedom without increasing the computational complexity.

5.8 ADAPTIVE CONTROL

Most of the schemes discussed in the previous sections control the arm at the hand or joint level and emphasize nonlinear compensations of the interaction forces between the various joints. These control algorithms sometimes are inadequate because they require accurate modeling of the arm dynamics and neglect the changes of the load in a task cycle. These changes in the payload of the controlled system often are significant enough to render the above feedback control strategies ineffective. The result is reduced servo response speed and damping, which limits the precision and speed of the end-effector. Any significant gain in performance for tracking the desired time-based trajectory as closely as possible over a wide range of manipulator motion and payloads require the consideration of adaptive control techniques.

5.8.1 Model-Referenced Adaptive Control

Among various adaptive control methods, the model-referenced adaptive control (MRAC) is the most widely used and it is also relatively easy to implement. The concept of model-referenced adaptive control is based on selecting an appropriate reference model and adaptation algorithm which modifies the feedback gains to the actuators of the actual system. The adaptation algorithm is driven by the errors between the reference model outputs and the actual system outputs. A general control block diagram of the model-referenced adaptive control system is shown in Fig. 5.13.

Dubowsky and DesForges [1979] proposed a simple model-referenced adaptive control for the control of mechanical manipulators. In their analysis, the payload is taken into consideration by combining it to the final link, and the end-effector dimension is assumed to be small compared with the length of other links. Then, the selected reference model provides an effective and flexible means of specifying desired closed-loop performance of the controlled system. A linear second-order time invariant differential equation is selected as the reference model for each degree of freedom of the robot arm. The manipulator is controlled by adjusting the position and velocity feedback gains to follow the model so that its closed-loop performance characteristics closely match the set of desired performance characteristics in the reference model. As a result, this adaptive control scheme only

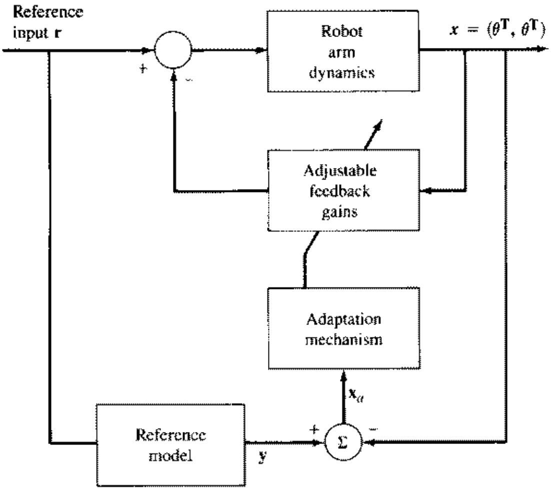

Figure 5.13 A general control block diagram for model-referenced adaptive control.

requires moderate computations which can be implemented with a low-cost microprocessor. Such a model-referenced, adaptive control algorithm does not require complex mathematical models of the system dynamics nor the a priori knowledge of the environment (loads, etc). The resulting model-referenced adaptive system is capable of maintaining uniformly good performance over a wide range of motions and payloads.

Defining the vector $\mathbf{y}(t)$ to represent the reference model response and the vector $\mathbf{x}(t)$ to represent the manipulator response, the joint i of the reference model can be described by

$$a_i \ddot{y}_i(t) + b_i \dot{y}_i(t) + y_i(t) = r_i(t) \tag{5.8-1}$$

In terms of natural frequency ω_{ni} and damping ratio ζ_i of a second-order linear system, a_i and b_i correspond to

$$a_i = \frac{1}{\omega_{ni}^2} \quad \text{and} \quad b_i = \frac{2\zeta_i}{\omega_{ni}} \tag{5.8-2}$$

If we assume that the manipulator is controlled by position and velocity feedback gains, and that the coupling terms are negligible, then the manipulator dynamic equation for joint i can be written as

$$\alpha_i(t)\ddot{x}_i(t) + \beta_i(t)\dot{x}_i(t) + x_i(t) = r_i(t) \tag{5.8-3}$$

where the system parameters $\alpha_i(t)$ and $\beta_i(t)$ are assumed to vary slowly with time.

Several techniques are available to adjust the feedback gains of the controlled system. Due to its simplicity, a steepest descent method is used to minimize a quadratic function of the system error, which is the difference between the response of the actual system [Eq. (5.8-3)] and the response of the reference model [Eq. (5.8-1)]:

$$J_i(e_i) = \tfrac{1}{2}(k_2^i\ddot{e}_i + k_1^i\dot{e}_i + k_0^i e_i)^2 \qquad i = 1, 2, \ldots, n \qquad (5.8\text{-}4)$$

where $e_i = y_i - x_i$, and the values of the weighting factors, k_j^i, are selected from stability considerations to obtain stable system behavior.

Using a steepest descent method, the system parameters adjustment mechanism which will minimize the system error is governed by

$$\dot{\alpha}_i(t) = [k_2^i\ddot{e}_i(t) + k_1^i\dot{e}_i(t) + k_0^i e_i(t)][k_2^i\ddot{u}_i(t) + k_1^i\dot{u}_i(t) + k_0^i u_i(t)] \qquad (5.8\text{-}5)$$

$$\dot{\beta}_i(t) = [k_2^i\ddot{e}_i(t) + k_1^i\dot{e}_i(t) + k_0^i e_i(t)][k_2^i\ddot{w}_i(t) + k_1^i\dot{w}_i(t) + k_0^i w_i(t)] \qquad (5.8\text{-}6)$$

where $u_i(t)$ and $w_i(t)$ and their derivatives are obtained from the solutions of the following differential equations:

$$a_i\ddot{u}_i(t) + b_i\dot{u}_i(t) + u_i(t) = -\ddot{y}_i(t) \qquad (5.8\text{-}7)$$

$$a_i\ddot{w}_i(t) + b_i\dot{w}_i(t) + w_i(t) = -\dot{y}_i(t) \qquad (5.8\text{-}8)$$

and $\dot{y}_i(t)$ and $\ddot{y}_i(t)$ are the first two time derivatives of response of the reference model. The closed-loop adaptive system involves solving the reference model equations for a given desired input; then the differential equations in Eqs. (5.8-7) and (5.8-8) are solved to yield $u_i(t)$ and $w_i(t)$ and their derivatives for Eqs. (5.8-5) and (5.8-6). Finally, solving the differential equations in Eqs. (5.8-5) and (5.8-6), yields $\alpha_i(t)$ and $\beta_i(t)$.

The fact that this control approach is not dependent on a complex mathematical model is one of its major advantages, but stability considerations of the closed-loop adaptive system are critical. A stability analysis is difficult, and Dubowsky and DesForges [1979] carried out an investigation of this adaptive system using a linearized model. However, the adaptability of the controller can become questionable if the interaction forces among the various joints are severe.

5.8.2 Adaptive Control Using an Autoregressive Model

Koivo and Guo [1983] proposed an adaptive, self-tuning controller using an autoregressive model to fit the input-output data from the manipulator. The control algorithm assumes that the interaction forces among the joints are negligible. A block diagram of the control system is shown in Fig. 5.14. Let the input torque to joint i be u_i, and the output angular position of the manipulator be y_i. The input-output

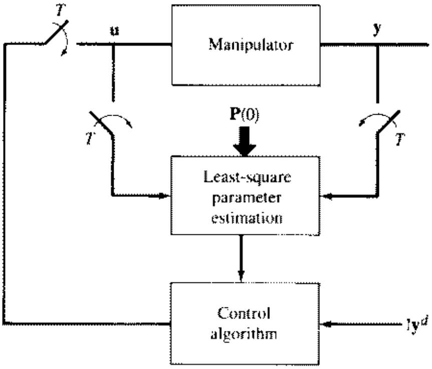

Figure 5.14 Adaptive control with autoregressive model.

pairs (u_i, y_i) may be described by an autoregressive model which match these pairs as closely as possible:

$$y_i(k) = \sum_{m=1}^{n} [a_i^m y_i(k-m) + b_i^m u_i(k-m)] + a_i^0 + e_i(k) \qquad (5.8\text{-}9)$$

where a_i^0 is a constant forcing term, $e_i(k)$ is the modeling error which is assumed to be white gaussian noise with zero mean and independent of u_i and $y_i(k-m)$, $m \geqslant 1$. The parameters a_i^m and b_i^m are determined so as to obtain the best least-squares fit of the measured input-output data pairs. These parameters can be obtained by minimizing the following criterion:

$$E_N^i(\alpha_i) = \frac{1}{N+1}\sum_{k=0}^{N} e_i^2(k) \qquad (5.8\text{-}10)$$

where N is the number of measurements. Let α_i be the ith parameter vector:

$$\alpha_i = (a_i^0, a_i^1, \ldots, a_i^n, b_i^0, b_i^1, \ldots, b_i^n)^T \qquad (5.8\text{-}11)$$

and let $\psi_i(k-1)$ be the vector of the input-output pairs:

$$\psi_i(k-1) = [1, y_i(k-1), \ldots, y_i(k-n), u_i(k-1), \ldots, u_i(k-n)]^T \qquad (5.8\text{-}12)$$

Then, a recursive least-squares estimation of α_i can be found as

$$\hat{\alpha}_i(N) = \hat{\alpha}_i(N-1) + \mathbf{P}_i(N)\psi_i(N-1)[y_i(N) - \hat{\alpha}_i^T(N-1)\psi_i(N-1)] \qquad (5.8\text{-}13)$$

with

$$\mathbf{P}_i(N) = \frac{1}{\mu_i}\left[\frac{\mathbf{P}_i(N-1)\psi_i(N-1)\psi_i^T(N-1)\mathbf{P}_i(N-1)}{\mu_i + \psi_i^T(N-1)\mathbf{P}_i(N-1)\psi_i(N-1)}\right] \tag{5.8-14}$$

where $0 < \mu_i \leqslant 1$ is a "forgetting" factor which provides an exponential weighting of past data in the estimation algorithm and by which the algorithm allows a slow drift of the parameters; $\mathbf{P}_i$ is a $(2n + 1) \times (2n + 1)$ symmetric matrix, and the hat notation is used to indicate an estimate of the parameters.

Using the above equations to compute the estimates of the autoregressive model, the model can be represented by:

$$y_i(k) = \hat{\alpha}_i^T\psi_i(k-1) + e_i(k) \tag{5.8-15}$$

In order to track the trajectory set points, a performance criterion for joint i is defined as

$$J_i^k(\mathbf{u}) = E\{[y_i(k+2) - y_i^d(k+2)]^2 + \gamma_i u_i^2(k+1)|\psi_i(k)\} \tag{5.8-16}$$

where $E[\,\cdot\,]$ represents an expectation operation conditioned on $\psi_i(k)$ and γ_i is a user-defined nonnegative weighting factor.

The optimal control that minimizes the above performance criterion is found to be:

$$\begin{aligned} u_i(k+1) &\\ = \frac{-\hat{b}_i^1(k)}{[\hat{b}_i^1(k)]^2 + \gamma_i} &\left\{\hat{a}_i^0(k) + \hat{a}_i^1(k)[\hat{\alpha}_i^T\psi_i(k)] + \sum_{m=2}^{n}\hat{a}_i^m(k)y_i(k+2-m)\right. \\ &\left. + \sum_{m=2}^{n}\hat{b}_i^m(k)u_i(k+2-m) - y_i^d(k+2)\right\} \end{aligned} \tag{5.8-17}$$

where $\hat{a}_i^m$, $\hat{b}_i^m$, and $\hat{\alpha}_i^m$ are the estimates of the parameters from Eqs. (5.8-13) and (5.8-14).

In summary, this adaptive control uses an autoregressive model [Eq. (5.8-9)] to fit the input-output data from the manipulator. The recursive least-squares identification scheme [Eqs. (5.8-13) and (5.8-14)] is used to estimate the parameters which are used in the optimal control [Eq. (5.8-17)] to servo the manipulator.

5.8.3 Adaptive Perturbation Control

Based on perturbation theory, Lee and Chung [1984, 1985] proposed an adaptive control strategy which tracks a desired time-based manipulator trajectory as closely as possible for all times over a wide range of manipulator motion and payloads. Adaptive perturbation control differs from the above adaptive schemes in the sense that it takes all the interactions among the various joints into consideration. The adaptive control discussed in this section is based on linearized perturbation equations in the vicinity of a nominal trajectory. The nominal trajectory is specified by an interpolated joint trajectory whose angular position, angular velocity, and angular acceleration are known at every sampling instant. The highly coupled nonlinear dynamic equations of a manipulator are then linearized about the planned manipulator trajectory to obtain the linearized perturbation system. The controlled system is characterized by feedforward and feedback components which can be computed separately and simultaneously. Using the Newton-Euler equations of motion as inverse dynamics of the manipulator, the feedforward component computes the nominal torques which compensate all the interaction forces between the various joints along the nominal trajectory. The feedback component computes the perturbation torques which reduce the position and velocity errors of the manipulator to zero along the nominal trajectory. An efficient, recursive, real-time, least-squares identification scheme is used to identify the system parameters in the perturbation equations. A one-step optimal control law is designed to control the linearized perturbation system about the nominal trajectory. The parameters and the feedback gains of the linearized system are updated and adjusted in each sampling period to obtain the necessary control effort. The total torques applied to the joint actuators then consist of the nominal torques computed from the Newton-Euler equations of motion and the perturbation torques computed from the one-step optimal control law of the linearized system. This adaptive control strategy reduces the manipulator control problem from nonlinear control to controlling a linear system about a nominal trajectory.

The adaptive control is based on the linearized perturbation equations about the referenced trajectory. We need to derive appropriate linearized perturbation equations suitable for developing the feedback controller which computes perturbation joint torques to reduce position and velocity errors along the nominal trajectory. The L-E equations of motion of an n-link manipulator can be expressed in state space representation as in Eq. (5.4-4). With this formulation, the control problem is to find a feedback control law $\mathbf{u}(t) = \mathbf{g}[\mathbf{x}(t)]$ such that the closed loop control system $\dot{\mathbf{x}}(t) = \mathbf{f}\{\mathbf{x}(t), \mathbf{g}[\mathbf{x}(t)]\}$ is asymptotically stable and tracks a desired trajectory as closely as possible over a wide range of payloads for all times.

Suppose that the nominal states $\mathbf{x}_n(t)$ of the system [Eq. (5.4-4)] are known from the planned trajectory, and the corresponding nominal torques $\mathbf{u}_n(t)$ are also known from the computations of the joint torques using the N-E equations of motion. Then, both $\mathbf{x}_n(t)$ and $\mathbf{u}_n(t)$ satisfy Eq. (5.4-4):

$$\dot{\mathbf{x}}_n(t) = \mathbf{f}[\mathbf{x}_n(t), \mathbf{u}_n(t)] \tag{5.8-18}$$

Using the Taylor series expansion on Eq. (5.4-4) about the nominal trajectory, subtracting Eq. (5.8-18) from it, and assuming that the higher order terms are negligible, the associated linearized perturbation model for this control system can be expressed as

$$\begin{aligned}\delta\dot{\mathbf{x}}(t) &= \nabla_x \mathbf{f}|_n \, \delta\mathbf{x}(t) + \nabla_u \mathbf{f}|_n \, \delta\mathbf{u}(t) \\ &= \mathbf{A}(t)\,\delta\mathbf{x}(t) + \mathbf{B}(t)\,\delta\mathbf{u}(t) \qquad (5.8\text{-}19)\end{aligned}$$

where $\nabla_x \mathbf{f}|_n$ and $\nabla_u \mathbf{f}|_n$ are the jacobian matrices of $\mathbf{f}[\mathbf{x}(t), \mathbf{u}(t)]$ evaluated at $\mathbf{x}_n(t)$ and $\mathbf{u}_n(t)$, respectively, $\delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_n(t)$ and $\delta\mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_n(t)$.

The system parameters, $\mathbf{A}(t)$ and $\mathbf{B}(t)$, of Eq. (5.8-19) depend on the instantaneous manipulator position and velocity along the nominal trajectory and thus, vary slowly with time. Because of the complexity of the manipulator equations of motion, it is extremely difficult to find the elements of $\mathbf{A}(t)$ and $\mathbf{B}(t)$ explicitly. However, the design of a feedback control law for the perturbation equations requires that the system parameters of Eq. (5.8-19) be known at all times. Thus, parameter identification techniques must be used to identify the unknown elements in $\mathbf{A}(t)$ and $\mathbf{B}(t)$.

As a result of this formulation, the manipulator control problem is reduced to determining $\delta\mathbf{u}(t)$, which drives $\delta\mathbf{x}(t)$ to zero at all times along the nominal trajectory. The overall controlled system is thus characterized by a feedforward component and a feedback component. Given the planned trajectory set points $\mathbf{q}^d(t)$, $\dot{\mathbf{q}}^d(t)$, and $\ddot{\mathbf{q}}^d(t)$, the feedforward component computes the corresponding nominal torques $\mathbf{u}_n(t)$ from the N-E equations of motion. The feedback component computes the corresponding perturbation torques $\delta\mathbf{u}(t)$ which provide control effort to compensate for small deviations from the nominal trajectory. The computation of the perturbation torques is based on a one-step optimal control law. The main advantages of this formulation are twofold. First, it reduces a nonlinear control problem to a linear control problem about a nominal trajectory; second, the computations of the nominal and perturbation torques can be performed separately and simultaneously. Because of this parallel computational structure, adaptive control techniques can be easily implemented using present day low-cost microprocessors. A control block diagram of the method is shown in Fig. 5.15.

For implementation on a digital computer, Eq. (5.8-19) needs to be discretized to obtain an appropriate discrete linear equations for parameter identification:

$$\mathbf{x}[(k+1)T] = \mathbf{F}(kT)\mathbf{x}(kT) + \mathbf{G}(kT)\mathbf{u}(kT)$$

$$k = 0, 1, \ldots \qquad (5.8\text{-}20)$$

where T is the sampling period, $\mathbf{u}(kT)$ is an n-dimensional piecewise constant control input vector of $\mathbf{u}(t)$ over the time interval between any two consecutive sam-

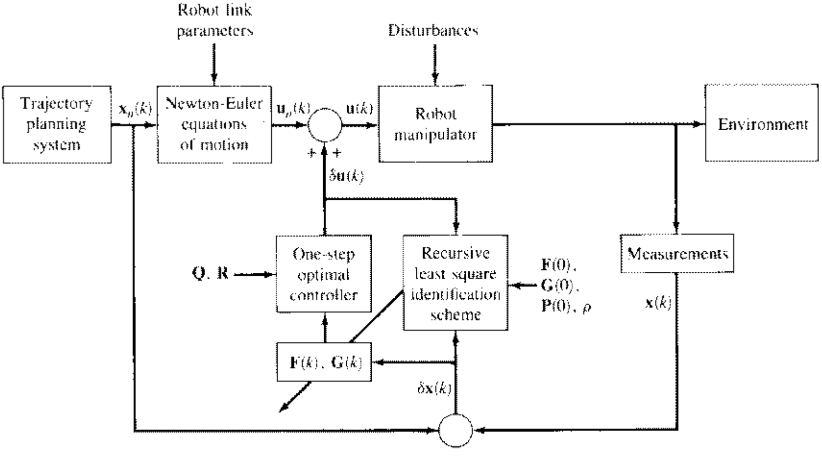

Figure 5.15 The adaptive perturbation control.

pling instants for $kT \leqslant t < (k+1)T$, and $\mathbf{x}(kT)$ is a $2n$-dimensional perturbed state vector which is given by

$$\mathbf{x}(kT) = \boldsymbol{\Gamma}(kT, t_0)\mathbf{x}(t_0) + \int_{t_0}^{kT} \boldsymbol{\Gamma}(kT, t)\mathbf{B}(t)\mathbf{u}(t)dt \qquad (5.8\text{-}21)$$

and $\boldsymbol{\Gamma}(kT, t_0)$ is the state-transition matrix of the system. $\mathbf{F}(kT)$ and $\mathbf{G}(kT)$ are, respectively, $2n \times 2n$ and $2n \times n$ matrices and are given by

$$\mathbf{F}(kT) = \boldsymbol{\Gamma}[(k+1)T, kT] \qquad (5.8\text{-}22)$$

and

$$\mathbf{G}(kT)\mathbf{u}(kT) = \int_{kT}^{(k+1)T} \boldsymbol{\Gamma}[(k+1)T, t]\mathbf{B}(t)\mathbf{u}(t)dt \qquad (5.8\text{-}23)$$

With this model, a total of $6n^2$ parameters in the $\mathbf{F}(kT)$ and $\mathbf{G}(kT)$ matrices need to be identified. Without confusion, we shall drop the sampling period T from the rest of the equations for clarity and simplicity.

Various identification algorithms, such as the methods of least squares, maximum likelihood, instrumental variable, cross correlation, and stochastic approximation, have been applied successfully to the parameter identification problem. Due to its simplicity and ease of application, a recursive real-time least-squares parameter identification scheme is selected here for identifying the system parameters in $\mathbf{F}(k)$ and $\mathbf{G}(k)$. In the parameter identification scheme, we make the following assumptions: (1) the parameters of the system are slowly time-varying but the variation speed is slower than the adaptation speed; (2) measurement noise is negligible; and (3) the state variables $\mathbf{x}(k)$ of Eq. (5.8-20) are measurable.

In order to apply the recursive least-squares identification algorithm to Eq. (5.8-20), we need to rearrange the system equations in a form that is suitable for parameter identification. Defining and expressing the ith row of the unknown parameters of the system at the kth instant of time in a $3n$-dimensional vector, we have

$$\boldsymbol{\theta}_i^T(k) = [f_{i1}(k), \ldots, f_{ip}(k), g_{i1}(k), \ldots, g_{in}(k)] \qquad (5.8\text{-}24)$$

$$i = 1, 2, \ldots, p$$

or, expressed in matrix form, as

$$\boldsymbol{\Theta}(k) = \begin{bmatrix} f_{11}(k) & \cdots & f_{p1}(k) \\ \cdot & & \cdot \\ \cdot & \cdots & \cdot \\ \cdot & & \cdot \\ f_{1p}(k) & \cdots & f_{pp}(k) \\ g_{11}(k) & \cdots & g_{p1}(k) \\ \cdot & & \cdot \\ \cdot & \cdots & \cdot \\ \cdot & & \cdot \\ g_{1n}(k) & \cdots & g_{pn}(k) \end{bmatrix} = [\boldsymbol{\theta}_1(k), \boldsymbol{\theta}_2(k), \ldots, \boldsymbol{\theta}_p(k)] \qquad (5.8\text{-}25)$$

where $p = 2n$. Similarly, defining the outputs and inputs of the perturbation system [Eq. (5.8-20)] at the kth instant of time in a $3n$-dimensional vector as

$$\mathbf{z}^T(k) = [x_1(k), x_2(k), \ldots, x_p(k), u_1(k), u_2(k), \ldots, u_n(k)] \quad (5.8\text{-}26)$$

and the states at the kth instant of time in a $2n$-dimensional vector as

$$\mathbf{x}^T(k) = [x_1(k), x_2(k), \ldots, x_p(k)] \qquad (5.8\text{-}27)$$

we have that the corresponding system equation in Eq. (5.8-20) can be written as

$$x_i(k+1) = \mathbf{z}^T(k)\boldsymbol{\theta}_i(k) \qquad i = 1, 2, \ldots, p \qquad (5.8\text{-}28)$$

With this formulation, we wish to identify the parameters in each column of $\boldsymbol{\Theta}(k)$ based on the measurement vector $\mathbf{z}(k)$. In order to examine the "goodness" of the least-squares estimation algorithm, a $2n$-dimensional error vector $\mathbf{e}(k)$, often called a *residual*, is included to account for the modeling error and noise in Eq. (5.8-20):

$$e_i(k) = x_i(k+1) - \mathbf{z}^T(k)\hat{\boldsymbol{\theta}}_i(k) \qquad i = 1, 2, \ldots, p \qquad (5.8\text{-}29)$$

Basic least-squares parameter estimation assumes that the unknown parameters are constant values and the solution is based on batch processing N sets of measurement data, which are weighted equally, to estimate the unknown parameters. Unfortunately, this algorithm cannot be applied to time-varying parameters. Furthermore, the solution requires matrix inversion which is computational intensive. In order to reduce the number of numerical computations and to track the time-varying parameters $\Theta(k)$ at each sampling period, a sequential least-squares identification scheme which updates the unknown parameters at each sampling period based on the new set of measurements at each sampling interval provides an efficient algorithmic solution to the identification problem. Such a recursive, real-time, least-squares parameter identification algorithm can be found by minimizing an exponentially weighted error criterion which has an effect of placing more weights on the squared errors of the more recent measurements; that is,

$$J_N = \sum_{j=1}^{N} \rho^{N-j} e_i^2(j) \tag{5.8-30}$$

where the error vector is weighted as

$$\mathbf{e}_i^T(N) = [\sqrt{\rho}^{N-1} e_i(1), \sqrt{\rho}^{N-2} e_i(2), \ldots, e_i(N)] \tag{5.8-31}$$

and $N > 3n$ is the number of measurements used to estimate the parameters $\boldsymbol{\theta}_i(N)$. Minimizing the error criterion in Eq. (5.8-30) with respect to the unknown parameters vector $\boldsymbol{\theta}_i$ and utilizing the matrix inverse lemma, a recursive real-time least-squares identification scheme can be obtained for $\boldsymbol{\theta}_i(k)$ after simple algebraic manipulations:

$$\hat{\boldsymbol{\theta}}_i(k+1) = \hat{\boldsymbol{\theta}}_i(k) + \gamma(k)\mathbf{P}(k)\mathbf{z}(k)\,[x_i(k+1) - \mathbf{z}^T(k)\hat{\boldsymbol{\theta}}_i(k)\,] \tag{5.8-32}$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \gamma(k)\,\mathbf{P}(k)\mathbf{z}(k)\mathbf{z}^T(k)\mathbf{P}(k) \tag{5.8-33}$$

and $$\gamma(k) = [\mathbf{z}^T(k)\mathbf{P}(k)\mathbf{z}(k) + \rho]^{-1} \tag{5.8-34}$$

where $0 < \rho < 1$, the hat notation is used to indicate the estimate of the parameters $\boldsymbol{\theta}_i(k)$, and $\mathbf{P}(k) = \rho[\mathbf{Z}(k)\mathbf{Z}^T(k)]^{-1}$ is a $3n \times 3n$ symmetric positive definite matrix, where $\mathbf{Z}(k) = [\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(k)]$ is the measurement matrix up to the kth sampling instant. If the errors $e_i(k)$ are identically distributed and independent with zero mean and variance σ^2, then $\mathbf{P}(k)$ can be interpreted as the covariance matrix of the estimate if ρ is chosen as σ^2.

The above recursive equations indicate that the estimate of the parameters $\hat{\boldsymbol{\theta}}_i(k+1)$ at the $(k+1)$th sampling period is equal to the previous estimate $\hat{\boldsymbol{\theta}}_i(k)$ corrected by the term proportional to $[x_i(k+1) - \mathbf{z}^T(k)\hat{\boldsymbol{\theta}}_i(k)]$. The term $\mathbf{z}^T(k)\hat{\boldsymbol{\theta}}_i(k)$ is the prediction of the value $x_i(k+1)$ based on the estimate of the parameters $\boldsymbol{\theta}_i(k)$ and the measurement vector $\mathbf{z}(k)$. The components of the vector $\gamma(k)\mathbf{P}(k)\mathbf{z}(k)$ are weighting factors which indicate how the corrections and the previous estimate should be weighted to obtain the new estimate $\hat{\boldsymbol{\theta}}_i(k+1)$. The parameter ρ is a weighting factor and is commonly used for tracking slowly time-varying parameters by exponentially forgetting the "aged" measurements. If $\rho << 1$, a large weighting factor is placed on the more recent sampled data by rapidly weighing out previous samples. If $\rho \approx 1$, accuracy in tracking the time-varying parameters will be lost due to the truncation of the measured data sequences. We can compromise between fast adaptation capabilities and loss of accuracy in parameter identification by adjusting the weighting factor ρ. In most applications for tracking slowly time-varying parameters, ρ is usually chosen to be $0.90 \leqslant \rho < 1.0$.

Finally, the above identification scheme [Eqs. (5.8-32) to (5.8-34)] can be started by choosing the initial values of $\mathbf{P}(0)$ to be

$$\mathbf{P}(0) = \alpha \mathbf{I}_{3n} \tag{5.8-35}$$

where α is a large positive scalar and $\mathbf{I}_{3n}$ is a $3n \times 3n$ identity matrix. The initial estimate of the unknown parameters $\mathbf{F}(k)$ and $\mathbf{G}(k)$ can be approximated by the following equations:

$$\mathbf{F}(0) \approx \mathbf{I}_{2n} + \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\} T + \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\}^2 \frac{T^2}{2} \tag{5.8-36}$$

$$\begin{aligned}\mathbf{G}(0) \approx & \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{u}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\} T + \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\} \\ & \times \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{u}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\} T^2 \\ & + \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{x}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\}^2 \left\{ \frac{\partial \mathbf{f}}{\partial \mathbf{u}}[\mathbf{x}_n(0), \mathbf{u}_n(0)] \right\} \frac{T^3}{2}\end{aligned} \tag{5.8-37}$$

where T is the sampling period.

With the determination of the parameters in $\mathbf{F}(k)$ and $\mathbf{G}(k)$, proper control laws can be designed to obtain the required correction torques to reduce the position and velocity errors of the manipulator along a nominal trajectory. This can be done by finding an optimal control $\mathbf{u}^*(k)$ which minimizes the performance index $J(k)$ while satisfying the constraints of Eq. (5.8-20):

$$J(k) = ½[\mathbf{x}^T(k+1)\mathbf{Q}\mathbf{x}(k+1) + \mathbf{u}^T(k)\mathbf{R}\mathbf{u}(k)] \tag{5.8-38}$$

where $\mathbf{Q}$ is a $p \times p$ semipositive definite weighting matrix and $\mathbf{R}$ is an $n \times n$ positive definite weighting matrix. The one-step performance index in Eq. (5.8-38) indicates that the objective of the optimal control is to drive the position and velocity errors of the manipulator to zero along the nominal trajectory in a coordinated position and rate control per interval step while, at the same time, attaching a cost to the use of control effort. The optimal control solution which minimizes the functional in Eq. (5.8-38) subject to the constraints of Eq. (5.8-20) is well known and is found to be (Saridis and Lobbia [1972])

$$\mathbf{u}^*(k) = -[\mathbf{R} + \hat{\mathbf{G}}^T(k)\mathbf{Q}\hat{\mathbf{G}}(k)]^{-1}\hat{\mathbf{G}}^T(k)\,\mathbf{Q}\,\hat{\mathbf{F}}(k)\,\mathbf{x}(k) \qquad (5.8\text{-}39)$$

where $\hat{\mathbf{F}}(k)$ and $\hat{\mathbf{G}}(k)$ are the system parameters obtained from the identification algorithm [Eqs. (5.8-32) to (5.8-34)] at the kth sampling instant.

The identification and control algorithms in Eqs. (5.8-32) to (5.8-34) and Eq. (5.8-39) do not require complex computations. In Eq. (5.8-34), $[\mathbf{z}^T(k)\mathbf{P}(k)\mathbf{z}(k) + \rho]$ gives a scalar, so its inversion is trivial. Although the weighting factor ρ can be adjusted for each ith parameter vector $\boldsymbol{\theta}_i(k)$ as desired, this requires excessive computations in the $\mathbf{P}(k + 1)$ matrix. For real-time robot arm control, such adjustments are not desirable. $\mathbf{P}(k + 1)$ is computed only once at each sampling time using the same weighting factor ρ. Moreover, since $\mathbf{P}(k)$ is a symmetric positive definite matrix, only the upper diagonal matrix of $\mathbf{P}(k)$ needs to be computed. The combined identification and control algorithm can be computed in $O(n^3)$ time. The computational requirements of the adaptive perturbation control are tabulated in Table 5.1. Based on the specifications of a DEC PDP 11/45 computer, an ADDF (floating point addition) instruction requires 5.17μs and a MULF (floating point multiply) instruction requires 7.17μs. If we assume that for each ADDF and MULF instruction, we need to fetch data from the core memory twice and the memory cycle time is 450 ns, then the adaptive perturbation control requires approximately 7.5 ms to compute the necessary joint torques to servo the first three joints of a PUMA robot arm for a trajectory set point.

A computer simulation study of a three-joint PUMA manipulator was conducted (Lee and Chung [1984,1985]) to evaluate and compare the performance of the adaptive controller with the controller [Eq. (5.3-65)], which is basically a proportional plus derivative control (PD controller). The study was carried out for various loading conditions along a given trajectory. The performances of the PD and adaptive controllers are compared and evaluated for three different loading

Table 5.1 Computations of the adaptive controller

Adaptive controller	Multiplications	Additions
Newton-Euler equations of motion	$117n - 24$	$103n - 21$
Least-squares identification algorithm	$30n^2 + 5n + 1$	$30n^2 + 3n - 1$
Control algorithm	$8n^3 + 2n^2 + 39$	$8n^3 - n^2 - n + 18$
Total	$8n^3 + 32n^2 + 5n + 40$	$8n^3 + 29n^2 + 2n + 17$

Table 5.2 Comparisons of the PD and adaptive controllers

		PD controller			Adaptive controller		
		Trajectory tracking			Trajectory tracking		
Various loading conditions	Joint	Max. error (degrees)	Max. error (mm)	Final position error (degrees)	Max. error (degrees)	Max. error (mm)	Final position error (degrees)
No-load and	1	0.089	1.55	0.025	0.020	0.34	0.000
10% error	2	0.098	1.71	0.039	0.020	0.36	0.004
In inertia tensor	3	0.328	2.86	0.121	0.032	0.28	0.002
½ max. load	1	0.121	2.11	0.054	0.045	0.78	0.014
and 10% error	2	0.147	2.57	0.078	0.065	1.14	0.050
In inertia tensor	3	0.480	4.19	0.245	0.096	0.83	0.077
Max. load	1	0.145	2.53	0.082	0.069	1.20	0.023
and 10% error	2	0.185	3.23	0.113	0.069	1.22	0.041
In inertia tensor	3	0.607	5.30	0.360	0.066	0.58	0.019

conditions and the results are tabulated in Table 5.2: (1) no-load and 10 percent error in inertia tensor, (2) half of maximum load and 10 percent error in inertia tensor, and (3) maximum load (5 lb) and 10 percent error in inertia tensor. In each case, a 10 percent error in inertia matrices means ± 10 percent error about its measured inertial values. For all the above cases, the adaptive controller shows better performance than the PD controller with constant feedback gains both in trajectory tracking and the final position errors. Plots of angular position errors for the above cases for the adaptive control are shown in Figs. 5.16 to 5.18. Additional details of the simulation result can be found in Lee and Chung [1984, 1985].

5.8.4 Resolved Motion Adaptive Control

The adaptive control strategy of Sec. 5.8.3 in the joint variable space can be extended to control the manipulator in cartesian coordinates under various loading conditions by adopting the ideas of resolved motion rate and acceleration controls. The resolved motion adaptive control is performed at the hand level and is based on the linearized perturbation system along a desired time-based hand trajectory. The resolved motion adaptive control differs from the resolved motion acceleration control by minimizing the position/orientation and angular and linear velocities of the manipulator hand along the hand coordinate axes instead of position and orientation errors. Similar to the previous adaptive control, the controlled system is characterized by feedforward and feedback components which can be computed separately and simultaneously. The feedforward component resolves the specified

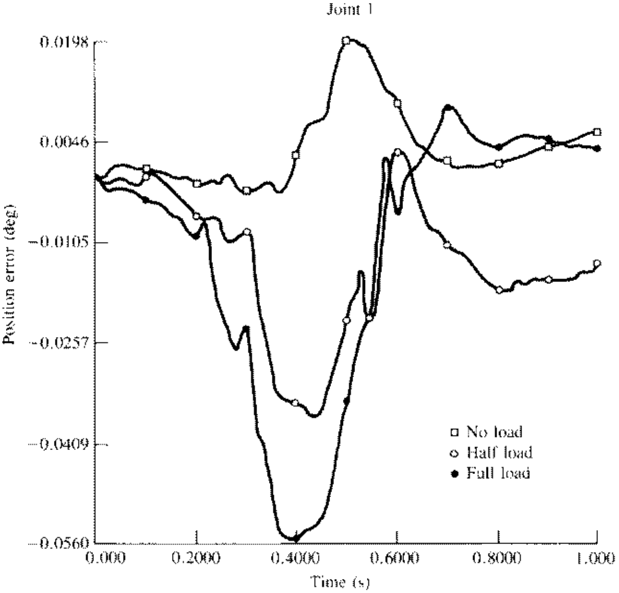

Figure 5.16 Joint 1 position error under various loads.

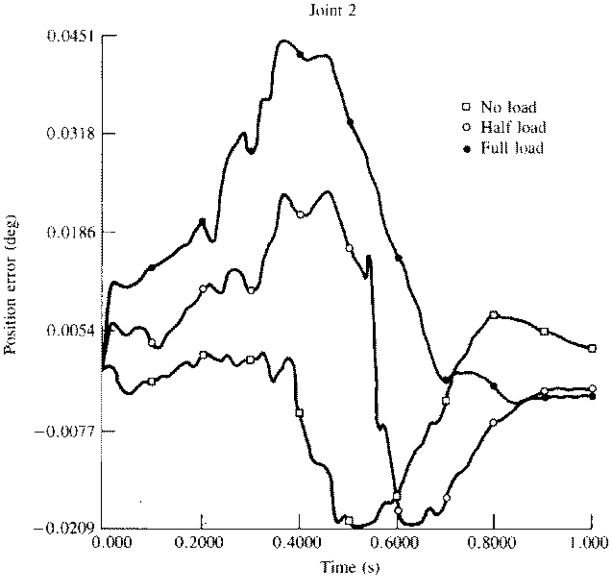

Figure 5.17 Joint 2 position error under various loads.

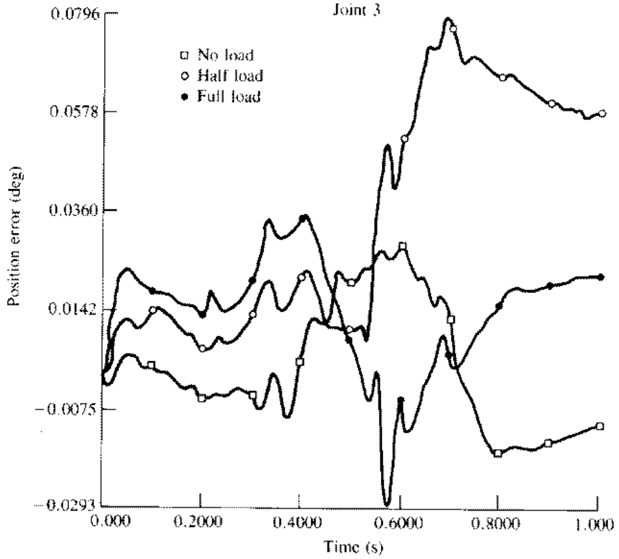

Figure 5.18 Joint 3 position error under various loads.

positions, velocities, and accelerations of the hand into a set of values of joint positions, velocities, and accelerations from which the nominal joint torques are computed using the Newton-Euler equations of motion to compensate for all the interaction forces among the various joints. The feedback component computes the perturbation joint torques which reduce the manipulator hand position and velocity errors along the nominal hand trajectory. A recursive least-squares identification scheme is again used to perform on-line parameter identification of the linearized system.

Using the kinematic relationship between the joint coordinates and the cartesian coordinates derived previously in Eqs. (5.7-1) to (5.7-14), the equations of motion of the manipulator in cartesian coordinates can be easily obtained. The acceleration of the manipulator has been obtained previously in Eq. (5.7-13) and is repeated here for convenience:

$$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\mathbf{\Omega}}(t) \end{bmatrix} = \dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{N}^{-1}(\mathbf{q}) \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{\Omega}(t) \end{bmatrix} + \mathbf{N}(\mathbf{q})\ddot{\mathbf{q}}(t) \qquad (5.8\text{-}40)$$

In order to include the dynamics of the manipulator into the above kinematics equation [Eq. (5.8-40)], we need to use the L-E equations of motion given in Eq. (3.2-26). Since $\mathbf{D}(\mathbf{q})$ is always nonsingular, $\ddot{\mathbf{q}}(t)$ can be obtained from Eq. (3.2-

26) and substituted into Eq. (5.8-40) to obtain the accelerations of the manipulator hand:

$$\begin{bmatrix}\dot{\mathbf{v}}(t)\\ \dot{\boldsymbol{\Omega}}(t)\end{bmatrix} = \dot{\mathbf{N}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{N}^{-1}(\mathbf{q})\begin{bmatrix}\mathbf{v}(t)\\ \boldsymbol{\Omega}(t)\end{bmatrix} + \mathbf{N}(\mathbf{q})\mathbf{D}^{-1}(\mathbf{q})[\boldsymbol{\tau}(t)-\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})-\mathbf{c}(\mathbf{q})] \tag{5.8-41}$$

For convenience, let us partition $\mathbf{N}(\mathbf{q})$, $\mathbf{N}^{-1}(\mathbf{q})$, and $\mathbf{D}^{-1}(\mathbf{q})$ into 3×3 submatrices and $\mathbf{h}(\mathbf{q},\dot{\mathbf{q}})$, $\mathbf{c}(\mathbf{q})$, and $\boldsymbol{\tau}(t)$ into 3×1 submatrices:

$$\mathbf{N}(\mathbf{q}) \triangleq \begin{bmatrix}\mathbf{N}_{11}(\mathbf{q}) & \mathbf{N}_{12}(\mathbf{q})\\ \mathbf{N}_{21}(\mathbf{q}) & \mathbf{N}_{22}(\mathbf{q})\end{bmatrix} \tag{5.8-42a}$$

$$\mathbf{N}^{-1}(\mathbf{q}) \triangleq \mathbf{K}(\mathbf{q}) \triangleq \begin{bmatrix}\mathbf{K}_{11}(\mathbf{q}) & \mathbf{K}_{12}(\mathbf{q})\\ \mathbf{K}_{21}(\mathbf{q}) & \mathbf{K}_{22}(\mathbf{q})\end{bmatrix} \tag{5.8-42b}$$

$$\mathbf{D}^{-1}(\mathbf{q}) \triangleq \mathbf{E}(\mathbf{q}) \triangleq \begin{bmatrix}\mathbf{E}_{11}(\mathbf{q}) & \mathbf{E}_{12}(\mathbf{q})\\ \mathbf{E}_{21}(\mathbf{q}) & \mathbf{E}_{22}(\mathbf{q})\end{bmatrix} \tag{5.8-43a}$$

$$\mathbf{h}(\mathbf{q},\dot{\mathbf{q}}) \triangleq \begin{bmatrix}\mathbf{h}_1(\mathbf{q},\dot{\mathbf{q}})\\ \mathbf{h}_2(\mathbf{q},\dot{\mathbf{q}})\end{bmatrix} \tag{5.8-43b}$$

$$\mathbf{c}(\mathbf{q}) \triangleq \begin{bmatrix}\mathbf{c}_1(\mathbf{q})\\ \mathbf{c}_2(\mathbf{q})\end{bmatrix} \tag{5.8-44a}$$

$$\boldsymbol{\tau}(t) \triangleq \begin{bmatrix}\tau_1(t)\\ \tau_2(t)\end{bmatrix} \tag{5.8-44b}$$

Combining Eqs. (5.7-4), (5.7-8), and (5.8-41), and using Eqs. (5.8-42) to (5.8-44), we can obtain the state equations of the manipulator in cartesian coordinates:

$$\begin{bmatrix}\dot{\mathbf{p}}(t)\\ \dot{\boldsymbol{\Phi}}(t)\\ \dot{\mathbf{v}}(t)\\ \dot{\boldsymbol{\Omega}}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}(\boldsymbol{\Phi})\\ \mathbf{0} & \mathbf{0} & \dot{\mathbf{N}}_{11}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{11}(\mathbf{q})+\dot{\mathbf{N}}_{12}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{21}(\mathbf{q}) & \dot{\mathbf{N}}_{11}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{12}(\mathbf{q})+\dot{\mathbf{N}}_{12}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{22}(\mathbf{q})\\ \mathbf{0} & \mathbf{0} & \dot{\mathbf{N}}_{21}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{11}(\mathbf{q})+\dot{\mathbf{N}}_{22}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{21}(\mathbf{q}) & \dot{\mathbf{N}}_{21}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{12}(\mathbf{q})+\dot{\mathbf{N}}_{22}(\mathbf{q},\dot{\mathbf{q}})\mathbf{K}_{22}(\mathbf{q})\end{bmatrix}$$

$$\times\begin{bmatrix}\mathbf{p}(t)\\ \boldsymbol{\Phi}(t)\\ \mathbf{v}(t)\\ \boldsymbol{\Omega}(t)\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\\ \mathbf{N}_{11}(\mathbf{q})\mathbf{E}_{11}(\mathbf{q})+\mathbf{N}_{12}(\mathbf{q})\mathbf{E}_{21}(\mathbf{q}) & \mathbf{N}_{11}(\mathbf{q})\mathbf{E}_{12}(\mathbf{q})+\mathbf{N}_{12}(\mathbf{q})\mathbf{E}_{22}(\mathbf{q})\\ \mathbf{N}_{21}(\mathbf{q})\mathbf{E}_{11}(\mathbf{q})+\mathbf{N}_{22}(\mathbf{q})\mathbf{E}_{21}(\mathbf{q}) & \mathbf{N}_{21}(\mathbf{q})\mathbf{E}_{12}(\mathbf{q})+\mathbf{N}_{22}(\mathbf{q})\mathbf{E}_{22}(\mathbf{q})\end{bmatrix} \times$$

(continued on next page)

$$\times \begin{bmatrix} -\mathbf{h}_1(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{c}_1(\mathbf{q}) + \tau_1(t) \\ \\ -\mathbf{h}_2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{c}_2(\mathbf{q}) + \tau_2(t) \end{bmatrix} \qquad (5.8\text{-}45)$$

where $\mathbf{0}$ is a 3×3 zero matrix. It is noted that the leftmost and middle vectors are 12×1, the center left matrix is 12×12, the right matrix is 12×6, and the rightmost vector is 6×1. Equation (5.8-45) represents the state equations of the manipulator and will be used to derive an adaptive control scheme in cartesian coordinates.

Defining the state vector for the manipulator hand as

$$\begin{aligned} \mathbf{x}(t) &\triangleq (x_1, x_2, \ldots, x_{12})^T \\ &\triangleq (p_x, p_y, p_z, \alpha, \beta, \gamma, v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T \\ &\triangleq (\mathbf{p}^T, \Phi^T, \mathbf{v}^T, \Omega^T)^T \end{aligned} \qquad (5.8\text{-}46)$$

and the input torque vector as

$$\mathbf{u}(t) \triangleq (\tau_1, \ldots, \tau_6)^T \triangleq (u_1, \ldots, u_6)^T \qquad (5.8\text{-}47)$$

Eq. (5.8-45) can be expressed in state space representation as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t)] \qquad (5.8\text{-}48)$$

where $\mathbf{x}(t)$ is a $2n$-dimensional vector, $\mathbf{u}(t)$ is an n-dimensional vector, $\mathbf{f}(\cdot)$ is a $2n \times 1$ continuously differentiable, nonlinear vector-valued function, and $n = 6$ is the number of degrees of freedom of the manipulator.

Equation (5.8-48) can be expressed as

$$\begin{aligned} \dot{x}_1(t) &= f_1(\mathbf{x}, \mathbf{u}) = x_7(t) \\ \dot{x}_2(t) &= f_2(\mathbf{x}, \mathbf{u}) = x_8(t) \\ \dot{x}_3(t) &= f_3(\mathbf{x}, \mathbf{u}) = x_9(t) \\ \dot{x}_4(t) &= f_4(\mathbf{x}, \mathbf{u}) = -\sec x_5 (x_{10} \cos x_6 + x_{11} \sin x_6) \qquad (5.8\text{-}49) \\ \dot{x}_5(t) &= f_5(\mathbf{x}, \mathbf{u}) = \sec x_5 (x_{10} \cos x_5 \sin x_6 - x_{11} \cos x_5 \cos x_6) \\ \dot{x}_6(t) &= f_6(\mathbf{x}, \mathbf{u}) = -\sec x_5 (x_{10} \sin x_5 \cos x_6 + x_{11} \sin x_5 \sin x_6 + x_{12} \cos x_5) \end{aligned}$$

$$\begin{aligned} \dot{x}_{i+6}(t) &= f_{i+6}(\mathbf{x}, \mathbf{u}) \\ &= g_{i+6}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{x}(t) + b_{i+6}(\mathbf{q})\boldsymbol{\lambda}(\mathbf{q}, \dot{\mathbf{q}}) + b_{i+6}(\mathbf{q})\mathbf{u}(t) \end{aligned}$$

where $i = 1, \ldots, 6$ and $g_{i+6}(\mathbf{q}, \dot{\mathbf{q}})$ is the $(i + 6)$th row of the matrix:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}(\boldsymbol{\Phi}) \\ \mathbf{0} & \mathbf{0} & \dot{\mathbf{N}}_{11}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{11}(\mathbf{q}) + \dot{\mathbf{N}}_{12}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{21}(\mathbf{q}) & \dot{\mathbf{N}}_{11}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{12}(\mathbf{q}) + \dot{\mathbf{N}}_{12}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{22}(\mathbf{q}) \\ \mathbf{0} & \mathbf{0} & \dot{\mathbf{N}}_{21}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{11}(\mathbf{q}) + \dot{\mathbf{N}}_{22}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{21}(\mathbf{q}) & \dot{\mathbf{N}}_{21}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{12}(\mathbf{q}) + \dot{\mathbf{N}}_{22}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{K}_{22}(\mathbf{q}) \end{bmatrix}$$

and $b_{i+6}(\mathbf{q})$ is the $(i + 6)$th row of the matrix:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{N}_{11}(\mathbf{q})\mathbf{E}_{11}(\mathbf{q}) + \mathbf{N}_{12}(\mathbf{q})\mathbf{E}_{21}(\mathbf{q}) & \mathbf{N}_{11}(\mathbf{q})\mathbf{E}_{12}(\mathbf{q}) + \mathbf{N}_{12}(\mathbf{q})\mathbf{E}_{22}(\mathbf{q}) \\ \mathbf{N}_{21}(\mathbf{q})\mathbf{E}_{11}(\mathbf{q}) + \mathbf{N}_{22}(\mathbf{q})\mathbf{E}_{21}(\mathbf{q}) & \mathbf{N}_{21}(\mathbf{q})\mathbf{E}_{12}(\mathbf{q}) + \mathbf{N}_{22}(\mathbf{q})\mathbf{E}_{22}(\mathbf{q}) \end{bmatrix}$$

and

$$\boldsymbol{\lambda}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\mathbf{h}_1(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{c}_1(\mathbf{q}) \\ \\ -\mathbf{h}_2(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{c}_2(\mathbf{q}) \end{bmatrix}$$

Equation (5.8-49) describes the complete manipulator dynamics in cartesian coordinates, and the control problem is to find a feedback control law $\mathbf{u}(t) = \mathbf{g}[\mathbf{x}(t)]$ to minimize the manipulator hand error along the desired hand trajectory over a wide range of payloads. Again, perturbation theory is used and Taylor series expansion is applied to Eq. (5.8-49) to obtain the associated linearized system and to design a feedback control law about the desired hand trajectory. The determination of the feedback control law for the linearized system is identical to the one in the joint coordinates [Eqs. (5.8-32) to (5.8-34) and Eq. (5.8-39)]. The resolved motion adaptive control block diagram is shown in Fig. 5.19.

The overall resolved motion adaptive control system is again characterized by a feedforward component and a feedback component. Such a formulation has the advantage of employing parallel schemes in computing these components. The feedforward component computes the desired joint torques as follows: (1) The hand trajectory set points $\mathbf{p}^d(t)$, $\boldsymbol{\Phi}^d(t)$, $\mathbf{v}^d(t)$, $\boldsymbol{\Omega}^d(t)$, $\dot{\mathbf{v}}^d(t)$, and $\dot{\boldsymbol{\Omega}}^d(t)$ are resolved into a set of values of desired joint positions, velocities, and accelerations; (2) the desired joint torques along the hand trajectory are computed from the Newton-Euler equations of motion using the computed sets of values of joint positions, velocities, and accelerations. These computed torques constitute the nominal torque values $\mathbf{u}_n(t)$. The feedback component computes the perturbation joint torques $\delta\mathbf{u}(t)$ the same way as in Eq. (5.8-39), using the recursive least-squares identification scheme in Eqs. (5.8-32) to (5.8-34).

A feasibility study of implementing the adaptive controller based on a 60-Hz sampling frequency and using present-day low-cost microprocessors can be conducted by looking at the computational requirements in terms of mathematical mul-

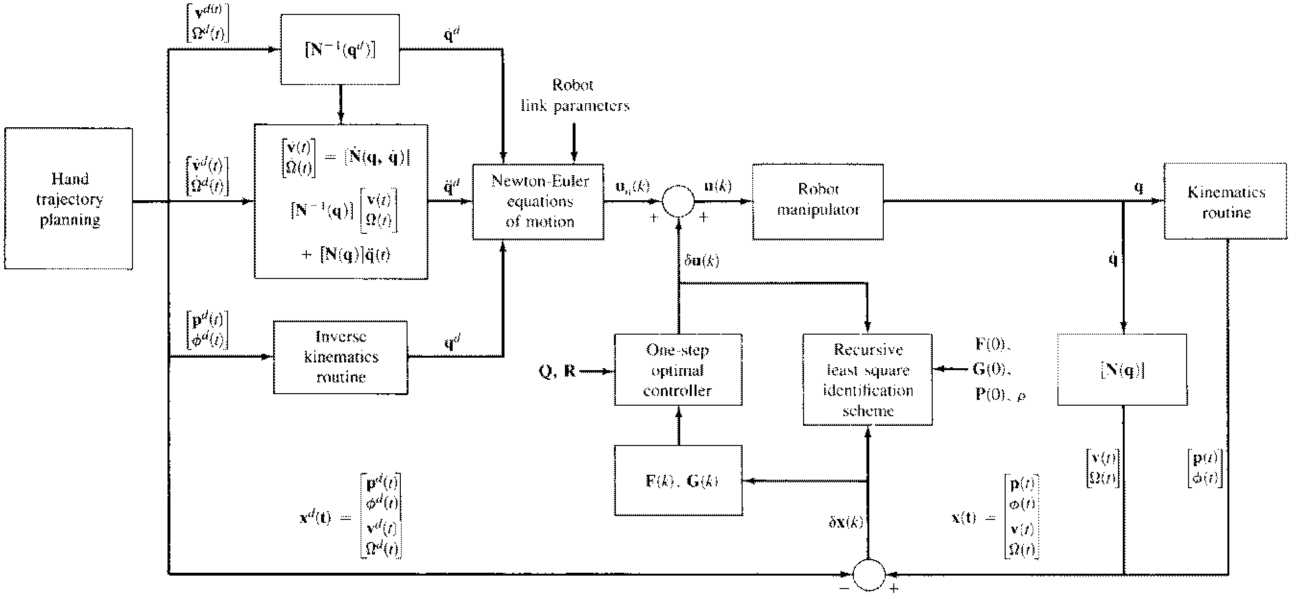
Hand trajectory planning
$\begin{bmatrix} \mathbf{v}^d(t) \\ \Omega^d(t) \end{bmatrix}$
$[\mathbf{N}^{-1}(\mathbf{q}^d)]$
$\dot{\mathbf{q}}^d$
$\begin{bmatrix} \dot{\mathbf{v}}^d(t) \\ \dot{\Omega}^d(t) \end{bmatrix}$
$\begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\Omega}(t) \end{bmatrix} = [\dot{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})]$
$[\mathbf{N}^{-1}(\mathbf{q})] \begin{bmatrix} \mathbf{v}(t) \\ \Omega(t) \end{bmatrix}$
$+ [\mathbf{N}(\mathbf{q})]\ddot{\mathbf{q}}(t)$
$\ddot{\mathbf{q}}^d$
Robot link parameters
Newton-Euler equations of motion
$\begin{bmatrix} \mathbf{p}^d(t) \\ \phi^d(t) \end{bmatrix}$
Inverse kinematics routine
$\mathbf{q}^d$
$\mathbf{u}_n(k)$
+
+
$\mathbf{u}(k)$
$\delta\mathbf{u}(k)$
Robot manipulator
$\mathbf{q}$
Kinematics routine
$\dot{\mathbf{q}}$
Q, R
One-step optimal controller
Recursive least square identification scheme
F(0), G(0), P(0), ρ
$[\mathbf{N}(\mathbf{q})]$
F(k), G(k)
$\begin{bmatrix} \mathbf{v}(t) \\ \Omega(t) \end{bmatrix}$
$\begin{bmatrix} \mathbf{p}(t) \\ \phi(t) \end{bmatrix}$
$\mathbf{x}^d(\mathbf{t}) = \begin{bmatrix} \mathbf{p}^d(t) \\ \phi^d(t) \\ \mathbf{v}^d(t) \\ \Omega^d(t) \end{bmatrix}$
$\delta\mathbf{x}(k)$
$\mathbf{x}(\mathbf{t}) = \begin{bmatrix} \mathbf{p}(t) \\ \phi(t) \\ \mathbf{v}(t) \\ \Omega(t) \end{bmatrix}$
−
+

Figure 5.19 The resolved motion adaptive control.

tiplication and addition operations. We assume that multiprocessors are available for parallel computation of the controller. The feedforward component which computes the nominal joint torques along a desired hand trajectory can be computed serially in four separate stages. It requires a total of 1386 multiplications and 988 additions for a six-joint manipulator. The feedback control component which computes the perturbation joint torques can be conveniently computed serially in three separate stages. It requires about 3348 multiplications and 3118 additions for a six-joint manipulator. Since the feedforward and feedback components can be computed in parallel, the resolved motion adaptive control requires a total of 3348 multiplications and 3118 additions in each sampling period. Computational requirements in terms of multiplications and additions for the adaptive controller for a n-joint manipulator are tabulated in Table 5.3.

Based on the specification sheet of an INTEL 8087 microprocessor, an integer multiply requires 19 μs, an addition requires 17 μs, and a memory fetch or store requires 9 μs. Assuming that two memory fetches are required for each multiplication and addition operation, the proposed controller can be computed in about 233 ms which is not fast enough for closing the servo loop. (Recall from Sec. 5.3.5 that a minimum of 16 msec is required if the sampling frequency is 60 Hz). Similarly, looking at the specification sheet of a Motorola MC68000 microprocessor, an integer multiply requires 5.6 μs, an addition requires 0.96 μs, and a memory fetch or store requires 0.32 μs, the proposed controller can be computed in about 26.24 ms which is still not fast enough for closing the servo loop. Finally, looking at the specification sheet of a PDP 11/45 computer, an integer multiply requires 3.3 μs, an addition requires 300 ns, and a memory fetch or store requires 450 ns, the proposed controller can be computed in about 18 ms which translates to a sampling frequency of approximately 55 Hz. However, the PDP 11/45 is a uniprocessor machine and the parallel computation assumption is not valid. This exercise should give the reader an idea of the required processing speed for adaptive control of a manipulator. We anticipate that faster microprocessors, which will be able to compute the proposed resolved motion adaptive controller within 10 ms, will be available in a few years.

5.9 CONCLUDING REMARKS

We have reviewed various robot manipulator control methods. They vary from a simple servomechanism to advanced control schemes such as adaptive control with an identification algorithm. The control techniques are discussed in joint motion control, resolved motion control, and adaptive control. Most of the joint motion and resolved motion control methods discussed servo the arm at the hand or the joint level and emphasize nonlinear compensations of the coupling forces among the various joints. We have also discussed various adaptive control strategies. The model-referenced adaptive control is easy to implement, but suitable reference models are difficult to choose and it is difficult to establish any stability analysis of

Table 5.3 Computations of the resolved motion adaptive control†

	Adaptive controller	Number of multiplications	Number of additions
stage 1	Compute $\mathbf{q}^d$ (inverse kinematics)	(39)	(32)
stage 2	Compute $\dot{\mathbf{q}}^d$	$n^2 + 27n + 327$ (525)	$n^2 + 18n + 89$ (233)
stage 3	Compute $\ddot{\mathbf{q}}^d$	$4n^2$ (144)	$4n^2 - 3n$ (126)
stage 4	Compute $\boldsymbol{\tau}$	$117n - 24$ (678)	$103n - 21$ (597)
Total feedforward computations		$5n^2 + 144n + 342$ (1386)	$5n^2 + 118n + 100$ (988)
stage 1	Compute $(\mathbf{p}^T\boldsymbol{\Phi}^T)^T$	(48)	(22)
	Compute $(\mathbf{v}^T\boldsymbol{\Omega}^T)^T$	$n^2 + 27n - 21$ (177)	$n^2 + 18n - 15$ (129)
stage 2	Compute hand errors $[\mathbf{x}(k) - \mathbf{x}_n(k)]$	0 (0)	2n (12)
	Identification scheme	$33n^2 + 9n + 2$ (1244)	$34\frac{1}{2}n^2 - 1\frac{1}{2}n$ (1233)
stage 3	Compute adaptive controller	$8n^3 + 4n^2 + n + 1$ (1879)	$8n^3 - n$ (1722)
Total feedback computations		$8n^3 + 38n^2 + 37n + 30$ (3348)	$8n^3 + 35\frac{1}{2}n^2 + 17\frac{1}{2}n + 7$ (3118)
Total mathematical operations		$8n^3 + 38n^2 + 37n + 30$ (3348)	$8n^3 + 35\frac{1}{2}n^2 + 17\frac{1}{2}n + 7$ (3118)

† Number inside parentheses indicate computations for $n = 6$.

the controlled system. Self-tuning adaptive control fits the input-output data of the system with an autoregressive model. Both methods neglect the coupling forces between the joints which may be severe for manipulators with rotary joints. Adaptive control using perturbation theory may be more appropriate for various manipulators because it takes all the interaction forces between the joints into consideration. The adaptive perturbation control strategy was found suitable for controlling the manipulator in both the joint coordinates and cartesian coordinates. An adaptive perturbation control system is characterized by a feedforward component and a feedback component which can be computed separately and simultaneously in parallel. The computations of the adaptive control for a six-link robot arm may be implemented in low-cost microprocessors for controlling in the joint variable space, while the resolved motion adaptive control cannot be implemented in present-day low-cost microprocessors because they still do not have the required speed to compute the controller parameters for the "standard" 60-Hz sampling frequency.

REFERENCES

Further readings on computed torque control techniques can be found in Paul [1972], Bejczy [1974], Markiewicz [1973], Luh et al. [1980*b*], and Lee [1982]. Minimum-time control can be found in Kahn and Roth [1971], and minimum-time control with torque constraint is discussed by Bobrow and Dubowsky [1983]. Young [1978] discusses the design of a variable structure control for the control of manipulators. More general theory in variable structure control can be found in Utkin [1977] and Itkis [1976]. Various researchers have discussed nonlinear decoupled control, including Falb and Wolovich [1967], Hemami and Camana [1976], Saridis and Lee [1979], Horowitz and Tomizuka [1980], Freund [1982], Tarn et al. [1984], and Gilbert and Ha [1984].

Further readings on resolved motion control can be found in Whitney [1969, 1972] who discussed resolved motion rate control. Luh et al. [1980*b*] extended this concept to include resolved acceleration control. The disadvantage of resolved motion control lies in the fact that the inverse jacobian matrix requires intensive computations.

In order to compensate for the varying parameters of a manipulator and the changing loads that it carries, various adaptive control schemes, both in joint and cartesian coordinates, have been developed. These adaptive control schemes can be found in Dubowsky and DesForges [1979], Horowitz and Tomizuka [1980], Koivo and Guo [1983], Lee and Chung [1984, 1985], Lee and Lee [1984], and Lee et al. [1984].

An associated problem relating to control is the investigation of efficient control system architectures for computing the control laws within the required servo time. Papers written by Lee et al. [1982], Luh and Lin [1982], Orin [1984], Nigam and Lee [1985], and Lee and Chang [1986*b*], are oriented toward this goal.

PROBLEMS

5.1 Consider the development of a single-joint positional controller, as discussed in Sec. 5.3.2. If the applied voltage $v_a(t)$ is linearly proportional to the position error and to the rate of the output angular position, what is the open-loop transfer function $\Theta_L(s)/E(s)$ and the closed-loop transfer function $\Theta_L(s)/\Theta_L^d(s)$ of the system?

5.2 For the applied voltage used in Prob. 5.1, discuss the steady-state error of the system due to a step input. Repeat for a ramp input.

5.3 In the computed torque control technique, if the Newton-Euler equations of motion are used to compute the applied joint torques for a 6 degree-of-freedom manipulator with rotary joints, what is the required number of multiplications and additions per trajectory set point?

5.4 In the computed torque control technique, the analysis is performed in the continuous time, while the actual control on the robot arm is done in discrete time (i.e., by a sampled-data system) because we use a digital computer for implementing the controller. Explain the condition under which this practice is valid.

5.5 The equations of motion of the two-link robot arm in Sec. 3.2.6 can be written in a compact matrix-vector form as:

$$\begin{bmatrix} d_{11}(\theta_2) & d_{12}(\theta_2) \\ d_{12}(\theta_2) & d_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1(t) \\ \ddot{\theta}_2(t) \end{bmatrix} + \begin{bmatrix} \beta_{12}(\theta_2)\dot{\theta}_2^2 + 2\beta_{12}(\theta_2)\dot{\theta}_1\dot{\theta}_2 \\ -\beta_{12}(\theta_2)\dot{\theta}_1^2 \end{bmatrix} + \begin{bmatrix} c_1(\theta_1, \theta_2)g \\ c_2(\theta_1, \theta_2)g \end{bmatrix} = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \end{bmatrix}$$

where g is the gravitational constant. Choose an appropriate state variable vector $\mathbf{x}(t)$ and a control vector $\mathbf{u}(t)$ for this dynamic system. Assuming that $\mathbf{D}^{-1}(\boldsymbol{\theta})$ exists, express the equations of motion of this robot arm explicitly in terms of d_{ij}'s, β_{ij}'s, and c_i's in a state-space representation with the chosen state-variable vector and control vector.

5.6 Design a variable structure controller for the robot in Prob. 5.5. (See Sec. 5.5.)

5.7 Design a nonlinear decoupled feedback controller for the robot in Prob. 5.5. (See Sec. 5.6.)

5.8 Find the jacobian matrix in the base coordinate frame for the robot in Prob. 5.5. (See Appendix B.)

5.9 Give two main disadvantages of using the resolved motion rate control.

5.10 Give two main disadvantages of using the resolved motion acceleration control.

5.11 Give two main disadvantages of using the model-referenced adaptive control.

5.12 Give two main disadvantages of using the adaptive perturbation control.