8

Motion Control

In Chap. 4, trajectory planning techniques have been presented which allow the generation of the reference inputs to the motion control system. The problem of controlling a manipulator can be formulated as that to determine the time history of the generalized forces (forces or torques) to be developed by the joint actuators, so as to guarantee execution of the commanded task while satisfying given transient and steady-state requirements. The task may regard either the execution of specified motions for a manipulator operating in free space, or the execution of specified motions and contact forces for a manipulator whose end-effector is constrained by the environment. In view of problem complexity, the two aspects will be treated separately; first, motion control in free space, and then control of the interaction with the environment. The problem of *motion control* of a manipulator is the topic of this chapter. A number of *joint space* control techniques are presented. These can be distinguished between *decentralized control* schemes, i.e., when the single manipulator joint is controlled independently of the others, and *centralized control* schemes, i.e., when the dynamic interaction effects between the joints are taken into account. Finally, as a premise to the interaction control problem, the basic features of *operational space* control schemes are illustrated.

8.1 The Control Problem

Several techniques can be employed for controlling a manipulator. The technique followed, as well as the way it is implemented, may have a significant influence on the manipulator performance and then on the possible range of applications. For instance, the need for trajectory tracking control in the operational space may lead to hardware/software implementations, which differ from those allowing point-to-point control, where only reaching of the final position is of concern.

On the other hand, the manipulator mechanical design has an influence on the kind of control scheme utilized. For instance, the control problem of

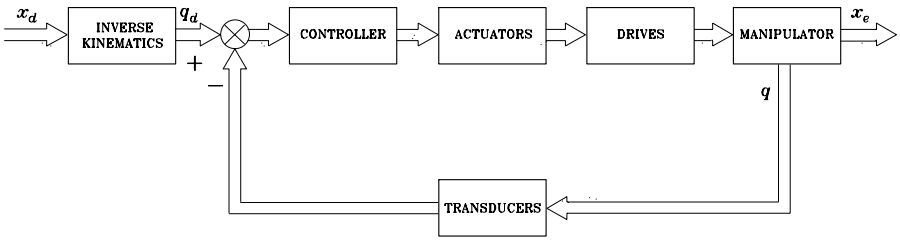

Fig. 8.1. General scheme of joint space control

a Cartesian manipulator is substantially different from that of an anthropomorphic manipulator.

The driving system of the joints also has an effect on the type of control strategy used. If a manipulator is actuated by electric motors with reduction gears of high ratios, the presence of gears tends to linearize system dynamics, and thus to decouple the joints in view of the reduction of nonlinearity effects. The price to pay, however, is the occurrence of joint friction, elasticity and backlash that may limit system performance more than it is due to configuration-dependent inertia, Coriolis and centrifugal forces, and so forth. On the other hand, a robot actuated with direct drives eliminates the drawbacks due to friction, elasticity and backlash, but the weight of nonlinearities and couplings between the joints becomes relevant. As a consequence, different control strategies have to be thought of to obtain high performance.

Without any concern to the specific type of mechanical manipulator, it is worth remarking that task specification (end-effector motion and forces) is usually carried out in the operational space, whereas control actions (joint actuator generalized forces) are performed in the joint space. This fact naturally leads to considering two kinds of general control schemes, namely, a *joint space control* scheme (Fig. 8.1) and an *operational space control* scheme (Fig. 8.2). In both schemes, the control structure has closed loops to exploit the good features provided by feedback, i.e., robustness to modelling uncertainties and reduction of disturbance effects. In general terms, the following considerations should be made.

The *joint space control* problem is actually articulated in two subproblems. First, manipulator inverse kinematics is solved to transform the motion requirements $\boldsymbol{x}_d$ from the operational space into the corresponding motion $\boldsymbol{q}_d$ in the joint space. Then, a joint space control scheme is designed that allows the actual motion $\boldsymbol{q}$ to track the reference inputs. However, this solution has the drawback that a joint space control scheme does not influence the operational space variables $\boldsymbol{x}_e$ which are controlled in an open-loop fashion through the manipulator mechanical structure. It is then clear that any uncertainty of the structure (construction tolerance, lack of calibration, gear backlash, elasticity) or any imprecision in the knowledge of the end-effector pose relative

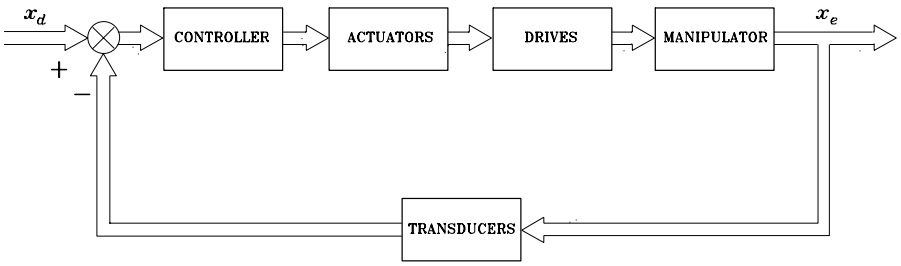

Fig. 8.2. General scheme of operational space control

to an object to manipulate causes a loss of accuracy on the operational space variables.

The *operational space control* problem follows a global approach that requires a greater algorithmic complexity; notice that inverse kinematics is now embedded into the feedback control loop. Its conceptual advantage regards the possibility of acting directly on operational space variables; this is somewhat only a potential advantage, since measurement of operational space variables is often performed not directly, but through the evaluation of direct kinematics functions starting from measured joint space variables.

On the above premises, in the following, joint space control schemes for manipulator motion in the free space are presented first. In the sequel, operational space control schemes will be illustrated which are logically at the basis of control of the interaction with the environment.

8.2 Joint Space Control

In Chap. 7, it was shown that the equations of motion of a manipulator in the absence of external end-effector forces and, for simplicity, of static friction (difficult to model accurately) are described by

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{8.1}$$

with obvious meaning of the symbols. To control the motion of the manipulator in free space means to determine the n components of generalized forces — torques for revolute joints, forces for prismatic joints — that allow execution of a motion $\boldsymbol{q}(t)$ so that

$$\boldsymbol{q}(t) = \boldsymbol{q}_d(t),$$

as closely as possible, where $\boldsymbol{q}_d(t)$ denotes the vector of desired joint trajectory variables.

The generalized forces are supplied by the actuators through proper transmissions to transform the motion characteristics. Let $\boldsymbol{q}_m$ denote the vector of joint actuator displacements; the transmissions — assumed to be rigid and with no backlash — establish the following relationship:

$$\boldsymbol{K}_r\boldsymbol{q} = \boldsymbol{q}_m, \tag{8.2}$$

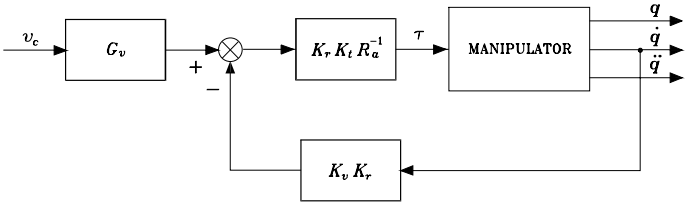

Fig. 8.3. Block scheme of the manipulator and drives system as a voltage-controlled system

where $\boldsymbol{K}_r$ is an $(n \times n)$ diagonal matrix, whose elements are defined in (7.22) and are much greater than unity.[1]

In view of (8.2), if $\boldsymbol{\tau}_m$ denotes the vector of actuator driving torques, one can write

$$\boldsymbol{\tau}_m = \boldsymbol{K}_r^{-1}\boldsymbol{\tau}. \tag{8.3}$$

With reference to (5.1)–(5.4), the n driving systems can be described in compact matrix form by the equations:

$$\boldsymbol{K}_r^{-1}\boldsymbol{\tau} = \boldsymbol{K}_t\boldsymbol{i}_a \tag{8.4}$$

$$\boldsymbol{v}_a = \boldsymbol{R}_a\boldsymbol{i}_a + \boldsymbol{K}_v\dot{\boldsymbol{q}}_m \tag{8.5}$$

$$\boldsymbol{v}_a = \boldsymbol{G}_v\boldsymbol{v}_c. \tag{8.6}$$

In (8.4), $\boldsymbol{K}_t$ is the diagonal matrix of torque constants and $\boldsymbol{i}_a$ is the vector of armature currents of the n motors; in (8.5), $\boldsymbol{v}_a$ is the vector of armature voltages, $\boldsymbol{R}_a$ is the diagonal matrix of armature resistances,[2] and $\boldsymbol{K}_v$ is the diagonal matrix of voltage constants of the n motors; in (8.6), $\boldsymbol{G}_v$ is the diagonal matrix of gains of the n amplifiers and $\boldsymbol{v}_c$ is the vector of control voltages of the n servomotors.

On reduction of (8.1), (8.2), (8.4), (8.5), (8.6), the dynamic model of the system given by the manipulator and drives is described by

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \tag{8.7}$$

where the following positions have been made:

$$\boldsymbol{F} = \boldsymbol{F}_v + \boldsymbol{K}_r\boldsymbol{K}_t\boldsymbol{R}_a^{-1}\boldsymbol{K}_v\boldsymbol{K}_r \tag{8.8}$$

$$\boldsymbol{u} = \boldsymbol{K}_r\boldsymbol{K}_t\boldsymbol{R}_a^{-1}\boldsymbol{G}_v\boldsymbol{v}_c. \tag{8.9}$$

From (8.1), (8.7), (8.8), (8.9) it is

$$\boldsymbol{K}_r\boldsymbol{K}_t\boldsymbol{R}_a^{-1}\boldsymbol{G}_v\boldsymbol{v}_c = \boldsymbol{\tau} + \boldsymbol{K}_r\boldsymbol{K}_t\boldsymbol{R}_a^{-1}\boldsymbol{K}_v\boldsymbol{K}_r\dot{\boldsymbol{q}} \tag{8.10}$$

[1] Assuming a diagonal $\boldsymbol{K}_r$ leads to excluding the presence of kinematic couplings in the transmission, that is the motion of each actuator does not induce motion on a joint other than that actuated.

[2] The contribution of the inductance has been neglected.

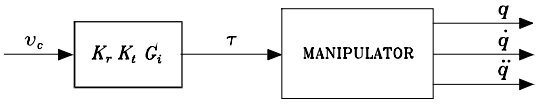

Fig. 8.4. Block scheme of the manipulator and drives system as a torque-controlled system

and thus

$$\boldsymbol{\tau} = \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{R}_a^{-1} (\boldsymbol{G}_v \boldsymbol{v}_c - \boldsymbol{K}_v \boldsymbol{K}_r \dot{\boldsymbol{q}}). \tag{8.11}$$

The overall system is then *voltage-controlled* and the corresponding block scheme is illustrated in Fig. 8.3. If the following assumptions hold:

- the elements of matrix $\boldsymbol{K}_r$, characterizing the transmissions, are much greater than unity;
- the elements of matrix $\boldsymbol{R}_a$ are very small, which is typical in the case of high-efficiency servomotors;
- the values of the torques $\boldsymbol{\tau}$ required for the execution of the desired motions are not too large;

then it can be assumed that

$$\boldsymbol{G}_v \boldsymbol{v}_c \approx \boldsymbol{K}_v \boldsymbol{K}_r \dot{\boldsymbol{q}}. \tag{8.12}$$

The proportionality relationship obtained between $\dot{\boldsymbol{q}}$ and $\boldsymbol{v}_c$ is independent of the values attained by the manipulator parameters; the smaller the joint velocities and accelerations, the more valid this assumption. Hence, velocity (or voltage) control shows an inherent robustness with respect to parameter variations of the manipulator model, which is enhanced by the values of the gear reduction ratios.

In this case, the scheme illustrated in Fig. 8.3 can be taken as the reference structure for the design of the control system. Having assumed that

$$\boldsymbol{v}_c \approx \boldsymbol{G}_v^{-1} \boldsymbol{K}_v \boldsymbol{K}_r \dot{\boldsymbol{q}} \tag{8.13}$$

implies that the velocity of the i-th joint depends only on the i-th control voltage, since the matrix $\boldsymbol{G}_v^{-1} \boldsymbol{K}_v \boldsymbol{K}_r$ is diagonal. Therefore, the joint position control system can be designed according to a *decentralized control structure*, since each joint can be controlled independently of the others. The results, evaluated in the terms of the tracking accuracy of the joint variables with respect to the desired trajectories, are improved in the case of higher gear reduction ratios and less demanding values of required speeds and accelerations.

On the other hand, if the desired manipulator motion requires large joint speeds and/or accelerations, the approximation (8.12) no longer holds, in view of the magnitude of the required driving torques; this occurrence is even more evident for direct-drive actuation ($\boldsymbol{K}_r = \boldsymbol{I}$).

In this case, by resorting to an inverse dynamics technique, it is possible to find the joint torques $\boldsymbol{\tau}(t)$ needed to track any specified motion in terms of the joint accelerations $\ddot{\boldsymbol{q}}(t)$, velocities $\dot{\boldsymbol{q}}(t)$ and positions $\boldsymbol{q}(t)$. Obviously, this solution requires the accurate knowledge of the manipulator dynamic model. The determination of the torques to be generated by the drive system can thus refer to a *centralized control structure*, since to compute the torque history at the i-th joint it is necessary to know the time evolution of the motion of all the joints. By recalling that

$$\boldsymbol{\tau} = \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{i}_a, \tag{8.14}$$

to find a relationship between the torques $\boldsymbol{\tau}$ and the control voltages $\boldsymbol{v}_c$, using (8.5), (8.6) leads to

$$\boldsymbol{\tau} = \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{R}_a^{-1} \boldsymbol{G}_v \boldsymbol{v}_c - \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{R}_a^{-1} \boldsymbol{K}_v \boldsymbol{K}_r \dot{\boldsymbol{q}}. \tag{8.15}$$

If the actuators have to provide torque contributions computed on the basis of the manipulator dynamic model, the control voltages — to be determined according to (8.15) — depend on the torque values and also on the joint velocities; this relationship depends on the matrices $\boldsymbol{K}_t$, $\boldsymbol{K}_v$ and $\boldsymbol{R}_a^{-1}$, whose elements are influenced by the operating conditions of the motors. To reduce sensitivity to parameter variations, it is worth considering driving systems characterized by a current control rather than by a voltage control. In this case the actuators behave as torque-controlled generators; the equation in (8.5) becomes meaningless and is replaced by

$$\boldsymbol{i}_a = \boldsymbol{G}_i \boldsymbol{v}_c, \tag{8.16}$$

which gives a proportional relation between the armature currents $\boldsymbol{i}_a$ (and thus the torques $\boldsymbol{\tau}$) and the control voltages $\boldsymbol{v}_c$ established by the constant matrix $\boldsymbol{G}_i$. As a consequence, (8.9) becomes

$$\boldsymbol{\tau} = \boldsymbol{u} = \boldsymbol{K}_r \boldsymbol{K}_t \boldsymbol{G}_i \boldsymbol{v}_c \tag{8.17}$$

which shows a reduced dependence of $\boldsymbol{u}$ on the motor parameters. The overall system is now *torque-controlled* and the resulting block scheme is illustrated in Fig. 8.4.

The above presentation suggests resorting for the decentralized structure — where the need for robustness prevails — to feedback control systems, while for the centralized structure — where the computation of inverse dynamics is needed — it is necessary to refer to control systems with feedforward actions. Nevertheless, it should be pointed out that centralized control still requires the use of error contributions between the desired and the actual trajectory, no matter whether they are implemented in a feedback or in a feedforward fashion. This is a consequence of the fact that the considered dynamic model, even though a quite complex one, is anyhow an idealization of reality which

does not include effects, such as joint Coulomb friction, gear backlash, dimension tolerance, and the simplifying assumptions in the model, e.g., link rigidity, and so on.

As already pointed out, the drive systems is anyhow inserted into a feedback control system. In the case of decentralized control, the drive will be characterized by the model describing its behaviour as a velocity-controlled generator. Instead, in the case of centralized control, since the driving torque is to be computed on a complete or reduced manipulator dynamic model, the drive will be characterized as a torque-controlled generator.

8.3 Decentralized Control

The simplest control strategy that can be thought of is one that regards the manipulator as formed by n independent systems (the n joints) and controls each joint axis as a *single-input/single-output system*. Coupling effects between joints due to varying configurations during motion are treated as *disturbance* inputs.

In order to analyze various control schemes and their performance, it is worth considering the model of the system manipulator with drives in terms of mechanical quantities at the motor side; in view of (8.2), (8.3), it is

$$\boldsymbol{K}_r^{-1}\boldsymbol{B}(\boldsymbol{q})\boldsymbol{K}_r^{-1}\ddot{\boldsymbol{q}}_m + \boldsymbol{K}_r^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{K}_r^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{K}_r^{-1}\boldsymbol{F}_v\boldsymbol{K}_r^{-1} + \boldsymbol{K}_r^{-1}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m. \tag{8.18}$$

By observing that the diagonal elements of $\boldsymbol{B}(\boldsymbol{q})$ are formed by constant terms and configuration-dependent terms (functions of sine and cosine for revolute joints), one can set

$$\boldsymbol{B}(\boldsymbol{q}) = \bar{\boldsymbol{B}} + \Delta\boldsymbol{B}(\boldsymbol{q}) \tag{8.19}$$

where $\bar{\boldsymbol{B}}$ is the *diagonal* matrix whose constant elements represent the resulting average inertia at each joint. Substituting (8.19) into (8.1) yields

$$\boldsymbol{K}_r^{-1}\bar{\boldsymbol{B}}\boldsymbol{K}_r^{-1}\ddot{\boldsymbol{q}}_m + \boldsymbol{F}_m\dot{\boldsymbol{q}}_m + \boldsymbol{d} = \boldsymbol{\tau}_m \tag{8.20}$$

where

$$\boldsymbol{F}_m = \boldsymbol{K}_r^{-1}\boldsymbol{F}_v\boldsymbol{K}_r^{-1} \tag{8.21}$$

represents the matrix of viscous friction coefficients about the motor axes, and

$$\boldsymbol{d} = \boldsymbol{K}_r^{-1}\Delta\boldsymbol{B}(\boldsymbol{q})\boldsymbol{K}_r^{-1}\ddot{\boldsymbol{q}}_m + \boldsymbol{K}_r^{-1}\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{K}_r^{-1}\dot{\boldsymbol{q}}_m + \boldsymbol{K}_r^{-1}\boldsymbol{g}(\boldsymbol{q}) \tag{8.22}$$

represents the contribution depending on the configuration.

As illustrated by the block scheme of Fig. 8.5, the system of manipulator with drives is actually constituted by two subsystems; one has $\boldsymbol{\tau}_m$ as input and $\boldsymbol{q}_m$ as output, the other has $\boldsymbol{q}_m$, $\dot{\boldsymbol{q}}_m$, $\ddot{\boldsymbol{q}}_m$ as inputs, and $\boldsymbol{d}$ as output. The former is *linear* and *decoupled*, since each component of $\boldsymbol{\tau}_m$ influences only the corresponding component of $\boldsymbol{q}_m$. The latter is *nonlinear* and *coupled*, since

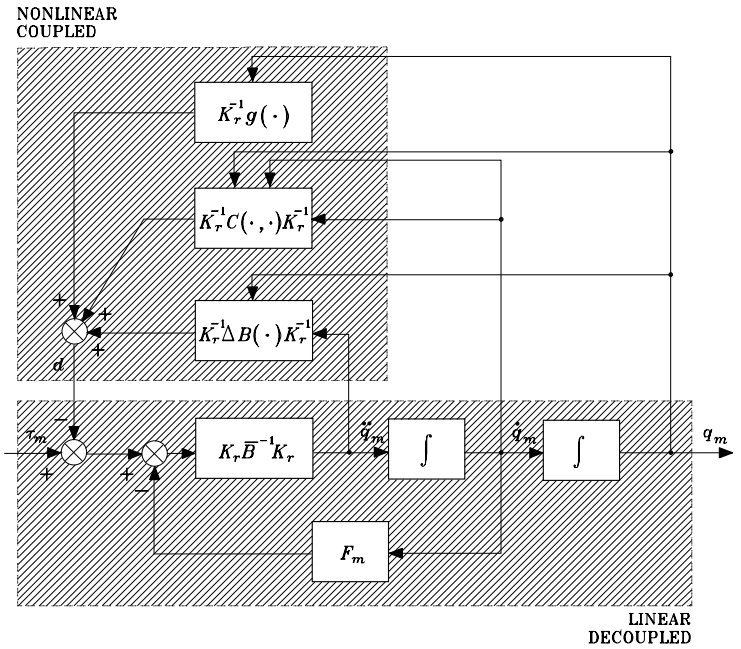

Fig. 8.5. Block scheme of the system of manipulator with drives

it accounts for all those nonlinear and coupling terms of manipulator joint dynamics.

On the basis of the above scheme, several control algorithms can be derived with reference to the detail of knowledge of the dynamic model. The simplest approach that can be followed, in case of high-gear reduction ratios and/or limited performance in terms of required velocities and accelerations, is to consider the component of the nonlinear interacting term $\boldsymbol{d}$ as a *disturbance* for the single joint servo.

The design of the control algorithm leads to a *decentralized control structure*, since each joint is considered independently of the others. The joint controller must guarantee good performance in terms of high disturbance rejection and enhanced trajectory tracking capabilities. The resulting control structure is substantially based on the error between the desired and actual output, while the input control torque at actuator i depends only on the error of output i.

Therefore, the system to control is Joint i drive corresponding to the single-input/single-output system of the decoupled and linear part of the scheme in Fig. 8.5. The interaction with the other joints is described by component i of the vector $\boldsymbol{d}$ in (8.22).

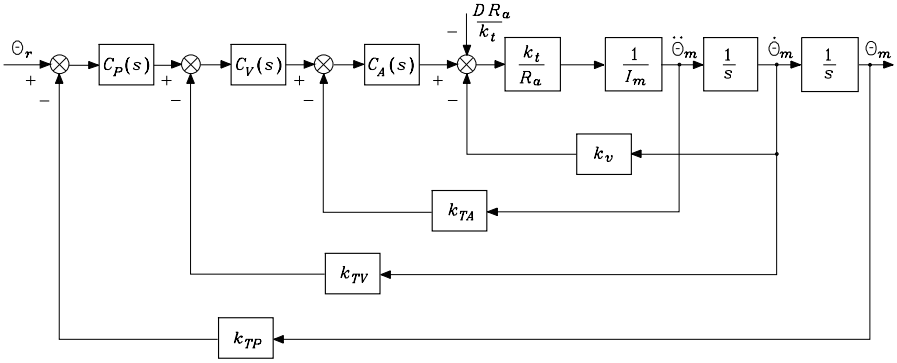

Fig. 8.6. Block scheme of general independent joint control

Assumed that the actuator is a rotary electric DC motor, the general scheme of drive control is that in Fig. 5.9 where I_m is the average inertia reported to the motor axis ($I_{mi} = \bar{b}_{ii}/k_{ri}^2$).[3]

8.3.1 Independent Joint Control

To guide selection of the controller structure, start noticing that an effective rejection of the disturbance d on the output ϑ_m is ensured by:

- a large value of the amplifier gain before the point of intervention of the disturbance,
- the presence of an integral action in the controller so as to cancel the effect of the gravitational component on the output at steady state (constant ϑ_m).

These requisites clearly suggest the use of a *proportional-integral* (PI) control action in the forward path whose transfer function is

$$C(s) = K_c \frac{1 + sT_c}{s}; \tag{8.23}$$

this yields zero error at steady state for a constant disturbance, and the presence of the real zero at $s = -1/T_c$ offers a stabilizing action. To improve dynamic performance, it is worth choosing the controller as a cascade of elementary actions with local feedback loops closed around the disturbance.

Besides closure of a position feedback loop, the most general solution is obtained by closing inner loops on velocity and acceleration. This leads to the scheme in Fig. 8.6, where $C_P(s)$, $C_V(s)$, $C_A(s)$ respectively represent *position*, *velocity*, *acceleration* controllers, and the inmost controller should

[3] Subscript i is to be dropped for notation compactness.

be of PI type as in (8.23) so as to obtain zero error at steady state for a constant disturbance. Further, k_{TP}, k_{TV}, k_{TA} are the respective transducer constants, and the amplifier gain G_v has been embedded in the gain of the inmost controller. In the scheme of Fig. 8.6, notice that ϑ_r is the reference input, which is related to the desired output ϑ_{md} as

$$\vartheta_r = k_{TP}\vartheta_{md}.$$

Further, the disturbance torque D has been suitably transformed into a voltage by the factor R_a/k_t.

In the following, a number of possible solutions that can be derived from the general scheme of Fig. 8.6 are presented; at this stage, the issue arising from possible lack of measurement of physical variables is not considered yet. Three case studies are considered which differ in the number of active feedback loops.[4]

Position feedback

In this case, the control action is characterized by

$$C_P(s) = K_P\frac{1+sT_P}{s} \qquad C_V(s) = 1 \qquad C_A(s) = 1$$

$$k_{TV} = k_{TA} = 0.$$

With these positions, the structure of the control scheme in Fig. 8.6 leads to the scheme illustrated in Fig. 5.10. From this scheme the transfer function of the forward path is

$$P(s) = \frac{k_m K_P(1+sT_P)}{s^2(1+sT_m)},$$

while that of the return path is

$$H(s) = k_{TP}.$$

A root locus analysis can be performed as a function of the gain of the position loop $k_m K_P k_{TP} T_P/T_m$. Three situations are illustrated for the poles of the closed-loop system with reference to the relation between T_P and T_m (Fig. 8.7). Stability of the closed-loop feedback system imposes some constraints on the choice of the parameters of the PI controller. If $T_P < T_m$, the system is inherently unstable (Fig. 8.7a). Then, it must be $T_P > T_m$ (Fig. 8.7b). As T_P increases, the absolute value of the real part of the two roots of the locus tending towards the asymptotes increases too, and the system has faster time response. Hence, it is convenient to render $T_P \gg T_m$ (Fig. 8.7c). In any case, the real part of the dominant poles cannot be less than $-1/2T_m$.

[4] See Appendix C for a brief brush-up on control of linear single-input/single-output systems.

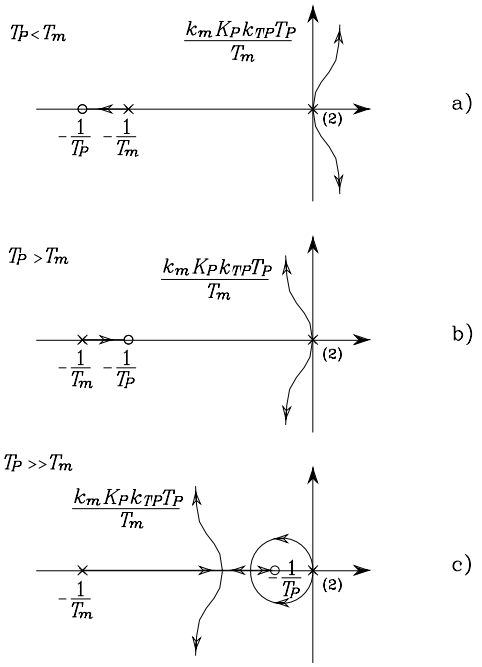

Fig. 8.7. Root loci for the position feedback control scheme

The closed-loop input/output transfer function is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{s^2(1+sT_m)}{k_m K_P k_{TP}(1+sT_P)}}, \tag{8.24}$$

which can be expressed in the form

$$W(s) = \frac{\dfrac{1}{k_{TP}}(1+sT_P)}{\left(1 + \dfrac{2\zeta s}{\omega_n} + \dfrac{s^2}{\omega_n^2}\right)(1+s\tau)},$$

where ω_n and ζ are respectively the natural frequency and damping ratio of the pair of complex poles and $-1/\tau$ locates the real pole. These values are assigned to define the joint drive dynamics as a function of the constant T_P; if $T_P > T_m$, then $1/\zeta\omega_n > T_P > \tau$ (Fig. 8.7b); if $T_P \gg T_m$ (Fig. 8.7c), for large values of the loop gain, then $\zeta\omega_n > 1/\tau \approx 1/T_P$ and the zero at $-1/T_P$ in the transfer function $W(s)$ tends to cancel the effect of the real pole.

The closed-loop disturbance/output transfer function is

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\dfrac{sR_a}{k_t K_P k_{TP}(1+sT_P)}}{1+\dfrac{s^2(1+sT_m)}{k_m K_P k_{TP}(1+sT_P)}}, \tag{8.25}$$

which shows that it is worth increasing K_P to reduce the effect of disturbance on the output during the transient. The function in (8.25) has two complex poles $(-\zeta\omega_n, \pm j\sqrt{1-\zeta^2}\omega_n)$, a real pole $(-1/\tau)$, and a zero at the origin. The zero is due to the PI controller and allows the cancellation of the effects of gravity on the angular position when ϑ_m is a constant.

In (8.25), it can be recognized that the term $K_P k_{TP}$ is the reduction factor imposed by the feedback gain on the amplitude of the output due to disturbance; hence, the quantity

$$X_R = K_P k_{TP} \tag{8.26}$$

can be interpreted as the *disturbance rejection factor*, which in turn is determined by the gain K_P. However, it is not advisable to increase K_P too much, because small damping ratios would result leading to unacceptable oscillations of the output. An estimate T_R of the *output recovery time* needed by the control system to recover the effects of the disturbance on the angular position can be evaluated by analyzing the modes of evolution of (8.25). Since $\tau \approx T_P$, such estimate is expressed by

$$T_R = \max\left\{T_P, \frac{1}{\zeta\omega_n}\right\}. \tag{8.27}$$

Position and velocity feedback

In this case, the control action is characterized by

$$C_P(s) = K_P \qquad C_V(s) = K_V\frac{1+sT_V}{s} \qquad C_A(s) = 1$$

$$k_{TA} = 0;$$

with these positions, the structure of the control scheme in Fig. 8.6 leads to scheme illustrated in Fig. 5.11. To carry out a root locus analysis as a function of the velocity feedback loop gain, it is worth reducing the velocity loop in parallel to the position loop by following the usual rules for moving blocks. From the scheme in Fig. 5.11 the transfer function of the forward path is

$$P(s) = \frac{k_m K_P K_V(1+sT_V)}{s^2(1+sT_m)},$$

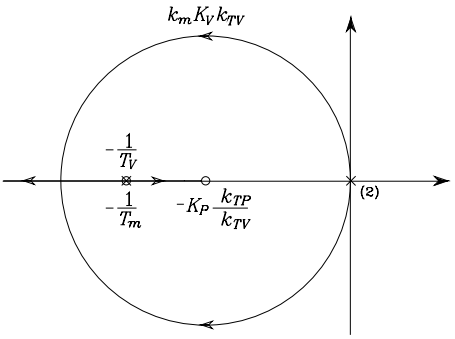

Fig. 8.8. Root locus for the position and velocity feedback control scheme

while that of the return path is

$$H(s) = k_{TP}\left(1 + s\frac{k_{TV}}{K_P k_{TP}}\right).$$

The zero of the controller at $s = -1/T_V$ can be chosen so as to cancel the effects of the real pole of the motor at $s = -1/T_m$. Then, by setting

$$T_V = T_m,$$

the poles of the closed-loop system move on the root locus as a function of the loop gain $k_m K_V k_{TV}$, as shown in Fig. 8.8. By increasing the position feedback gain K_P, it is possible to confine the closed-loop poles into a region of the complex plane with large absolute values of the real part. Then, the actual location can be established by a suitable choice of K_V.

The closed-loop input/output transfer function is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{sk_{TV}}{K_P k_{TP}} + \dfrac{s^2}{k_m K_P k_{TP} K_V}}, \tag{8.28}$$

which can be compared with the typical transfer function of a second-order system

$$W(s) = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{2\zeta s}{\omega_n} + \dfrac{s^2}{\omega_n^2}}. \tag{8.29}$$

It can be recognized that, with a suitable choice of the gains, it is possible to obtain any value of natural frequency ω_n and damping ratio ζ. Hence, if ω_n and ζ are given as design requirements, the following relations can be found:

$$K_V k_{TV} = \frac{2\zeta\omega_n}{k_m} \tag{8.30}$$

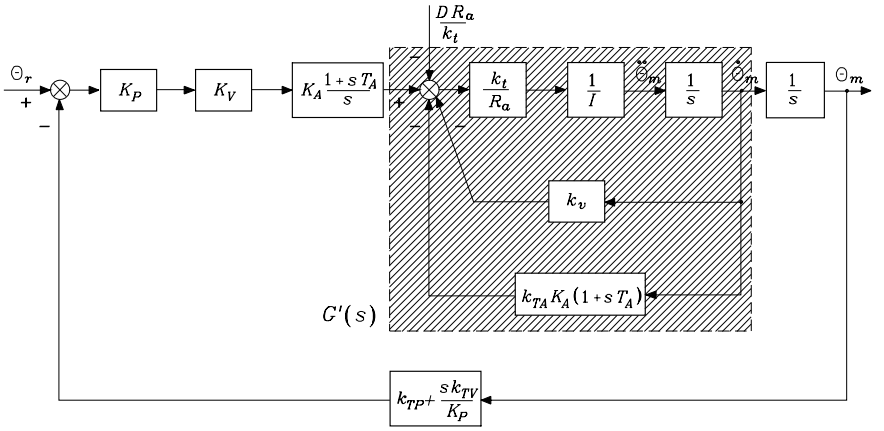

Fig. 8.9. Block scheme of position, velocity and acceleration feedback control

$$K_P k_{TP} K_V = \frac{\omega_n^2}{k_m}. \tag{8.31}$$

For given transducer constants k_{TP} and k_{TV}, once K_V has been chosen to satisfy (8.30), the value of K_P is obtained from (8.31).

The closed-loop disturbance/output transfer function is

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\dfrac{sR_a}{k_t K_P k_{TP} K_V (1+sT_m)}}{1+\dfrac{sk_{TV}}{K_P k_{TP}}+\dfrac{s^2}{k_m K_P k_{TP} K_V}}, \tag{8.32}$$

which shows that the *disturbance rejection factor* is

$$X_R = K_P k_{TP} K_V \tag{8.33}$$

and is fixed, once K_P and K_V have been chosen via (8.30), (8.31). Concerning disturbance dynamics, the presence of a zero at the origin introduced by the PI, of a real pole at $s = -1/T_m$, and of a pair of complex poles having real part $-\zeta\omega_n$ should be noticed. Hence, in this case, an estimate of the *output recovery time* is given by the time constant

$$T_R = \max\left\{T_m, \frac{1}{\zeta\omega_n}\right\}; \tag{8.34}$$

which reveals an improvement with respect to the previous case in (8.27), since $T_m \ll T_P$ and the real part of the dominant poles is not constrained by the inequality $\zeta\omega_n < 1/2T_m$.

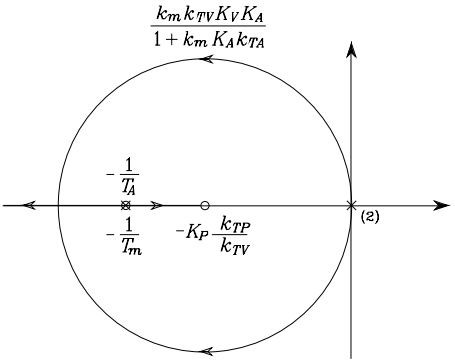

Fig. 8.10. Root locus for the position, velocity and acceleration feedback control scheme

Position, velocity and acceleration feedback

In this case, the control action is characterized by

$$C_P(s) = K_P \qquad C_V(s) = K_V \qquad C_A(s) = K_A \frac{1 + sT_A}{s}.$$

After some manipulation, the block scheme of Fig. 8.6 can be reduced to that of Fig. 8.9 where $G'(s)$ indicates the following transfer function:

$$G'(s) = \frac{k_m}{(1 + k_m K_A k_{TA}) \left(1 + \dfrac{sT_m \left(1 + k_m K_A k_{TA} \dfrac{T_A}{T_m} \right)}{(1 + k_m K_A k_{TA})} \right)}.$$

The transfer function of the forward path is

$$P(s) = \frac{K_P K_V K_A (1 + sT_A)}{s^2} G'(s),$$

while that of the return path is

$$H(s) = k_{TP} \left(1 + \frac{s k_{TV}}{K_P k_{TP}} \right).$$

Also in this case, a suitable pole cancellation is worthy which can be achieved either by setting

$$T_A = T_m,$$

or by making

$$k_m K_A k_{TA} T_A \gg T_m \qquad k_m K_A k_{TA} \gg 1.$$

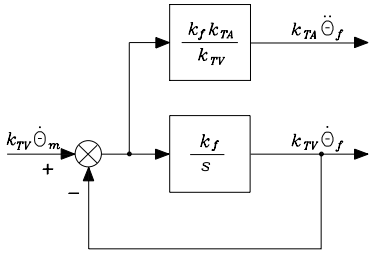

Fig. 8.11. Block scheme of a first-order filter

The two solutions are equivalent as regards dynamic performance of the control system. In both cases, the poles of the closed-loop system are constrained to move on the root locus as a function of the loop gain $k_m K_P K_V K_A/(1 + k_m K_A k_{TA})$ (Fig. 8.10). A close analogy with the previous scheme can be recognized, in that the resulting closed-loop system is again of second-order type.

The closed-loop input/output transfer function is

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{\dfrac{1}{k_{TP}}}{1 + \dfrac{sk_{TV}}{K_P k_{TP}} + \dfrac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}}, \tag{8.35}$$

while the closed-loop disturbance/output transfer function is

$$\frac{\Theta_m(s)}{D(s)} = -\frac{\dfrac{sR_a}{k_t K_P k_{TP} K_V K_A (1 + sT_A)}}{1 + \dfrac{sk_{TV}}{K_P k_{TP}} + \dfrac{s^2(1 + k_m K_A k_{TA})}{k_m K_P k_{TP} K_V K_A}}. \tag{8.36}$$

The resulting *disturbance rejection factor* is given by

$$X_R = K_P k_{TP} K_V K_A, \tag{8.37}$$

while the *output recovery time* is given by the time constant

$$T_R = \max\left\{T_A, \frac{1}{\zeta\omega_n}\right\} \tag{8.38}$$

where T_A can be made less than T_m, as pointed out above.

With reference to the transfer function in (8.29), the following relations can be established for design purposes, once ζ, ω_n, X_R have been specified:

$$\frac{2K_P k_{TP}}{k_{TV}} = \frac{\omega_n}{\zeta} \tag{8.39}$$

$$k_m K_A k_{TA} = \frac{k_m X_R}{\omega_n^2} - 1 \tag{8.40}$$

$$K_P k_{TP} K_V K_A = X_R. \tag{8.41}$$

For given k_{TP}, k_{TV}, k_{TA}, K_P is chosen to satisfy (8.39), K_A is chosen to satisfy (8.40), and then K_V is obtained from (8.41). Notice how admissible solutions for the controller typically require large values for the rejection factor X_R. Hence, in principle, not only does the acceleration feedback allow the achievement of any desired dynamic behaviour but, with respect to the previous case, it also allows the prescription of the disturbance rejection factor as long as $k_m X_R/\omega_n^2 > 1$.

In deriving the above control schemes, the issue of measurement of feedback variables was not considered explicitly. With reference to the typical position control servos that are implemented in industrial practice, there is no problem of measuring position and velocity, while a direct measurement of acceleration, in general, either is not available or is too expensive to obtain. Therefore, for the scheme of Fig. 8.9, an indirect measurement can be obtained by reconstructing acceleration from direct velocity measurement through a first-order *filter* (Fig. 8.11). The filter is characterized by a bandwidth $\omega_{3f} = k_f$. By choosing this bandwidth wide enough, the effects due to measurement lags are not appreciable, and then it is feasible to take the acceleration filter output as the quantity to feed back. Some problem may occur concerning the noise superimposed on the filtered acceleration signal, though.

Resorting to a filtering technique may be useful when only the direct position measurement is available. In this case, by means of a second-order state variable filter, it is possible to reconstruct velocity and acceleration. However, the greater lags induced by the use of a second-order filter typically degrade the performance with respect to the use of a first-order filter, because of limitations imposed on the filter bandwidth by numerical implementation of the controller and filter.

Notice that the above derivation is based on an ideal dynamic model, i.e., when the effects of transmission elasticity as well as those of amplifier and motor electrical time constants are neglected. This implies that satisfaction of design requirements imposing large values of feedback gains may not be verified in practice, since the existence of unmodelled dynamics — such as electric dynamics, elastic dynamics due to non-perfectly rigid transmissions, filter dynamics for the third scheme — might lead to degrading the system and eventually driving it to instability. In summary, the above solutions constitute design guidelines whose limits should be emphasized with regard to the specific application.

8.3.2 Decentralized Feedforward Compensation

When the joint control servos are required to track reference trajectories with high values of speed and acceleration, the tracking capabilities of the scheme in Fig. 8.6 are unavoidably degraded. The adoption of a *decentralized feedforward compensation* allows a reduction of the tracking error. Therefore, in view of the closed-loop input/output transfer functions in (8.24), (8.28), (8.35),

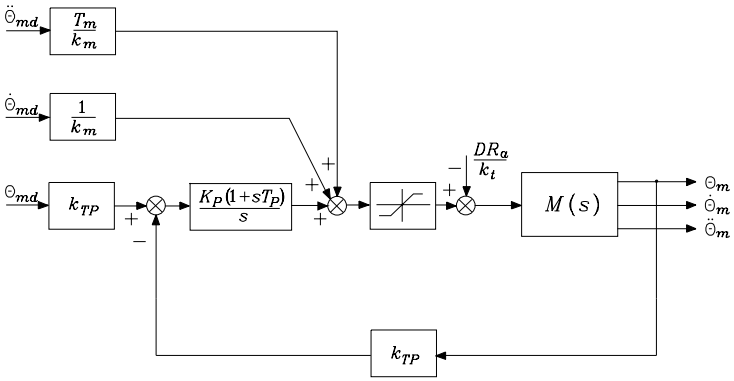

Fig. 8.12. Block scheme of position feedback control with decentralized feedforward compensation

the reference inputs to the three control structures analyzed in the previous section can be respectively modified into

$$\Theta'_r(s) = \left(k_{TP} + \frac{s^2(1+sT_m)}{k_m K_P(1+sT_P)}\right)\Theta_{md}(s) \tag{8.42}$$

$$\Theta'_r(s) = \left(k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2}{k_m K_P K_V}\right)\Theta_{md}(s) \tag{8.43}$$

$$\Theta'_r(s) = \left(k_{TP} + \frac{sk_{TV}}{K_P} + \frac{s^2(1+k_m K_A k_{TA})}{k_m K_P K_V K_A}\right)\Theta_{md}(s); \tag{8.44}$$

in this way, tracking of the desired joint position $\Theta_{md}(s)$ is achieved, if not for the effect of disturbances. Notice that computing time derivatives of the desired trajectory is not a problem, once $\vartheta_{md}(t)$ is known analytically. The tracking control schemes, resulting from simple manipulation of (8.42), (8.43), (8.44) are reported respectively in Figs. 8.12, 8.13, 8.14, where $M(s)$ indicates the motor transfer function in (5.11), with k_m and T_m as in (5.12).

All the solutions allow the input trajectory to be tracked within the range of validity and linearity of the models employed. It is worth noticing that, as the number of nested feedback loops increases, a less accurate knowledge of the system model is required to perform feedforward compensation. In fact, T_m and k_m are required for the scheme of Fig. 8.12, only k_m is required for the scheme of Fig. 8.13, and k_m again — but with reduced weight — for the scheme of Fig. 8.14.

It is worth recalling that *perfect* tracking can be obtained only under the assumption of exact matching of the controller and feedforward compensation parameters with the process parameters, as well as of exact modelling and linearity of the physical system. Deviations from the ideal values cause a performance degradation that should be analyzed case by case.

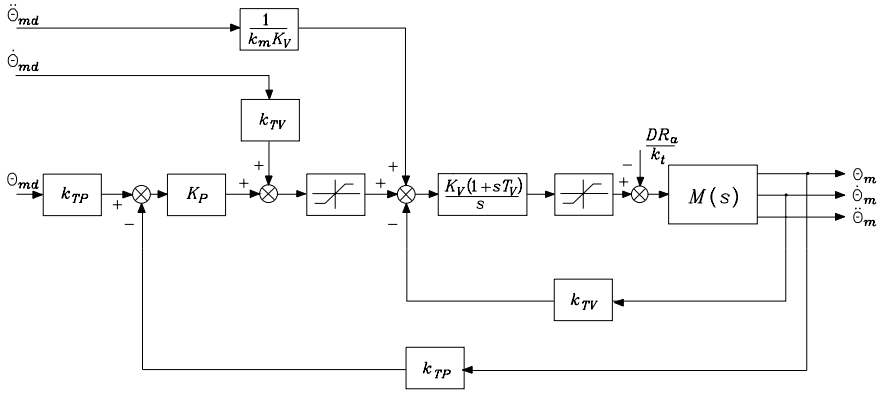

Fig. 8.13. Block scheme of position and velocity feedback control with decentralized feedforward compensation

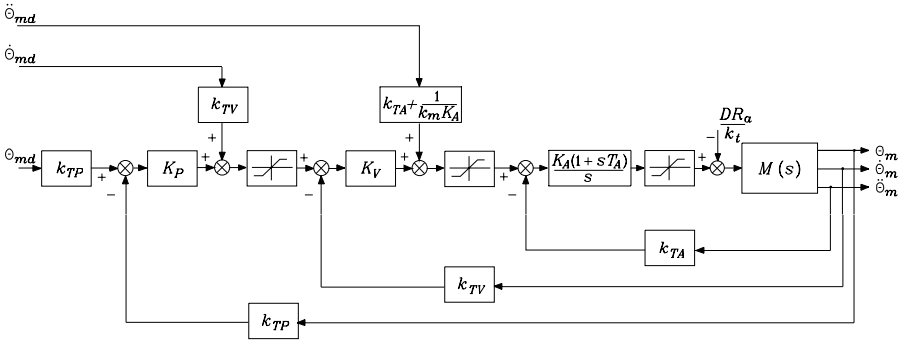

Fig. 8.14. Block scheme of position, velocity and acceleration feedback control with decentralized feedforward compensation

The presence of saturation blocks in the schemes of Figs. 8.12, 8.13, 8.14 is to be intended as intentional nonlinearities whose function is to limit relevant physical quantities during transients; the greater the number of feedback loops, the greater the number of quantities that can be limited (velocity, acceleration, and motor voltage). To this end, notice that trajectory tracking is obviously lost whenever any of the above quantities saturates. This situation often occurs for industrial manipulators required to execute point-to-point motions; in this case, there is less concern about the actual trajectories followed, and the actuators are intentionally taken to operate at the current limits so as to realize the fastest possible motions.

After simple block reduction on the above schemes, it is possible to determine equivalent control structures that utilize position feedback only and *regulators with standard actions*. It should be emphasized that the two solutions are equivalent in terms of disturbance rejection and trajectory tracking.

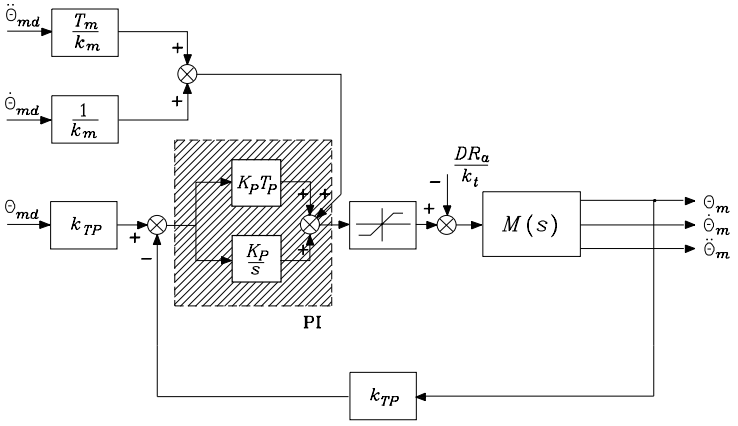

Fig. 8.15. Equivalent control scheme of PI type

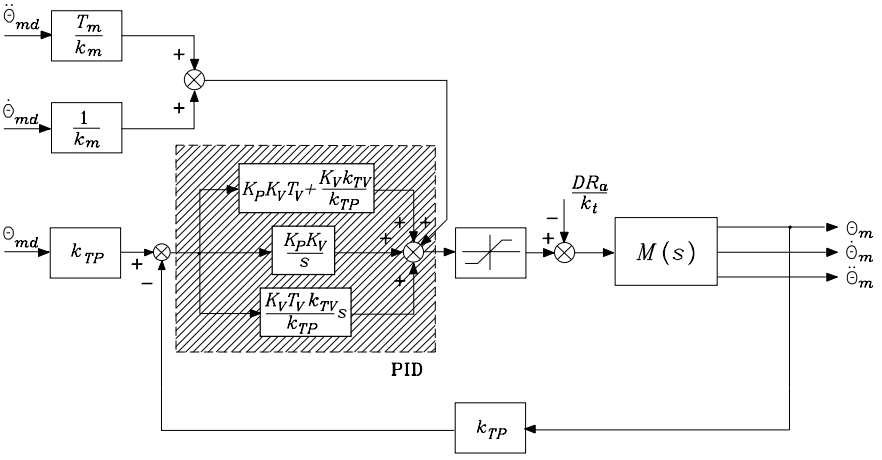

Fig. 8.16. Equivalent control scheme of PID type

However, tuning of regulator parameters is less straightforward, and the elimination of inner feedback loops prevents the possibility of setting saturations on velocity and/or acceleration. The control structures equivalent to those of Figs. 8.12, 8.13, 8.14 are illustrated in Figs. 8.15, 8.16, 8.17, respectively; control actions of PI, PID, PIDD[2] type are illustrated which are respectively equivalent to the cases of: position feedback; position and velocity feedback; position, velocity and acceleration feedback.

It is worth noticing that the equivalent control structures in Figs. 8.15–8.17 are characterized by the presence of the feedforward action $(T_m/k_m)\ddot{\vartheta}_{md} + (1/k_m)\dot{\vartheta}_{md}$. If the motor is current-controlled and not voltage-controlled, by recalling (5.13), the feedforward action is equal to $(k_i/k_t)(I_m\ddot{\vartheta}_{md} + F_m\dot{\vartheta}_{md})$. If $\dot{\vartheta}_m \approx \dot{\vartheta}_{md}$, $\ddot{\vartheta}_m \approx \ddot{\vartheta}_{md}$ and the disturbance is negligible, the term $I_m\ddot{\vartheta}_d +$

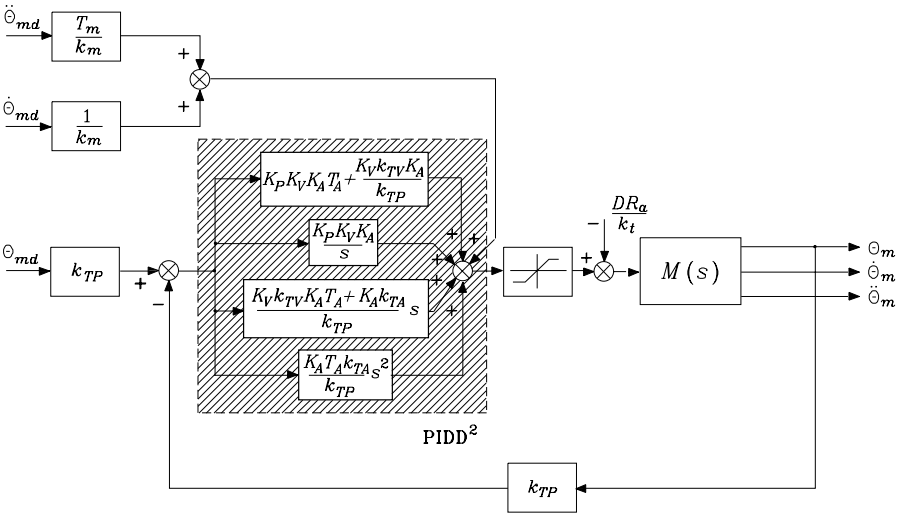

Fig. 8.17. Equivalent control scheme of PIDD^2 type

$F_m\dot{\vartheta}_d$ represents the driving torque providing the desired velocity and acceleration, as indicated by (5.3). By setting

$$i_{ad} = \frac{1}{k_t}(I_m\ddot{\vartheta}_{md} + F_m\dot{\vartheta}_{md}),$$

the feedforward action can be rewritten in the form $k_i i_{ad}$. This shows that, in the case the drive is current-controlled, it is possible to replace the acceleration and velocity feedforward actions with a current and thus a torque feedforward action, which is to be properly computed with reference to the desired motion.

This equivalence is illustrated in Fig. 8.18, where $M(s)$ has been replaced by the block scheme of an electric drive of Fig. 5.2, where the parameters of the current loop are chosen so as to realize a torque-controlled generator. The feedforward action represents a reference for the motor current, which imposes the generation of the nominal torque to execute the desired motion; the presence of the position reference allows the closure of a feedback loop which, in view of the adoption of a standard regulator with transfer function $C_R(s)$, confers robustness to the presented control structure. In summary, the performance that can be achieved with velocity and acceleration feedforward actions and voltage-controlled actuator can be achieved with a current-controlled actuator and a desired torque feedforward action.

The above schemes can incorporate the typical structure of the controllers actually implemented in the control architectures of industrial robots. In these systems it is important to choose the largest possible gains so that model inaccuracy and coupling terms do not appreciably affect positions of the single joints. As pointed out above, the upper limit on the gains is imposed by

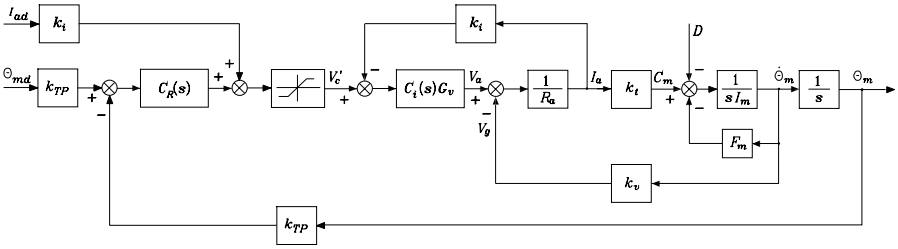

Fig. 8.18. Control scheme with current-controlled drive and current feedforward action

all those factors that have not been modelled, such as implementation of discrete-time controllers in lieu of the continuous-time controllers analyzed in theory, presence of finite sampling time, neglected dynamic effects (e.g., joint elasticity, structural resonance, finite transducer bandwidth), and sensor noise. In fact, the influence of such factors in the implementation of the above controllers may cause a severe system performance degradation for much too large values of feedback gains.

8.4 Computed Torque Feedforward Control

Define the tracking error $e(t) = \vartheta_{md}(t) - \vartheta_m(t)$. With reference to the most general scheme (Fig. 8.17), the output of the PIDD^2 regulator can be written as

$$a_2\ddot{e} + a_1\dot{e} + a_0 e + a_{-1}\int^t e(\varsigma)d\varsigma$$

which describes the time evolution of the error. The constant coefficients a_2, a_1, a_0, a_{-1} are determined by the particular solution adopted. Summing the contribution of the feedforward actions and of the disturbance to this expression yields

$$\frac{T_m}{k_m}\ddot{\vartheta}_{md} + \frac{1}{k_m}\dot{\vartheta}_{md} - \frac{R_a}{k_t}d,$$

where

$$\frac{T_m}{k_m} = \frac{I_m R_a}{k_t} \qquad k_m = \frac{1}{k_v}.$$

The input to the motor (Fig. 8.6) has then to satisfy the following equation:

$$a_2\ddot{e} + a_1\dot{e} + a_0 e + a_{-1}\int^t e(\varsigma)d\varsigma + \frac{T_m}{k_m}\ddot{\vartheta}_{md} + \frac{1}{k_m}\dot{\vartheta}_{md} - \frac{R_a}{k_t}d = \frac{T_m}{k_m}\ddot{\vartheta}_m + \frac{1}{k_m}\dot{\vartheta}_m.$$

With a suitable change of coefficients, this can be rewritten as

$$a_2'\ddot{e} + a_1'\dot{e} + a_0' e + a_{-1}'\int^t e(\varsigma)d\varsigma = \frac{R_a}{k_t}d.$$

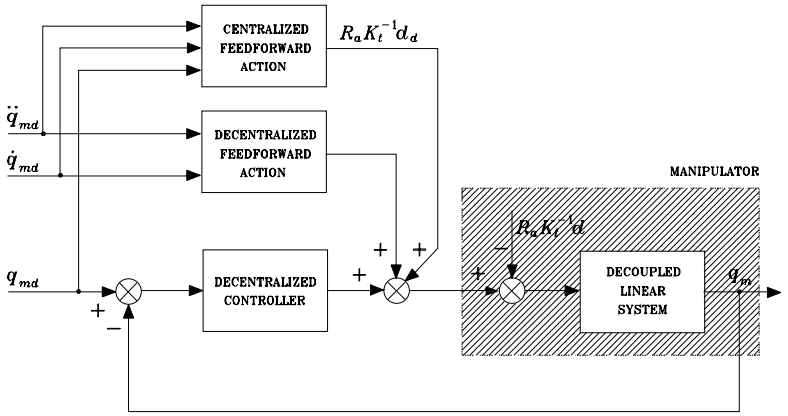

Fig. 8.19. Block scheme of computed torque feedforward control

This equation describes the *error dynamics* and shows that any physically executable trajectory is asymptotically tracked only if the disturbance term $d(t) = 0$. With the term *physically executable* it is meant that the saturation limits on the physical quantities, e.g., current and voltage in electric motors, are not violated in the execution of the desired trajectory.

The presence of the term $d(t)$ causes a tracking error whose magnitude is reduced as much as the disturbance frequency content is located off to the left of the lower limit of the bandwidth of the error system. The disturbance/error transfer function is given by

$$\frac{E(s)}{D(s)} = \frac{\dfrac{R_a}{k_t} s}{a'_2 s^3 + a'_1 s^2 + a'_0 s + a'_{-1}},$$

and thus the adoption of loop gains which are not realizable for the above discussed reasons is often required.

Nevertheless, even if the term $d(t)$ has been introduced as a disturbance, its expression is given by (8.22). It is then possible to add a further term to the previous feedforward actions which is able to compensate the disturbance itself rather than its effects. In other words, by taking advantage of model knowledge, the rejection effort of an independent joint control scheme can be lightened with notable simplification from the implementation viewpoint.

Let $\boldsymbol{q}_d(t)$ be the desired joint trajectory and $\boldsymbol{q}_{md}(t)$ the corresponding actuator trajectory as in (8.2). By adopting an *inverse model* strategy, the *feedforward* action $\boldsymbol{R}_a \boldsymbol{K}_t^{-1} \boldsymbol{d}_d$ can be introduced with

$$\boldsymbol{d}_d = \boldsymbol{K}_r^{-1} \boldsymbol{\Delta B}(\boldsymbol{q}_d) \boldsymbol{K}_r^{-1} \ddot{\boldsymbol{q}}_{md} + \boldsymbol{K}_r^{-1} \boldsymbol{C}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d) \boldsymbol{K}_r^{-1} \dot{\boldsymbol{q}}_{md} + \boldsymbol{K}_r^{-1} \boldsymbol{g}(\boldsymbol{q}_d), \quad (8.45)$$

where $\boldsymbol{R}_a$ and $\boldsymbol{K}_t$ denote the diagonal matrices of armature resistances and torque constants of the actuators. This action tends to compensate the actual

disturbance expressed by (8.22) and in turn allows the control system to operate in a better condition.

This solution is illustrated in the scheme of Fig. 8.19, which conceptually describes the control system of a manipulator with *computed torque* control. The feedback control system is representative of the n independent joint control servos; it is *decentralized*, since controller i elaborates references and measurements that refer to single Joint i. The interactions between the various joints, expressed by $\boldsymbol{d}$, are compensated by a *centralized* action whose function is to generate a feedforward action that depends on the joint references as well as on the manipulator dynamic model. This action compensates the nonlinear coupling terms due to inertial, Coriolis, centrifugal, and gravitational forces that depend on the structure and, as such, vary during manipulator motion.

Although the residual disturbance term $\widetilde{\boldsymbol{d}} = \boldsymbol{d}_d - \boldsymbol{d}$ vanishes only in the ideal case of perfect tracking ($\boldsymbol{q} = \boldsymbol{q}_d$) and exact dynamic modelling, $\widetilde{\boldsymbol{d}}$ is representative of interaction disturbances of considerably reduced magnitude with respect to $\boldsymbol{d}$. Hence, the computed torque technique has the advantage to alleviate the disturbance rejection task for the feedback control structure and in turn allows limited gains. Notice that expression (8.45) in general imposes a computationally demanding burden on the centralized part of the controller. Therefore, in those applications where the desired trajectory is generated in real time with regard to exteroceptive sensory data and commands from higher hierarchical levels of the robot control architecture,[5] on-line computation of the centralized feedforward action may require too much time.[6]

Since the actual controller is to be implemented on a computer with a finite sampling time, torque computation has to be carried out during this interval of time; in order not to degrade dynamic system performance, typical sampling times are of the order of the millisecond.

Therefore, it may be worth performing only a *partial* feedforward action so as to compensate those terms of (8.45) that give the most relevant contributions during manipulator motion. Since inertial and gravitational terms dominate velocity-dependent terms (at operational joint speeds not greater than a few radians per second), a partial compensation can be achieved by computing only the gravitational torques and the inertial torques due to the diagonal elements of the inertia matrix. In this way, only the terms depending on the global manipulator configuration are compensated while those deriving from motion interaction with the other joints are not.

Finally, it should be pointed out that, for repetitive trajectories, the above compensating contributions can be computed off-line and properly stored on the basis of a trade-off solution between memory capacity and computational requirements of the control architecture.

---

[5] See also Chap. 6.

[6] In this regard, the problem of real-time computation of compensating torques can be solved by resorting to efficient recursive formulations of manipulator inverse dynamics, such as the Newton–Euler algorithm presented in Chap. 7.

8.5 Centralized Control

In the previous sections several techniques have been discussed that allow the design of independent joint controllers. These are based on a single-input/single-output approach, since interaction and coupling effects between the joints have been considered as disturbances acting on each single joint drive system.

On the other hand, when large operational speeds are required or direct-drive actuation is employed ($\boldsymbol{K}_r = \boldsymbol{I}$), the nonlinear coupling terms strongly influence system performance. Therefore, considering the effects of the components of $\boldsymbol{d}$ as a disturbance may generate large tracking errors. In this case, it is advisable to design control algorithms that take advantage of a detailed knowledge of manipulator dynamics so as to compensate for the nonlinear coupling terms of the model. In other words, it is necessary to eliminate the causes rather than to reduce the effects induced by them; that is, to generate compensating torques for the nonlinear terms in (8.22). This leads to *centralized control* algorithms that are based on the (partial or complete) knowledge of the manipulator dynamic model.

Whenever the robot is endowed with the torque sensors at the joint motors presented in Sect. 5.4.1, those measurements can be conveniently utilized to generate the compensation action, thus avoiding the on-line computation of the terms of the dynamic model.

As shown by the dynamic model (8.1), the manipulator is not a set of n decoupled system but it is a multivariable system with n inputs (joint torques) and n outputs (joint positions) interacting between them by means of nonlinear relations.[7]

In order to follow a methodological approach which is consistent with control design, it is necessary to treat the control problem in the context of nonlinear multivariable systems. This approach will obviously account for the manipulator *dynamic model* and lead to finding *nonlinear centralized control* laws, whose implementation is needed for high manipulator dynamic performance. On the other hand, the above computed torque control can be interpreted in this framework, since it provides a model-based nonlinear control term to enhance trajectory tracking performance. Notice, however, that this action is inherently performed off line, as it is computed on the time history of the desired trajectory and not of the actual one.

In the following, the problem of the determination of the control law $\boldsymbol{u}$ ensuring a given performance to the system of manipulator with drives is tackles. Since (8.17) can be considered as a proportional relationship between $\boldsymbol{v}_c$ and $\boldsymbol{u}$, the centralized control schemes below refer directly to the generation of control toques $\boldsymbol{u}$.

[7] See Appendix C for the basic concepts on control of nonlinear mechanical systems.

8.5.1 PD Control with Gravity Compensation

Let a *constant* equilibrium posture be assigned for the system as the vector of desired joint variables $\boldsymbol{q}_d$. It is desired to find the structure of the controller which ensures global asymptotic stability of the above posture.

The determination of the control input which stabilizes the system around the equilibrium posture is based on the Lyapunov direct method.

Take the vector $[\,\widetilde{\boldsymbol{q}}^T \quad \dot{\boldsymbol{q}}^T\,]^T$ as the system state, where

$$\widetilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q} \tag{8.46}$$

represents the error between the desired and the actual posture. Choose the following positive definite quadratic form as Lyapunov function candidate:

$$V(\dot{\boldsymbol{q}}, \widetilde{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\widetilde{\boldsymbol{q}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{q}} > 0 \qquad \forall \dot{\boldsymbol{q}}, \widetilde{\boldsymbol{q}} \neq \boldsymbol{0} \tag{8.47}$$

where $\boldsymbol{K}_P$ is an $(n \times n)$ symmetric positive definite matrix. An energy-based interpretation of (8.47) reveals a first term expressing the system kinetic energy and a second term expressing the potential energy stored in the system of equivalent stiffness $\boldsymbol{K}_P$ provided by the n position feedback loops.

Differentiating (8.47) with respect to time, and recalling that $\boldsymbol{q}_d$ is constant, yields

$$\dot{V} = \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{q}}. \tag{8.48}$$

Solving (8.7) for $\boldsymbol{B}\ddot{\boldsymbol{q}}$ and substituting it in (8.48) gives

$$\dot{V} = \frac{1}{2}\dot{\boldsymbol{q}}^T \big(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\big)\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T \boldsymbol{F}\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \big(\boldsymbol{u} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{K}_P \widetilde{\boldsymbol{q}}\big). \tag{8.49}$$

The first term on the right-hand side is null since the matrix $\boldsymbol{N} = \dot{\boldsymbol{B}} - 2\boldsymbol{C}$ satisfies (7.49). The second term is negative definite. Then, the choice

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P \widetilde{\boldsymbol{q}}, \tag{8.50}$$

describing a controller with compensation of gravitational terms and a proportional action, leads to a negative semi-definite $\dot{V}$ since

$$\dot{V} = 0 \qquad \dot{\boldsymbol{q}} = \boldsymbol{0}, \forall \widetilde{\boldsymbol{q}}.$$

This result can be obtained also by taking the control law

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P \widetilde{\boldsymbol{q}} - \boldsymbol{K}_D \dot{\boldsymbol{q}}, \tag{8.51}$$

with $\boldsymbol{K}_D$ positive definite, corresponding to a *nonlinear compensation action of gravitational terms* with a *linear proportional-derivative* (PD) *action*. In fact, substituting (8.51) into (8.49) gives

$$\dot{V} = -\dot{\boldsymbol{q}}^T (\boldsymbol{F} + \boldsymbol{K}_D)\dot{\boldsymbol{q}}, \tag{8.52}$$

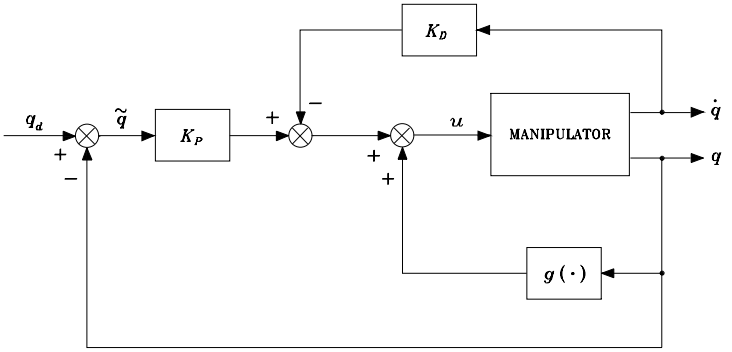

Fig. 8.20. Block scheme of joint space PD control with gravity compensation

which reveals that the introduction of the derivative term causes an increase of the absolute values of $\dot{V}$ along the system trajectories, and then it gives an improvement of system time response. Notice that the inclusion of a derivative action in the controller, as in (8.51), is crucial when direct-drive manipulators are considered. In that case, in fact, mechanical viscous damping is practically null, and current control does not allow the exploitation of the electrical viscous damping provided by voltage-controlled actuators.

According to the above, the function candidate V decreases as long as $\dot{\boldsymbol{q}} \neq \boldsymbol{0}$ for all system trajectories. It can be shown that the system reaches an *equilibrium posture*. To find such posture, notice that $\dot{V} \equiv 0$ only if $\dot{\boldsymbol{q}} \equiv \boldsymbol{0}$. System dynamics under control (8.51) is given by

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_P\tilde{\boldsymbol{q}} - \boldsymbol{K}_D\dot{\boldsymbol{q}}. \tag{8.53}$$

At the equilibrium ($\dot{\boldsymbol{q}} \equiv \boldsymbol{0}$, $\ddot{\boldsymbol{q}} \equiv \boldsymbol{0}$) it is

$$\boldsymbol{K}_P\tilde{\boldsymbol{q}} = \boldsymbol{0} \tag{8.54}$$

and then

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q} \equiv \boldsymbol{0}$$

is the sought equilibrium posture. The above derivation rigorously shows that any manipulator equilibrium posture is *globally asymptotically stable* under a controller with a PD linear action and a nonlinear gravity compensating action. Stability is ensured for any choice of $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$, as long as these are positive definite matrices. The resulting block scheme is shown in Fig. 8.20.

The control law requires the on-line computation of the term $\boldsymbol{g}(\boldsymbol{q})$. If compensation is imperfect, the above discussion does not lead to the same result; this aspect will be revisited later with reference to robustness of controllers performing nonlinear compensation.

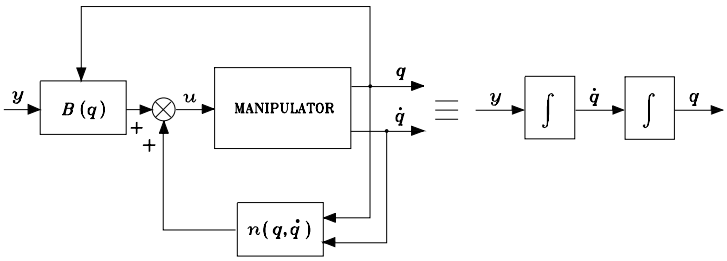

Fig. 8.21. Exact linearization performed by inverse dynamics control

8.5.2 Inverse Dynamics Control

Consider now the problem of tracking a joint space trajectory. The reference framework is that of control of nonlinear multivariable systems. The dynamic model of an n-joint manipulator is expressed by (8.7) which can be rewritten as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{u}, \tag{8.55}$$

where for simplicity it has been set

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}). \tag{8.56}$$

The approach that follows is founded on the idea to find a control vector $\boldsymbol{u}$, as a function of the system state, which is capable of realizing an input/output relationship of linear type; in other words, it is desired to perform not an approximate linearization but an *exact linearization* of system dynamics obtained by means of a *nonlinear state feedback*. The possibility of finding such a linearizing controller is guaranteed by the particular form of system dynamics. In fact, the equation in (8.55) is linear in the control $\boldsymbol{u}$ and has a full-rank matrix $\boldsymbol{B}(\boldsymbol{q})$ which can be inverted for any manipulator configuration.

Taking the control $\boldsymbol{u}$ as a function of the manipulator state in the form

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{8.57}$$

leads to the system described by

$$\ddot{\boldsymbol{q}} = \boldsymbol{y}$$

where $\boldsymbol{y}$ represents a new input vector whose expression is to be determined yet; the resulting block scheme is shown in Fig. 8.21. The nonlinear control law in (8.57) is termed *inverse dynamics control* since it is based on the computation of manipulator inverse dynamics. The system under control (8.57) is *linear* and *decoupled* with respect to the new input $\boldsymbol{y}$. In other words, the component y_i influences, with a double integrator relationship, only the joint variable q_i, independently of the motion of the other joints.

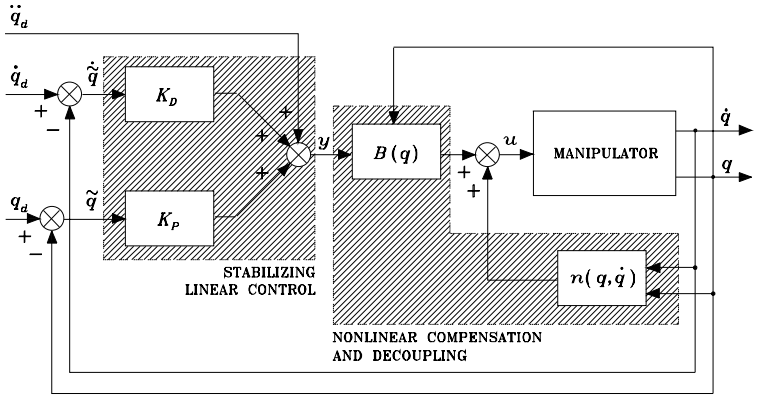

Fig. 8.22. Block scheme of joint space inverse dynamics control

In view of the choice (8.57), the manipulator control problem is reduced to that of finding a stabilizing control law $\boldsymbol{y}$. To this end, the choice

$$\boldsymbol{y} = -\boldsymbol{K}_P\boldsymbol{q} - \boldsymbol{K}_D\dot{\boldsymbol{q}} + \boldsymbol{r} \tag{8.58}$$

leads to the system of second-order equations

$$\ddot{\boldsymbol{q}} + \boldsymbol{K}_D\dot{\boldsymbol{q}} + \boldsymbol{K}_P\boldsymbol{q} = \boldsymbol{r} \tag{8.59}$$

which, under the assumption of positive definite matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$, is asymptotically stable. Choosing $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ as *diagonal* matrices of the type

$$\boldsymbol{K}_P = \text{diag}\{\omega_{n1}^2, \ldots, \omega_{nn}^2\} \qquad \boldsymbol{K}_D = \text{diag}\{2\zeta_1\omega_{n1}, \ldots, 2\zeta_n\omega_{nn}\},$$

gives a decoupled system. The reference component r_i influences only the joint variable q_i with a second-order input/output relationship characterized by a natural frequency ω_{ni} and a damping ratio ζ_i.

Given any desired trajectory $\boldsymbol{q}_d(t)$, tracking of this trajectory for the output $\boldsymbol{q}(t)$ is ensured by choosing

$$\boldsymbol{r} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\boldsymbol{q}}_d + \boldsymbol{K}_P\boldsymbol{q}_d. \tag{8.60}$$

In fact, substituting (8.60) into (8.59) gives the homogeneous second-order differential equation

$$\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_P\tilde{\boldsymbol{q}} = \boldsymbol{0} \tag{8.61}$$

expressing the dynamics of position error (8.46) while tracking the given trajectory. Such error occurs only if $\tilde{\boldsymbol{q}}(0)$ and/or $\dot{\tilde{\boldsymbol{q}}}(0)$ are different from zero and converges to zero with a speed depending on the matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ chosen.

The resulting block scheme is illustrated in Fig. 8.22, in which two feedback loops are represented; an inner loop based on the manipulator dynamic model, and an outer loop operating on the tracking error. The function of the *inner loop* is to obtain a *linear and decoupled input/output relationship*, whereas the *outer loop* is required to *stabilize the overall system*. The controller design for the outer loop is simplified since it operates on a linear and time-invariant system. Notice that the implementation of this control scheme requires computation of the inertia matrix $\boldsymbol{B}(\boldsymbol{q})$ and of the vector of Coriolis, centrifugal, gravitational, and damping terms $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ in (8.56). Unlike computed torque control, these terms must be computed *on-line* since control is now based on nonlinear feedback of the current system state, and thus it is not possible to precompute the terms off line as for the previous technique.

The above technique of nonlinear compensation and decoupling is very attractive from a control viewpoint since the nonlinear and coupled manipulator dynamics is replaced with n linear and decoupled second-order subsystems. Nonetheless, this technique is based on the assumption of perfect cancellation of dynamic terms, and then it is quite natural to raise questions about sensitivity and robustness problems due to unavoidably imperfect compensation.

Implementation of inverse dynamics control laws indeed requires that parameters of the system dynamic model are accurately known and the complete equations of motion are computed in real time. These conditions are difficult to verify in practice. On one hand, the model is usually known with a certain degree of uncertainty due to imperfect knowledge of manipulator mechanical parameters, existence of unmodelled dynamics, and model dependence on end-effector payloads not exactly known and thus not perfectly compensated. On the other hand, inverse dynamics computation is to be performed at sampling times of the order of a millisecond so as to ensure that the assumption of operating in the continuous time domain is realistic. This may pose severe constraints on the hardware/software architecture of the control system. In such cases, it may be advisable to lighten the computation of inverse dynamics and compute only the dominant terms.

On the basis of the above remarks, from an implementation viewpoint, *compensation* may be *imperfect* both for model uncertainty and for the approximations made in on-line computation of inverse dynamics. In the following, two control techniques are presented which are aimed at counteracting the effects of imperfect compensation. The first consists of the introduction of an additional term to an inverse dynamics controller which provides *robustness* to the control system by counteracting the effects of the approximations made in on-line computation of inverse dynamics. The second *adapts* the parameters of the model used for inverse dynamics computation to those of the true manipulator dynamic model.

8.5.3 Robust Control

In the case of *imperfect compensation*, it is reasonable to assume in (8.55) a control vector expressed by

$$\boldsymbol{u} = \widehat{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{y} + \widehat{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{8.62}$$

where $\widehat{\boldsymbol{B}}$ and $\widehat{\boldsymbol{n}}$ represent the adopted computational model in terms of estimates of the terms in the dynamic model. The error on the estimates, i.e., the *uncertainty*, is represented by

$$\widetilde{\boldsymbol{B}} = \widehat{\boldsymbol{B}} - \boldsymbol{B} \qquad \widetilde{\boldsymbol{n}} = \widehat{\boldsymbol{n}} - \boldsymbol{n} \tag{8.63}$$

and is due to imperfect model compensation as well as to intentional simplification in inverse dynamics computation. Notice that by setting $\widehat{\boldsymbol{B}} = \bar{\boldsymbol{B}}$ (where $\bar{\boldsymbol{B}}$ is the diagonal matrix of average inertia at the joint axes) and $\widehat{\boldsymbol{n}} = \boldsymbol{0}$, the above decentralized control scheme is recovered where the control action $\boldsymbol{y}$ can be of the general PID type computed on the error.

Using (8.62) as a nonlinear control law gives

$$\boldsymbol{B}\ddot{\boldsymbol{q}} + \boldsymbol{n} = \widehat{\boldsymbol{B}}\boldsymbol{y} + \widehat{\boldsymbol{n}} \tag{8.64}$$

where functional dependence has been omitted. Since the inertia matrix $\boldsymbol{B}$ is invertible, it is

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} + (\boldsymbol{B}^{-1}\widehat{\boldsymbol{B}} - \boldsymbol{I})\boldsymbol{y} + \boldsymbol{B}^{-1}\widetilde{\boldsymbol{n}} = \boldsymbol{y} - \boldsymbol{\eta} \tag{8.65}$$

where

$$\boldsymbol{\eta} = (\boldsymbol{I} - \boldsymbol{B}^{-1}\widehat{\boldsymbol{B}})\boldsymbol{y} - \boldsymbol{B}^{-1}\widetilde{\boldsymbol{n}}. \tag{8.66}$$

Taking as above

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{K}_P(\boldsymbol{q}_d - \boldsymbol{q}),$$

leads to

$$\ddot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{q}} = \boldsymbol{\eta}. \tag{8.67}$$

The system described by (8.67) is still nonlinear and coupled, since $\boldsymbol{\eta}$ is a nonlinear function of $\widetilde{\boldsymbol{q}}$ and $\dot{\widetilde{\boldsymbol{q}}}$; error convergence to zero is not ensured by the term on the left-hand side only.

To find control laws ensuring error convergence to zero while tracking a trajectory even in the face of uncertainties, a linear PD control is no longer sufficient. To this end, the Lyapunov direct method can be utilized again for the design of an outer feedback loop on the error which should be *robust* to the uncertainty $\boldsymbol{\eta}$.

Let the desired trajectory $\boldsymbol{q}_d(t)$ be assigned in the joint space and let $\widetilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q}$ be the position error. Its first time-derivative is $\dot{\widetilde{\boldsymbol{q}}} = \dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}$, while its second time-derivative in view of (8.65) is

$$\ddot{\widetilde{\boldsymbol{q}}} = \ddot{\boldsymbol{q}}_d - \boldsymbol{y} + \boldsymbol{\eta}. \tag{8.68}$$

By taking

$$\boldsymbol{\xi} = \begin{bmatrix} \widetilde{\boldsymbol{q}} \\ \dot{\widetilde{\boldsymbol{q}}} \end{bmatrix}, \tag{8.69}$$

as the system state, the following first-order differential matrix equation is obtained:

$$\dot{\boldsymbol{\xi}} = \boldsymbol{H}\boldsymbol{\xi} + \boldsymbol{D}(\ddot{\boldsymbol{q}}_d - \boldsymbol{y} + \boldsymbol{\eta}), \tag{8.70}$$

where $\boldsymbol{H}$ and $\boldsymbol{D}$ are block matrices of dimensions $(2n \times 2n)$ and $(2n \times n)$, respectively:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{D} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I} \end{bmatrix}. \tag{8.71}$$

Then, the problem of tracking a given trajectory can be regarded as the problem of finding a control law $\boldsymbol{y}$ which stabilizes the nonlinear time-varying error system (8.70).

Control design is based on the assumption that, even though the uncertainty $\boldsymbol{\eta}$ is unknown, an estimate on its range of variation is available. The sought control law $\boldsymbol{y}$ should guarantee asymptotic stability of (8.70) for any $\boldsymbol{\eta}$ varying in the above range. By recalling that $\boldsymbol{\eta}$ in (8.66) is a function of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}_d$, the following assumptions are made:

$$\sup_{t \geq 0} \|\ddot{\boldsymbol{q}}_d\| < Q_M < \infty \quad \forall \ddot{\boldsymbol{q}}_d \tag{8.72}$$

$$\|\boldsymbol{I} - \boldsymbol{B}^{-1}(\boldsymbol{q})\widehat{\boldsymbol{B}}(\boldsymbol{q})\| \leq \alpha \leq 1 \; \forall \boldsymbol{q} \tag{8.73}$$

$$\|\widetilde{\boldsymbol{n}}\| \leq \Phi < \infty \qquad \forall \boldsymbol{q}, \dot{\boldsymbol{q}}. \tag{8.74}$$

Assumption (8.72) is practically satisfied since any planned trajectory cannot require infinite accelerations.

Regarding assumption (8.73), since $\boldsymbol{B}$ is a positive definite matrix with upper and lower limited norms, the following inequality holds:

$$0 < B_m \leq \|\boldsymbol{B}^{-1}(\boldsymbol{q})\| \leq B_M < \infty \qquad \forall \boldsymbol{q}, \tag{8.75}$$

and then a choice for $\widehat{\boldsymbol{B}}$ always exists which satisfies (8.73). In fact, by setting

$$\widehat{\boldsymbol{B}} = \frac{2}{B_M + B_m}\boldsymbol{I},$$

from (8.73) it is

$$\|\boldsymbol{B}^{-1}\widehat{\boldsymbol{B}} - \boldsymbol{I}\| \leq \frac{B_M - B_m}{B_M + B_m} = \alpha < 1. \tag{8.76}$$

If $\widehat{\boldsymbol{B}}$ is a more accurate estimate of the inertia matrix, the inequality is satisfied with values of α that can be made arbitrarily small (in the limit, it is $\widehat{\boldsymbol{B}} = \boldsymbol{B}$ and $\alpha = 0$).

Finally, concerning assumption (8.74), observe that $\widetilde{\boldsymbol{n}}$ is a function of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. For revolute joints a periodical dependence on $\boldsymbol{q}$ is obtained, while for prismatic joints a linear dependence is obtained, but the joint ranges are limited and then the above contribution is also limited. On the other hand, regarding the dependence on $\dot{\boldsymbol{q}}$, unbounded velocities for an unstable system may arise in the limit, but in reality saturations exist on the maximum velocities of the motors. In summary, assumption (8.74) can be realistically satisfied, too.

With reference to (8.65), choose now

$$\boldsymbol{y} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{q}} + \boldsymbol{w} \tag{8.77}$$

where the PD term ensures stabilization of the error dynamic system matrix, $\ddot{\boldsymbol{q}}_d$ provides a feedforward term, and the term $\boldsymbol{w}$ is to be chosen to guarantee robustness to the effects of uncertainty described by $\boldsymbol{\eta}$ in (8.66).

Using (8.77) and setting $\boldsymbol{K} = [\,\boldsymbol{K}_P \quad \boldsymbol{K}_D\,]$ yields

$$\dot{\boldsymbol{\xi}} = \widetilde{\boldsymbol{H}}\boldsymbol{\xi} + \boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{w}), \tag{8.78}$$

where

$$\widetilde{\boldsymbol{H}} = (\boldsymbol{H} - \boldsymbol{D}\boldsymbol{K}) = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix}$$

is a matrix whose eigenvalues all have negative real parts — $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ being positive definite — which allows the desired error system dynamics to be prescribed. In fact, by choosing $\boldsymbol{K}_P = \mathrm{diag}\{\omega_{n1}^2, \ldots, \omega_{nn}^2\}$ and $\boldsymbol{K}_D = \mathrm{diag}\{2\zeta_1\omega_{n1}, \ldots, 2\zeta_n\omega_{nn}\}$, n decoupled equations are obtained as regards the linear part. If the uncertainty term vanishes, it is obviously $\boldsymbol{w} = \boldsymbol{0}$ and the above result with an exact inverse dynamics controller is recovered ($\widehat{\boldsymbol{B}} = \boldsymbol{B}$ and $\widehat{\boldsymbol{n}} = \boldsymbol{n}$).

To determine $\boldsymbol{w}$, consider the following positive definite quadratic form as Lyapunov function candidate:

$$V(\boldsymbol{\xi}) = \boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{\xi} > 0 \qquad \forall \boldsymbol{\xi} \neq \boldsymbol{0}, \tag{8.79}$$

where $\boldsymbol{Q}$ is a $(2n \times 2n)$ positive definite matrix. The derivative of V along the trajectories of the error system (8.78) is

$$\begin{aligned} \dot{V} &= \dot{\boldsymbol{\xi}}^T\boldsymbol{Q}\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{Q}\dot{\boldsymbol{\xi}} \\ &= \boldsymbol{\xi}^T(\widetilde{\boldsymbol{H}}^T\boldsymbol{Q} + \boldsymbol{Q}\widetilde{\boldsymbol{H}})\boldsymbol{\xi} + 2\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{w}). \end{aligned} \tag{8.80}$$

Since $\widetilde{\boldsymbol{H}}$ has eigenvalues with all negative real parts, it is well-known that for any symmetric positive definite matrix $\boldsymbol{P}$, the equation

$$\widetilde{\boldsymbol{H}}^T\boldsymbol{Q} + \boldsymbol{Q}\widetilde{\boldsymbol{H}} = -\boldsymbol{P} \tag{8.81}$$

gives a unique solution $\boldsymbol{Q}$ which is symmetric positive definite as well. In view of this, (8.80) becomes

$$\dot{V} = -\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + 2\boldsymbol{\xi}^T\boldsymbol{Q}\boldsymbol{D}(\boldsymbol{\eta} - \boldsymbol{w}). \tag{8.82}$$

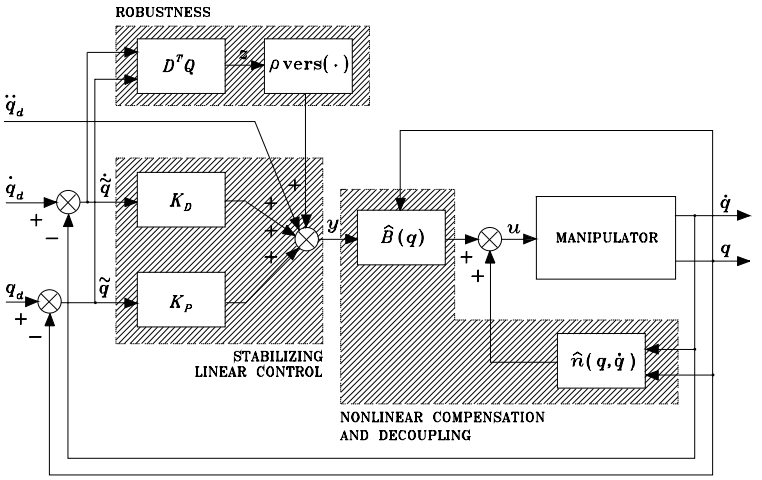

Fig. 8.23. Block scheme of joint space robust control

The first term on the right-hand side of (8.82) is negative definite and then the solutions converge if $\boldsymbol{\xi} \in \mathcal{N}(\boldsymbol{D}^T\boldsymbol{Q})$. If instead $\boldsymbol{\xi} \notin \mathcal{N}(\boldsymbol{D}^T\boldsymbol{Q})$, the control $\boldsymbol{w}$ must be chosen so as to render the second term in (8.82) less than or equal to zero. By setting $\boldsymbol{z} = \boldsymbol{D}^T\boldsymbol{Q}\boldsymbol{\xi}$, the second term in (8.82) can be rewritten as $\boldsymbol{z}^T(\boldsymbol{\eta} - \boldsymbol{w})$. Adopting the control law

$$\boldsymbol{w} = \frac{\rho}{\|\boldsymbol{z}\|}\boldsymbol{z} \qquad \rho > 0 \tag{8.83}$$

gives[8]

$$\begin{aligned} \boldsymbol{z}^T(\boldsymbol{\eta} - \boldsymbol{w}) &= \boldsymbol{z}^T\boldsymbol{\eta} - \frac{\rho}{\|\boldsymbol{z}\|}\boldsymbol{z}^T\boldsymbol{z} \\ &\leq \|\boldsymbol{z}\|\|\boldsymbol{\eta}\| - \rho\|\boldsymbol{z}\| \\ &= \|\boldsymbol{z}\|(\|\boldsymbol{\eta}\| - \rho). \end{aligned} \tag{8.84}$$

Then, if ρ is chosen so that

$$\rho \geq \|\boldsymbol{\eta}\| \qquad \forall \boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}_d, \tag{8.85}$$

the control (8.83) ensures that $\dot{V}$ is less than zero along all error system trajectories.

In order to satisfy (8.85), notice that, in view of the definition of $\boldsymbol{\eta}$ in (8.66) and of assumptions (8.72)–(8.74), and being $\|\boldsymbol{w}\| = \rho$, it is

$$\|\boldsymbol{\eta}\| \leq \|\boldsymbol{I} - \boldsymbol{B}^{-1}\widehat{\boldsymbol{B}}\|\big(\|\ddot{\boldsymbol{q}}_d\| + \|\boldsymbol{K}\|\,\|\boldsymbol{\xi}\| + \|\boldsymbol{w}\|\big) + \|\boldsymbol{B}^{-1}\|\,\|\widetilde{\boldsymbol{n}}\|$$

[8] Notice that it is necessary to divide $\boldsymbol{z}$ by the norm of $\boldsymbol{z}$ so as to obtain a linear dependence on $\boldsymbol{z}$ of the term containing the control $\boldsymbol{z}^T\boldsymbol{w}$, and thus to effectively counteract, for $\boldsymbol{z} \to \boldsymbol{0}$, the term containing the uncertainty $\boldsymbol{z}^T\boldsymbol{\eta}$ which is linear in $\boldsymbol{z}$.

$$\leq \alpha Q_M + \alpha \|\boldsymbol{K}\| \, \|\boldsymbol{\xi}\| + \alpha\rho + B_M\Phi. \tag{8.86}$$

Therefore, setting

$$\rho \geq \frac{1}{1-\alpha}(\alpha Q_M + \alpha\|\boldsymbol{K}\|\|\boldsymbol{\xi}\| + B_M\Phi) \tag{8.87}$$

gives

$$\dot{V} = -\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi} + 2\boldsymbol{z}^T\left(\boldsymbol{\eta} - \frac{\rho}{\|\boldsymbol{z}\|}\boldsymbol{z}\right) < 0 \qquad \forall \boldsymbol{\xi} \neq \boldsymbol{0}. \tag{8.88}$$

The resulting block scheme is illustrated in Fig. 8.23.

To summarize, the presented approach has lead to finding a *control* law which is formed by three different contributions:

- The term $\widehat{\boldsymbol{B}}\boldsymbol{y}+\widehat{\boldsymbol{n}}$ ensures an *approximate compensation of nonlinear effects and joint decoupling*.
- The term $\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{q}}$ introduces a *linear feedforward action* ($\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\boldsymbol{q}}_d + \boldsymbol{K}_P\boldsymbol{q}_d$) and *linear feedback action* ($-\boldsymbol{K}_D\dot{\boldsymbol{q}} - \boldsymbol{K}_P\boldsymbol{q}$) *which stabilizes the error system dynamics*.
- The term $\boldsymbol{w} = (\rho/\|\boldsymbol{z}\|)\boldsymbol{z}$ represents the *robust contribution that counteracts the indeterminacy* $\widetilde{\boldsymbol{B}}$ and $\widetilde{\boldsymbol{n}}$ in computing the nonlinear terms that depend on the manipulator state; the greater the uncertainty, the greater the positive scalar ρ. The resulting control law is of the *unit vector* type, since it is described by a vector of magnitude ρ aligned with the unit vector of $\boldsymbol{z} = \boldsymbol{D}^T\boldsymbol{Q}\boldsymbol{\xi}$, $\forall\boldsymbol{\xi}$.

All the resulting trajectories under the above robust control reach the subspace $\boldsymbol{z} = \boldsymbol{D}^T\boldsymbol{Q}\boldsymbol{\xi} = \boldsymbol{0}$ that depends on the matrix $\boldsymbol{Q}$ in the Lyapunov function V. On this *attractive* subspace, termed *sliding subspace*, the control $\boldsymbol{w}$ is ideally commuted at an infinite frequency and all error components tend to zero with a transient depending on the matrices $\boldsymbol{Q}$, $\boldsymbol{K}_P$, $\boldsymbol{K}_D$. A characterization of an error trajectory in the two-dimensional case is given in Fig. 8.24. Notice that in the case $\boldsymbol{\xi}(0) \neq \boldsymbol{0}$, with $\boldsymbol{\xi}(0) \notin \mathcal{N}(\boldsymbol{D}^T\boldsymbol{Q})$, the trajectory is attracted on the sliding hyperplane (a line) $\boldsymbol{z} = \boldsymbol{0}$ and tends towards the origin of the error state space with a time evolution governed by ρ.

In reality, the physical limits on the elements employed in the controller impose a control signal that commutes at a finite frequency, and the trajectories oscillate around the sliding subspace with a magnitude as low as the frequency is high.

Elimination of these high-frequency components (*chattering*) can be achieved by adopting a robust control law which, even if it does not guarantee error convergence to zero, ensures *bounded-norm* errors. A control law of this type is

$$\boldsymbol{w} = \begin{cases} \dfrac{\rho}{\|\boldsymbol{z}\|}\boldsymbol{z} & \text{per } \|\boldsymbol{z}\| \geq \epsilon \\ \dfrac{\rho}{\epsilon}\boldsymbol{z} & \text{per } \|\boldsymbol{z}\| < \epsilon. \end{cases} \tag{8.89}$$

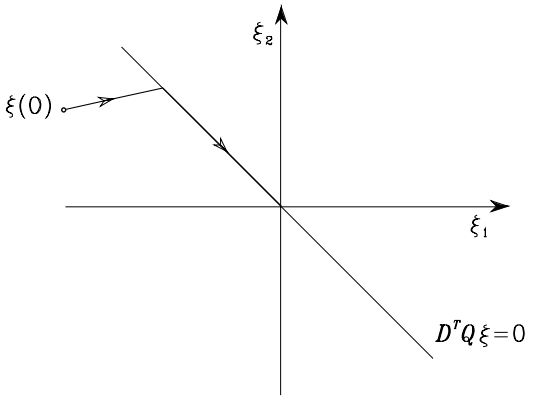

Fig. 8.24. Error trajectory with robust control

In order to provide an intuitive interpretation of this law, notice that (8.89) gives a null control input when the error is in the null space of matrix $\boldsymbol{D}^T\boldsymbol{Q}$. On the other hand, (8.83) has an equivalent gain tending to infinity when $\boldsymbol{z}$ tends to the null vector, thus generating a control input of limited magnitude. Since these inputs commute at an infinite frequency, they force the error system dynamics to stay on the sliding subspace. With reference to the above example, control law (8.89) gives rise to a hyperplane $\boldsymbol{z} = \boldsymbol{0}$ which is no longer attractive, and the error is allowed to vary within a boundary layer whose thickness depends on ϵ (Fig. 8.25).

The introduction of a contribution based on the computation of a suitable linear combination of the generalized error confers robustness to a control scheme based on nonlinear compensation. Even if the manipulator is accurately modeled, indeed, an exact nonlinear compensation may be computationally demanding, and thus it may require either a sophisticated hardware architecture or an increase of the sampling time needed to compute the control law. The solution then becomes weak from an engineering viewpoint, due either to infeasible costs of the control architecture, or to poor performance at decreased sampling rates. Therefore, considering a partial knowledge of the manipulator dynamic model with an accurate, pondered estimate of uncertainty may suggest robust control solutions of the kind presented above. It is understood that an estimate of the uncertainty should be found so as to impose control inputs which the mechanical structure can bear.

8.5.4 Adaptive Control

The computational model employed for computing inverse dynamics typically has the same structure as that of the true manipulator dynamic model, but parameter estimate uncertainty does exist. In this case, it is possible to devise solutions that allow an *on-line adaptation of the computational model to the*

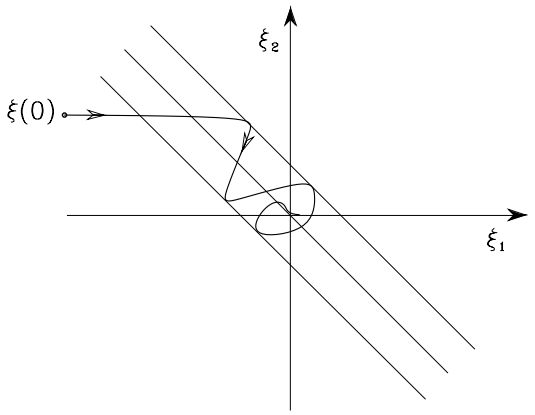

Fig. 8.25. Error trajectory with robust control and chattering elimination

dynamic model, thus performing a control scheme of the inverse dynamics type.

The possibility of finding adaptive control laws is ensured by the property of *linearity in the parameters* of the dynamic model of a manipulator. In fact, it is always possible to express the nonlinear equations of motion in a linear form with respect to a suitable set of constant dynamic parameters as in (7.81). The equation in (8.7) can then be written as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\pi} = \boldsymbol{u}, \tag{8.90}$$

where $\boldsymbol{\pi}$ is a $(p \times 1)$ vector of constant parameters and $\boldsymbol{Y}$ is an $(n \times p)$ matrix which is a function of joint positions, velocities and accelerations. This property of linearity in the dynamic parameters is fundamental for deriving adaptive control laws, among which the technique illustrated below is one of the simplest.

At first, a control scheme which can be derived through a combined computed torque/inverse dynamics approach is illustrated. The computational model is assumed to coincide with the dynamic model.

Consider the control law

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \boldsymbol{F}\dot{\boldsymbol{q}}_r + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{K}_D\boldsymbol{\sigma}, \tag{8.91}$$

with $\boldsymbol{K}_D$ a positive definite matrix. The choice

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}} \qquad \ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\dot{\tilde{\boldsymbol{q}}}, \tag{8.92}$$

with $\boldsymbol{\Lambda}$ a positive definite (usually diagonal) matrix, allows the nonlinear compensation and decoupling terms to be expressed as a function of the desired velocity and acceleration, corrected by the current state ($\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$) of the manipulator. In fact, notice that the term $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}$ weighs the contribution

that depends on velocity, not only on the basis of the desired velocity but also on the basis of the position tracking error. A similar argument also holds for the acceleration contribution, where a term depending on the velocity tracking error is considered besides the desired acceleration.

The term $\boldsymbol{K}_D\boldsymbol{\sigma}$ is equivalent to a PD action on the error if $\boldsymbol{\sigma}$ is taken as

$$\boldsymbol{\sigma} = \dot{\boldsymbol{q}}_r - \dot{\boldsymbol{q}} = \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\tilde{\boldsymbol{q}}. \tag{8.93}$$

Substituting (8.91) into (8.90) and accounting for (8.93) yields

$$\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\sigma} + \boldsymbol{F}\boldsymbol{\sigma} + \boldsymbol{K}_D\boldsymbol{\sigma} = \boldsymbol{0}. \tag{8.94}$$

Consider the Lyapunov function candidate

$$V(\boldsymbol{\sigma},\tilde{\boldsymbol{q}}) = \frac{1}{2}\boldsymbol{\sigma}^T\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\sigma} + \frac{1}{2}\tilde{\boldsymbol{q}}^T\boldsymbol{M}\tilde{\boldsymbol{q}} > 0 \qquad \forall \boldsymbol{\sigma},\tilde{\boldsymbol{q}} \neq \boldsymbol{0}, \tag{8.95}$$

where $\boldsymbol{M}$ is an $(n \times n)$ symmetric positive definite matrix; the introduction of the second term in (8.95) is necessary to obtain a Lyapunov function of the entire system state which vanishes for $\tilde{\boldsymbol{q}} = \boldsymbol{0}$ and $\dot{\tilde{\boldsymbol{q}}} = \boldsymbol{0}$. The time derivative of V along the trajectories of system (8.94) is

$$\begin{aligned}\dot{V} &= \boldsymbol{\sigma}^T\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \frac{1}{2}\boldsymbol{\sigma}^T\dot{\boldsymbol{B}}(\boldsymbol{q})\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^T\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}} \\ &= -\boldsymbol{\sigma}^T(\boldsymbol{F} + \boldsymbol{K}_D)\boldsymbol{\sigma} + \tilde{\boldsymbol{q}}^T\boldsymbol{M}\dot{\tilde{\boldsymbol{q}}},\end{aligned} \tag{8.96}$$

where the skew-symmetry propertyv of the matrix $\boldsymbol{N} = \dot{\boldsymbol{B}} - 2\boldsymbol{C}$ has been exploited. In view of the expression of $\boldsymbol{\sigma}$ in (8.93), with diagonal $\boldsymbol{\Lambda}$ and $\boldsymbol{K}_D$, it is convenient to choose $\boldsymbol{M} = 2\boldsymbol{\Lambda}\boldsymbol{K}_D$; this leads to

$$\dot{V} = -\boldsymbol{\sigma}^T\boldsymbol{F}\boldsymbol{\sigma} - \dot{\tilde{\boldsymbol{q}}}^T\boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} - \tilde{\boldsymbol{q}}^T\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\tilde{\boldsymbol{q}}. \tag{8.97}$$

This expression shows that the time derivative is negative definite since it vanishes only if $\tilde{\boldsymbol{q}} \equiv \boldsymbol{0}$ and $\dot{\tilde{\boldsymbol{q}}} \equiv \boldsymbol{0}$; thus, it follows that the origin of the state space $[\,\tilde{\boldsymbol{q}}^T \quad \boldsymbol{\sigma}^T\,]^T = \boldsymbol{0}$ is *globally asymptotically stable*. It is worth noticing that, unlike the robust control case, the error trajectory tends to the subspace $\boldsymbol{\sigma} = \boldsymbol{0}$ without the need of a high-frequency control.

On the basis of this notable result, the *control* law can be made *adaptive* with respect to the vector of parameters $\boldsymbol{\pi}$.

Suppose that the computational model has the same structure as that of the manipulator dynamic model, but its parameters are not known exactly. The control law (8.91) is then modified into

$$\begin{aligned}\boldsymbol{u} &= \widehat{\boldsymbol{B}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r + \widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r + \widehat{\boldsymbol{F}}\dot{\boldsymbol{q}}_r + \widehat{\boldsymbol{g}} + \boldsymbol{K}_D\boldsymbol{\sigma} \\ &= \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\widehat{\boldsymbol{\pi}} + \boldsymbol{K}_D\boldsymbol{\sigma},\end{aligned} \tag{8.98}$$

where $\widehat{\boldsymbol{\pi}}$ represents the available estimate on the parameters and, accordingly, $\widehat{\boldsymbol{B}}$, $\widehat{\boldsymbol{C}}$, $\widehat{\boldsymbol{F}}$, $\widehat{\boldsymbol{g}}$ denote the estimated terms in the dynamic model. Substituting control (8.98) into (8.90) gives

$$\begin{aligned}\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{\sigma}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{\sigma} + \boldsymbol{F}\boldsymbol{\sigma} + \boldsymbol{K}_D\boldsymbol{\sigma} &= -\widetilde{\boldsymbol{B}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_r - \widetilde{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_r - \widetilde{\boldsymbol{F}}\dot{\boldsymbol{q}}_r - \widetilde{\boldsymbol{g}}(\boldsymbol{q}) \\ &= -\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\widetilde{\boldsymbol{\pi}},\end{aligned} \tag{8.99}$$

where the property of linearity in the error parameter vector

$$\widetilde{\boldsymbol{\pi}} = \widehat{\boldsymbol{\pi}} - \boldsymbol{\pi} \tag{8.100}$$

has been conveniently exploited. In view of (8.63), the modelling error is characterized by

$$\widetilde{\boldsymbol{B}} = \widehat{\boldsymbol{B}} - \boldsymbol{B} \qquad \widetilde{\boldsymbol{C}} = \widehat{\boldsymbol{C}} - \boldsymbol{C} \qquad \widetilde{\boldsymbol{F}} = \widehat{\boldsymbol{F}} - \boldsymbol{F} \qquad \widetilde{\boldsymbol{g}} = \widehat{\boldsymbol{g}} - \boldsymbol{g}. \tag{8.101}$$

It is worth remarking that, in view of position (8.92), the matrix $\boldsymbol{Y}$ does not depend on the actual joint accelerations but only on their desired values; this avoids problems due to direct measurement of acceleration.

At this point, modify the Lyapunov function candidate in (8.95) into the form

$$V(\boldsymbol{\sigma},\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{\pi}}) = \frac{1}{2}\boldsymbol{\sigma}^T\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\sigma} + \widetilde{\boldsymbol{q}}^T\boldsymbol{\Lambda}\boldsymbol{K}_D\widetilde{\boldsymbol{q}} + \frac{1}{2}\widetilde{\boldsymbol{\pi}}^T\boldsymbol{K}_\pi\widetilde{\boldsymbol{\pi}} > 0 \qquad \forall \boldsymbol{\sigma},\widetilde{\boldsymbol{q}},\widetilde{\boldsymbol{\pi}} \neq \boldsymbol{0}, \tag{8.102}$$

which features an additional term accounting for the parameter error (8.100), with $\boldsymbol{K}_\pi$ symmetric positive definite. The time derivative of V along the trajectories of system (8.99) is

$$\dot{V} = -\boldsymbol{\sigma}^T\boldsymbol{F}\boldsymbol{\sigma} - \dot{\widetilde{\boldsymbol{q}}}^T\boldsymbol{K}_D\dot{\widetilde{\boldsymbol{q}}} - \widetilde{\boldsymbol{q}}^T\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\widetilde{\boldsymbol{q}} + \widetilde{\boldsymbol{\pi}}^T\big(\boldsymbol{K}_\pi\dot{\widetilde{\boldsymbol{\pi}}} - \boldsymbol{Y}^T(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma}\big). \tag{8.103}$$

If the estimate of the parameter vector is updated as in the adaptive law

$$\dot{\widehat{\boldsymbol{\pi}}} = \boldsymbol{K}_\pi^{-1}\boldsymbol{Y}^T(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\boldsymbol{\sigma}, \tag{8.104}$$

the expression in (8.103) becomes

$$\dot{V} = -\boldsymbol{\sigma}^T\boldsymbol{F}\boldsymbol{\sigma} - \dot{\widetilde{\boldsymbol{q}}}^T\boldsymbol{K}_D\dot{\widetilde{\boldsymbol{q}}} - \widetilde{\boldsymbol{q}}^T\boldsymbol{\Lambda}\boldsymbol{K}_D\boldsymbol{\Lambda}\widetilde{\boldsymbol{q}}$$

since $\dot{\widehat{\boldsymbol{\pi}}} = \dot{\widetilde{\boldsymbol{\pi}}}$ — $\boldsymbol{\pi}$ is constant.

By an argument similar to above, it is not difficult to show that the trajectories of the manipulator described by the model

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u},$$

under the control law

$$\boldsymbol{u} = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\dot{\boldsymbol{q}}_r,\ddot{\boldsymbol{q}}_r)\widehat{\boldsymbol{\pi}} + \boldsymbol{K}_D(\dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\widetilde{\boldsymbol{q}})$$

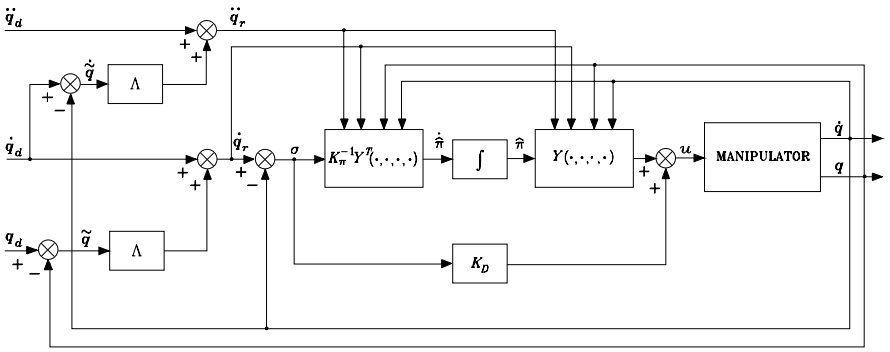

Fig. 8.26. Block scheme of joint space adaptive control

and the parameter adaptive law

$$\dot{\widehat{\boldsymbol{\pi}}} = \boldsymbol{K}_{\pi}^{-1}\boldsymbol{Y}^{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)(\dot{\widetilde{\boldsymbol{q}}} + \boldsymbol{\Lambda}\widetilde{\boldsymbol{q}}),$$

globally asymptotically converge to $\boldsymbol{\sigma} = \boldsymbol{0}$ and $\widetilde{\boldsymbol{q}} = \boldsymbol{0}$, which implies *convergence to zero of* $\widetilde{\boldsymbol{q}}$, $\dot{\widetilde{\boldsymbol{q}}}$, and *boundedness of* $\widehat{\boldsymbol{\pi}}$. The equation in (8.99) shows that asymptotically it is

$$\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)(\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}) = \boldsymbol{0}. \tag{8.105}$$

This equation does not imply that $\widehat{\boldsymbol{\pi}}$ tends to $\boldsymbol{\pi}$; indeed, convergence of parameters to their true values depends on the structure of the matrix $\boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \dot{\boldsymbol{q}}_r, \ddot{\boldsymbol{q}}_r)$ and then on the desired and actual trajectories. Nonetheless, the followed approach is aimed at solving a *direct* adaptive control problem, i.e., finding a control law that ensures limited tracking errors, and not at determining the actual parameters of the system (as in an indirect adaptive control problem). The resulting block scheme is illustrated in Fig. 8.26. To summarize, the above control law is formed by three different contributions:

- The term $\boldsymbol{Y}\widehat{\boldsymbol{\pi}}$ describes a control action of inverse dynamics type which ensures an *approximate compensation of nonlinear effects and joint decoupling*.
- The term $\boldsymbol{K}_D\boldsymbol{\sigma}$ introduces a *stabilizing linear control action of PD type on the tracking error*.
- The vector of parameter estimates $\widehat{\boldsymbol{\pi}}$ is updated by an *adaptive law of gradient type* so as to ensure asymptotic compensation of the terms in the manipulator dynamic model; the matrix $\boldsymbol{K}_{\pi}$ determines the convergence rate of parameters to their asymptotic values.

Notice that, with $\boldsymbol{\sigma} \approx \boldsymbol{0}$, the control law (8.98) is equivalent to a pure inverse dynamics compensation of the computed torque type on the basis of

desired velocities and accelerations; this is made possible by the fact that $\boldsymbol{Y}\hat{\boldsymbol{\pi}} \approx \boldsymbol{Y}\boldsymbol{\pi}$.

The control law with parameter adaptation requires the availability of a complete computational model and it does not feature any action aimed at reducing the effects of external disturbances. Therefore, a performance degradation is expected whenever unmodelled dynamic effects, e.g., when a reduced computational model is used, or external disturbances occur. In both cases, the effects induced on the output variables are attributed by the controller to parameter estimate mismatching; as a consequence, the control law attempts to counteract those effects by acting on quantities that did not provoke them originally.

On the other hand, robust control techniques provide a natural rejection to external disturbances, although they are sensitive to unmodelled dynamics; this rejection is provided by a high-frequency commuted control action that constrains the error trajectories to stay on the sliding subspace. The resulting inputs to the mechanical structure may be unacceptable. This inconvenience is in general not observed with the adoption of adaptive control techniques whose action has a naturally smooth time behaviour.

8.6 Operational Space Control

In all the above control schemes, it was always assumed that the desired trajectory is available in terms of the time sequence of the values of joint position, velocity and acceleration. Accordingly, the error for the control schemes was expressed in the joint space.

As often pointed out, motion specifications are usually assigned in the operational space, and then an inverse kinematics algorithm has to be utilized to transform operational space references into the corresponding joint space references. The process of kinematic inversion has an increasing computational load when, besides inversion of direct kinematics, inversion of first-order and second-order differential kinematics is also required to transform the desired time history of end-effector position, velocity and acceleration into the corresponding quantities at the joint level. It is for this reason that current industrial robot control systems compute the joint positions through kinematics inversion, and then perform a numerical differentiation to compute velocities and accelerations.

A different approach consists of considering control schemes developed directly in the operational space. If the motion is specified in terms of operational space variables, the measured joint space variables can be transformed into the corresponding operational space variables through direct kinematics relations. Comparing the desired input with the reconstructed variables allows the design of feedback control loops where trajectory inversion is replaced with a suitable coordinate transformation embedded in the feedback loop.

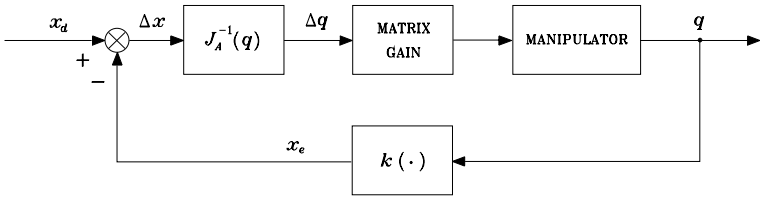

Fig. 8.27. Block scheme of Jacobian inverse control

All operational space control schemes present considerable computational requirements, in view of the necessity to perform a number of computations in the feedback loop which are somewhat representative of inverse kinematics functions. With reference to a numerical implementation, the presence of a computationally demanding load requires sampling times that may lead to degrading the performance of the overall control system.

In the face of the above limitations, it is worth presenting *operational space control* schemes, whose utilization becomes necessary when the problem of controlling interaction between the manipulator and the environment is of concern. In fact, joint space control schemes suffice only for motion control in the free space. When the manipulator's end-effector is constrained by the environment, e.g., in the case of end-effector in contact with an elastic environment, it is necessary to control both positions and contact forces and it is convenient to refer to operational space control schemes. Hence, below some solutions are presented; these are worked out for motion control, but they constitute the premise for the force/position control strategies that will be illustrated in the next chapter.

8.6.1 General Schemes

As pointed out above, operational space control schemes are based on a direct comparison of the inputs, specifying operational space trajectories, with the measurements of the corresponding manipulator outputs. It follows that the control system should incorporate some actions that allow the transformation from the operational space, in which the error is specified, to the joint space, in which control generalized forces are developed.

A possible control scheme that can be devised is the so-called *Jacobian inverse control* (Fig. 8.27). In this scheme, the end-effector pose in the operational space $\boldsymbol{x}_e$ is compared with the corresponding desired quantity $\boldsymbol{x}_d$, and then an operational space deviation $\boldsymbol{\Delta x}$ can be computed. Assumed that this deviation is sufficiently small for a good control system, $\boldsymbol{\Delta x}$ can be transformed into a corresponding joint space deviation $\boldsymbol{\Delta q}$ through the inverse of the manipulator Jacobian. Then, the control input generalized forces can be computed on the basis of this deviation through a suitable feedback matrix gain. The result is a presumable reduction of $\boldsymbol{\Delta q}$ and in turn of $\boldsymbol{\Delta x}$. In other words, the Jacobian inverse control leads to an overall system that intuitively

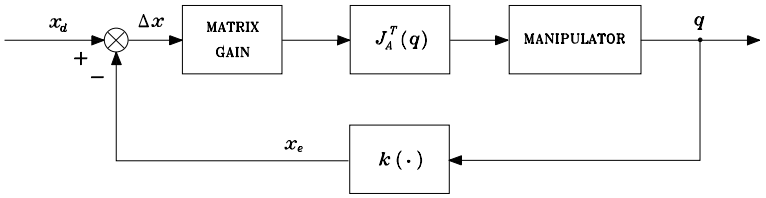

Fig. 8.28. Block scheme of Jacobian transpose control

behaves like a mechanical system with a generalized n-dimensional spring in the joint space, whose constant stiffness is determined by the feedback matrix gain. The role of such system is to take the deviation $\Delta \boldsymbol{q}$ to zero. If the matrix gain is diagonal, the generalized spring corresponds to n independent elastic elements, one for each joint.

A conceptually analogous scheme is the so-called *Jacobian transpose control* (Fig. 8.28). In this case, the operational space error is treated first through a matrix gain. The output of this block can then be considered as the elastic force generated by a generalized spring whose function in the operational space is that to reduce or to cancel the position deviation $\Delta \boldsymbol{x}$. In other words, the resulting force drives the end-effector along a direction so as to reduce $\Delta \boldsymbol{x}$. This operational space force has then to be transformed into the joint space generalized forces, through the transpose of the Jacobian, so as to realize the described behaviour.

Both Jacobian inverse and transpose control schemes have been derived in an intuitive fashion. Hence, there is no guarantee that such schemes are effective in terms of stability and trajectory tracking accuracy. These problems can be faced by presenting two mathematical solutions below, which will be shown to be substantially equivalent to the above schemes.

8.6.2 PD Control with Gravity Compensation

By analogy with joint space stability analysis, given a *constant* end-effector pose $\boldsymbol{x}_d$, it is desired to find the control structure so that the operational space error

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}_e \tag{8.106}$$

tends asymptotically to zero. Choose the following positive definite quadratic form as a Lyapunov function candidate:

$$V(\dot{\boldsymbol{q}}, \widetilde{\boldsymbol{x}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\widetilde{\boldsymbol{x}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{x}} > 0 \qquad \forall \dot{\boldsymbol{q}}, \widetilde{\boldsymbol{x}} \neq \boldsymbol{0}, \tag{8.107}$$

with $\boldsymbol{K}_P$ a symmetric positive definite matrix. Differentiating (8.107) with respect to time gives

$$\dot{V} = \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\widetilde{\boldsymbol{x}}}^T \boldsymbol{K}_P \widetilde{\boldsymbol{x}}.$$

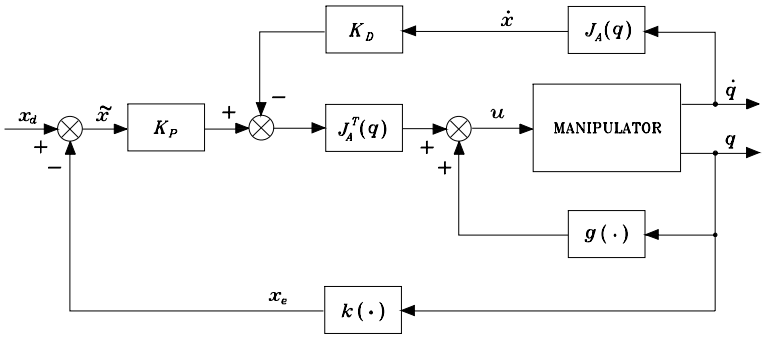

Fig. 8.29. Block scheme of operational space PD control with gravity compensation

Since $\dot{\boldsymbol{x}}_d = \boldsymbol{0}$, in view of (3.62) it is

$$\dot{\tilde{\boldsymbol{x}}} = -\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}$$

and then

$$\dot{V} = \dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}. \tag{8.108}$$

By recalling the expression of the joint space manipulator dynamic model in (8.7) and the property in (7.49), the expression in (8.108) becomes

$$\dot{V} = -\dot{\boldsymbol{q}}^T\boldsymbol{F}\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\big(\boldsymbol{u} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}}\big). \tag{8.109}$$

This equation suggests the structure of the controller; in fact, by choosing the control law

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{8.110}$$

with $\boldsymbol{K}_D$ positive definite, (8.109) becomes

$$\dot{V} = -\dot{\boldsymbol{q}}^T\boldsymbol{F}\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{8.111}$$

As can be seen from Fig. 8.29, the resulting block scheme reveals an analogy with the scheme of Fig. 8.28. Control law (8.110) performs a *nonlinear compensating action of joint space gravitational forces* and an *operational space linear PD control action*. The last term has been introduced to enhance system damping; in particular, if measurement of $\dot{\boldsymbol{x}}$ is deduced from that of $\dot{\boldsymbol{q}}$, one can simply choose the derivative term as $-\boldsymbol{K}_D\dot{\boldsymbol{q}}$.

The expression in (8.111) shows that, for any system trajectory, the Lyapunov function decreases as long as $\dot{\boldsymbol{q}} \neq \boldsymbol{0}$. The system then reaches an *equilibrium posture*. By a stability argument similar to that in the joint space (see (8.52)–(8.54)) this posture is determined by

$$\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{0}. \tag{8.112}$$

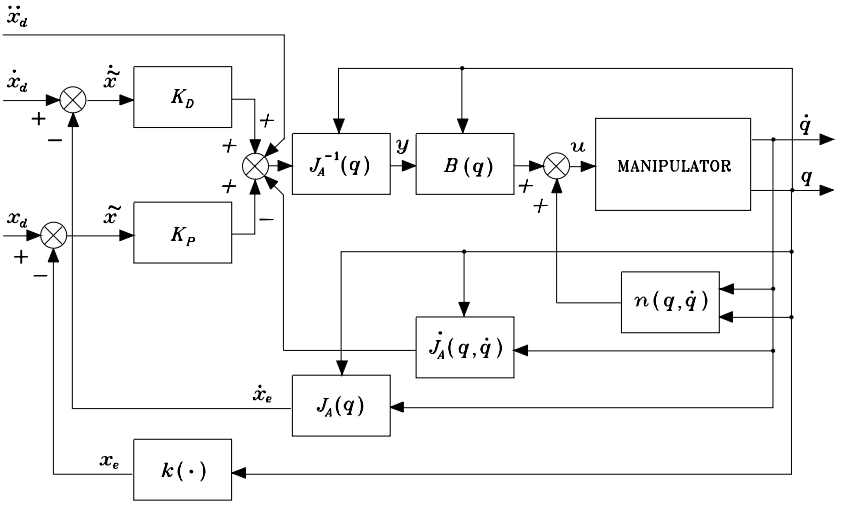

Fig. 8.30. Block scheme of operational space inverse dynamics control

From (8.112) it can be recognized that, under the assumption of *full-rank* Jacobian, it is

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}_e = \boldsymbol{0},$$

i.e., the sought result.

If measurements of $\boldsymbol{x}_e$ and $\dot{\boldsymbol{x}}_e$ are made directly in the operational space, $\boldsymbol{k}(\boldsymbol{q})$ and $\boldsymbol{J}_A(\boldsymbol{q})$ in the scheme of Fig. 8.45 are just indicative of direct kinematics functions; it is, however, necessary to measure $\boldsymbol{q}$ to update both $\boldsymbol{J}_A^T(\boldsymbol{q})$ and $\boldsymbol{g}(\boldsymbol{q})$ on-line. If measurements of operational space quantities are indirect, the controller has to compute the direct kinematics functions, too.

8.6.3 Inverse Dynamics Control

Consider now the problem of tracking an operational space trajectory. Recall the manipulator dynamic model in the form (8.55)

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{u},$$

where $\boldsymbol{n}$ is given by (8.56). As in (8.57), the choice of the *inverse dynamics linearizing control*

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

leads to the system of double integrators

$$\ddot{\boldsymbol{q}} = \boldsymbol{y}. \tag{8.113}$$

The new control input $\boldsymbol{y}$ is to be designed so as to yield tracking of a trajectory specified by $\boldsymbol{x}_d(t)$. To this end, the second-order differential equation in the form (3.98)

$$\ddot{\boldsymbol{x}}_e = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

suggests, for a nonredundant manipulator, the choice of the control law — formally analogous to (3.102) —

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\big(\ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\big) \tag{8.114}$$

with $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ positive definite (diagonal) matrices. In fact, substituting (8.114) into (8.113) gives

$$\ddot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\widetilde{\boldsymbol{x}}} + \boldsymbol{K}_P\widetilde{\boldsymbol{x}} = \boldsymbol{0} \tag{8.115}$$

which describes the operational space error dynamics, with $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ determining the error convergence rate to zero. The resulting inverse dynamics control scheme is reported in Fig. 8.30, which confirms the anticipated analogy with the scheme of Fig. 8.27. Again in this case, besides $\boldsymbol{x}_e$ and $\dot{\boldsymbol{x}}_e$, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are also to be measured. If measurements of $\boldsymbol{x}_e$ and $\dot{\boldsymbol{x}}_e$ are indirect, the controller must compute the direct kinematics functions $\boldsymbol{k}(\boldsymbol{q})$ and $\boldsymbol{J}_A(\boldsymbol{q})$ on-line.

A critical analysis of the schemes in Figs. 8.29, 8.30 reveals that the design of an operational space controller always requires computation of manipulator Jacobian. As a consequence, controlling a manipulator in the operational space is in general more complex than controlling it in the joint space. In fact, the presence of *singularities* and/or *redundancy* influences the Jacobian, and the induced effects are somewhat difficult to handle with an operational space controller. For instance, if a singularity occurs for the scheme of Fig. 8.29 and the error enters the null space of the Jacobian, the manipulator gets stuck at a different configuration from the desired one. This problem is even more critical for the scheme of Fig. 8.30 which would require the computation of a DLS inverse of the Jacobian. Yet, for a redundant manipulator, a joint space control scheme is naturally transparent to this situation, since redundancy has already been solved by inverse kinematics, whereas an operational space control scheme should incorporate a redundancy handling technique inside the feedback loop.

As a final remark, the above operational space control schemes have been derived with reference to a minimal description of orientation in terms of Euler angles. It is understood that, similar to what is presented in Sect. 3.7.3 for inverse kinematics algorithms, it is possible to adopt different definitions of orientation error, e.g., based on the angle and axis or the unit quaternion. The advantage is the use of the geometric Jacobian in lieu of the analytical Jacobian. The price to pay, however, is a more complex analysis of the stability and convergence characteristics of the closed-loop system. Even the inverse dynamics control scheme will not lead to a homogeneous error equation, and a Lyapunov argument should be invoked to ascertain its stability.

8.7 Comparison Among Various Control Schemes

In order to make a comparison between the various control schemes presented, consider the two-link planar arm with the same data of Example 7.2:

$$a_1 = a_2 = 1\,\text{m} \quad \ell_1 = \ell_2 = 0.5\,\text{m} \quad m_{\ell_1} = m_{\ell_2} = 50\,\text{kg} \quad I_{\ell_1} = I_{\ell_2} = 10\,\text{kg}\cdot\text{m}^2$$

$$k_{r1} = k_{r2} = 100 \qquad m_{m1} = m_{m2} = 5\,\text{kg} \qquad I_{m1} = I_{m2} = 0.01\,\text{kg}\cdot\text{m}^2.$$

The arm is assumed to be driven by two equal actuators with the following data:

$$F_{m1} = F_{m2} = 0.01\,\text{N}\cdot\text{m}\cdot\text{s/rad} \qquad R_{a1} = R_{a2} = 10\,\text{ohm}$$

$$k_{t1} = k_{t2} = 2\,\text{N}\cdot\text{m/A} \qquad k_{v1} = k_{v2} = 2\,\text{V}\cdot\text{s/rad};$$

it can be verified that $F_{m_i} \ll k_{vi}k_{ti}/R_{ai}$ for $i = 1, 2$.

The desired tip trajectories have a typical trapezoidal velocity profile, and thus it is anticipated that sharp torque variations will be induced. The tip path is a motion of 1.6 m along the horizontal axis, as in the path of Example 7.2. In the first case (*fast* trajectory), the acceleration time is 0.6 s and the maximum velocity is 1 m/s. In the second case (*slow* trajectory), the acceleration time is 0.6 s and the maximum velocity is 0.25 m/s. The motion of the controlled arm was simulated on a computer, by adopting a discrete-time implementation of the controller with a sampling time of 1 ms.

The following control schemes in the joint space and in the operational space have been utilized; an (analytical) inverse kinematics solution has been implemented to generate the reference inputs to the joint space control schemes:

A. Independent joint control with position and velocity feedback (Fig. 5.11) with the following data for each joint servo:

$$K_P = 5 \qquad K_V = 10 \qquad k_{TP} = k_{TV} = 1,$$

corresponding to $\omega_n = 5\,\text{rad/s}$ and $\zeta = 0.5$.

B. Independent joint control with position, velocity and acceleration feedback (Fig. 8.9) with the following data for each joint servo:

$$K_P = 5 \qquad K_V = 10 \qquad K_A = 2 \qquad k_{TP} = k_{TV} = k_{TA} = 1,$$

corresponding to $\omega_n = 5\,\text{rad/s}$, $\zeta = 0.5$, $X_R = 100$. To reconstruct acceleration, a first-order filter has been utilized (Fig. 8.11) characterized by $\omega_{3f} = 100\,\text{rad/s}$.

C. As in scheme **A** with the addition of a decentralized feedforward action (Fig. 8.13).

D. As in scheme **B** with the addition of a decentralized feedforward action (Fig. 8.14).

E. Joint space computed torque control (Fig. 8.19) with feedforward compensation of the diagonal terms of the inertia matrix and of gravitational terms, and decentralized feedback controllers as in scheme **A**.

F. Joint space PD control with gravity compensation (Fig. 8.20), modified by the addition of a feedforward velocity term $\boldsymbol{K}_D\dot{\boldsymbol{q}}_d$, with the following data:

$$\boldsymbol{K}_P = 3750\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 750\boldsymbol{I}_2.$$

G. Joint space inverse dynamics control (Fig. 8.22) with the following data:

$$\boldsymbol{K}_P = 25\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 5\boldsymbol{I}_2.$$

H. Joint space robust control (Fig. 8.23), under the assumption of constant inertia ($\widehat{\boldsymbol{B}} = \bar{\boldsymbol{B}}$) and compensation of friction and gravity ($\widehat{\boldsymbol{n}} = \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{g}$), with the following data:

$$\boldsymbol{K}_P = 25\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 5\boldsymbol{I}_2 \qquad \boldsymbol{P} = \boldsymbol{I}_2 \qquad \rho = 70 \qquad \epsilon = 0.004.$$

I. As in case **H** with $\epsilon = 0.01$.

J. Joint space adaptive control (Fig. 8.26) with a parameterization of the arm dynamic model (7.82) as in (7.83), (7.84). The initial estimate of the vector $\widehat{\boldsymbol{\pi}}$ is computed on the basis of the nominal parameters. The arm is supposed to carry a load which causes the following variations on the second link parameters:

$$\Delta m_2 = 10\,\mathrm{kg} \qquad \Delta m_2\ell_{C2} = 11\,\mathrm{kg{\cdot}m} \qquad \Delta\widehat{I}_2 = 12.12\,\mathrm{kg{\cdot}m^2}.$$

This information is obviously utilized only to update the simulated arm model. Further, the following data are set:

$$\boldsymbol{\Lambda} = 5\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 750\boldsymbol{I}_2 \qquad \boldsymbol{K}_\pi = 0.01\boldsymbol{I}_8.$$

K. Operational space PD control with gravity compensation (Fig. 8.29), modified by the addition of a feedforward velocity term $\boldsymbol{K}_D\dot{\boldsymbol{x}}_d$, with the following data:

$$\boldsymbol{K}_P = 16250\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 3250\boldsymbol{I}_2.$$

L. Operational space inverse dynamics control (Fig. 8.30) with the following data:

$$\boldsymbol{K}_P = 25\boldsymbol{I}_2 \qquad \boldsymbol{K}_D = 5\boldsymbol{I}_2.$$

It is worth remarking that the adopted model of the dynamic system of arm with drives is that described by (8.7). In the decentralized control schemes **A**–**E**, the joints have been voltage-controlled as in the block scheme of Fig. 8.3, with unit amplifier gains ($\boldsymbol{G}_v = \boldsymbol{I}$). On the other hand, in the centralized control schemes **F**–**L**, the joints have been current-controlled as in the block scheme of Fig. 8.4, with unit amplifier gains ($\boldsymbol{G}_i = \boldsymbol{I}$).

Regarding the parameters of the various controllers, these have been chosen in such a way as to allow a significant comparison of the performance of each scheme in response to congruent control actions. In particular, it can be observed that:

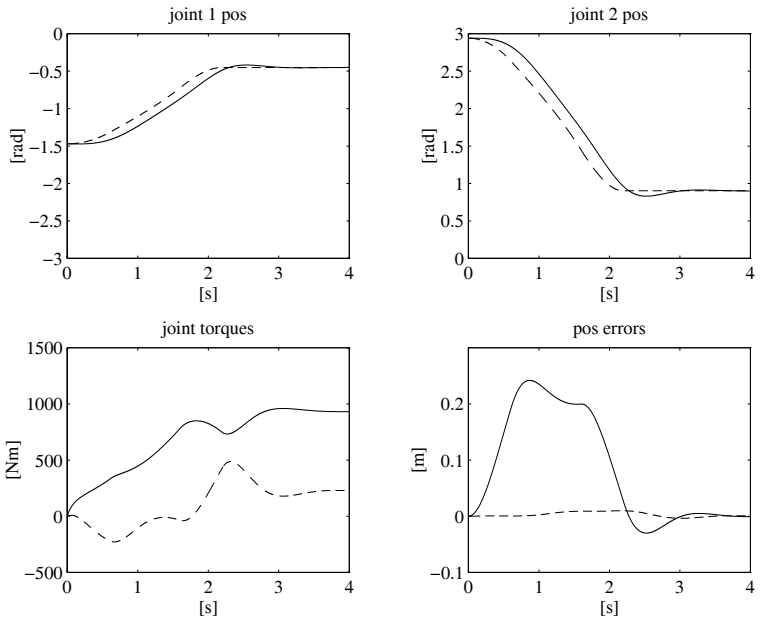

Fig. 8.31. Time history of the joint positions and torques and of the tip position errors for the *fast* trajectory with control scheme **A**

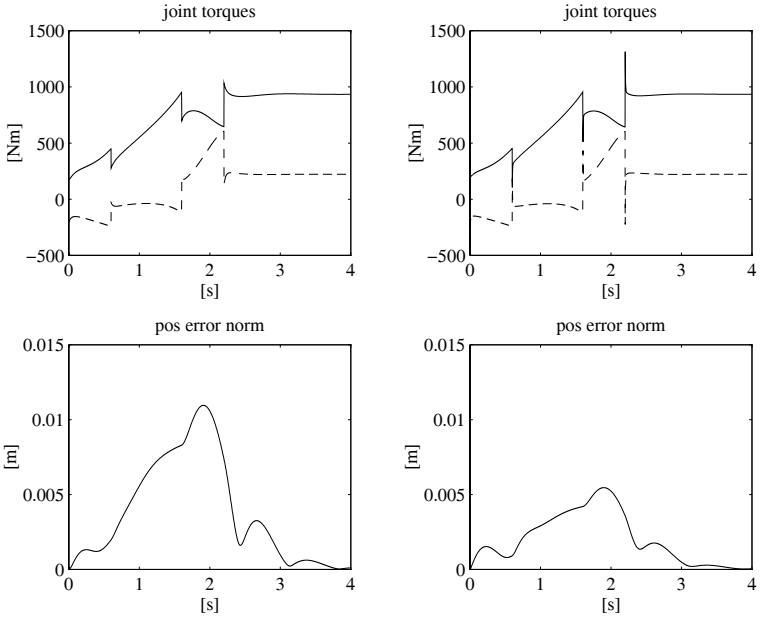

Fig. 8.32. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory; *left*: with control scheme **C**, *right*: with control scheme **D**

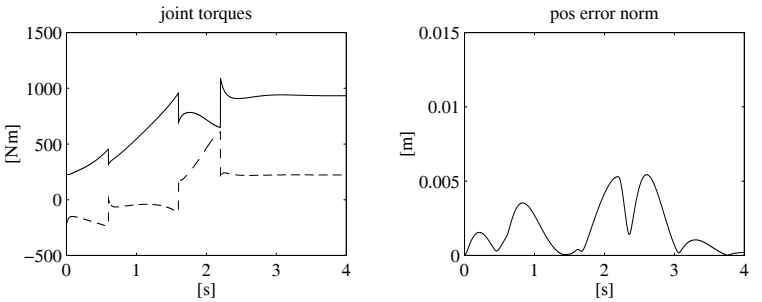

Fig. 8.33. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **E**

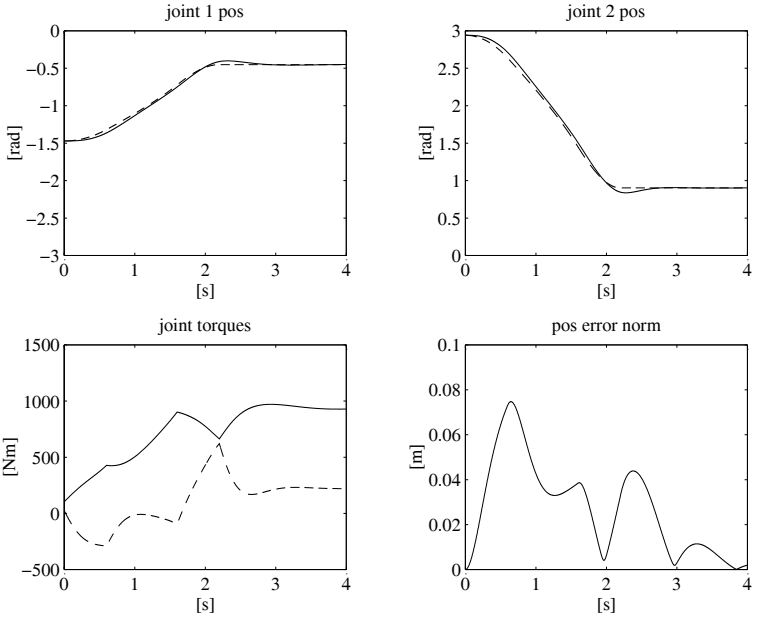

Fig. 8.34. Time history of the joint positions and torques and of the norm of tip position error for the *fast* trajectory with control scheme **F**

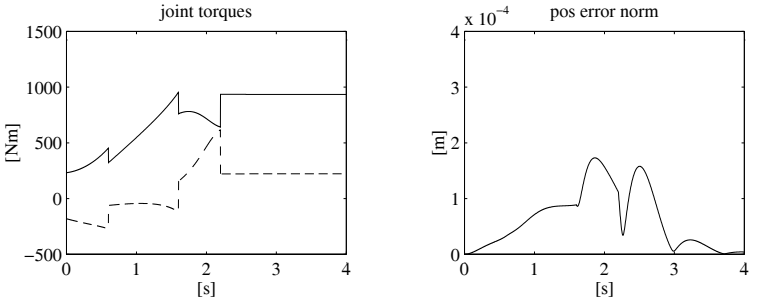

Fig. 8.35. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **G**

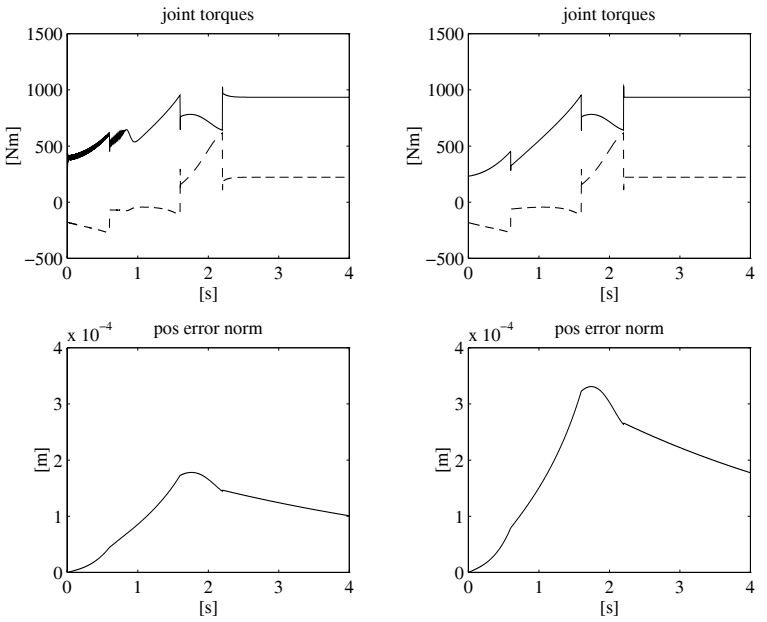

Fig. 8.36. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory; *left*: with control scheme **H**, *right*: with control scheme **I**

- The dynamic behaviour of the joints is the same for schemes **A**–**E**.
- The gains of the PD actions in schemes **G**, **H**, **I** and **L** have been chosen so as to obtain the same natural frequency and damping ratios as those of schemes **A**–**E**.

The results obtained with the various control schemes are illustrated in Figs. 8.31–8.39 for the *fast* trajectory and in Figs. 8.40–8.48 for the *slow* trajectory, respectively. In the case of two quantities represented in the same plot notice that:

- For the joint trajectories, the dashed line indicates the reference trajectory obtained from the tip trajectory via inverse kinematics, while the solid line indicates the actual trajectory followed by the arm.
- For the joint torques, the solid line refers to Joint 1 while the dashed line refers to Joint 2.
- For the tip position error, the solid line indicates the error component along the horizontal axis while the dashed line indicates the error component along the vertical axis.

Finally, the representation scales have been made as uniform as possible in order to allow a more direct comparison of the results.

Regarding performance of the various control schemes for the *fast* trajectory, the obtained results lead to the following considerations.

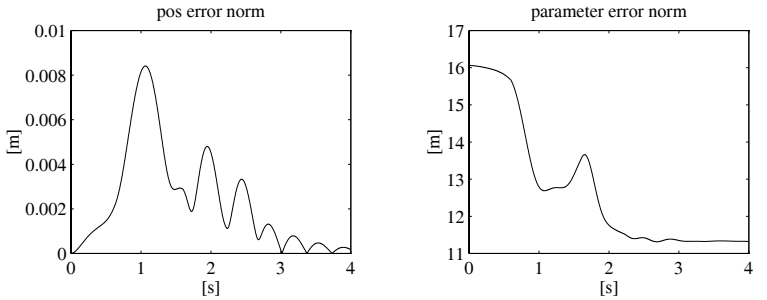

Fig. 8.37. Time history of the norm of tip position error and of the norm of parameter error vector for the *fast* trajectory with control scheme **J**

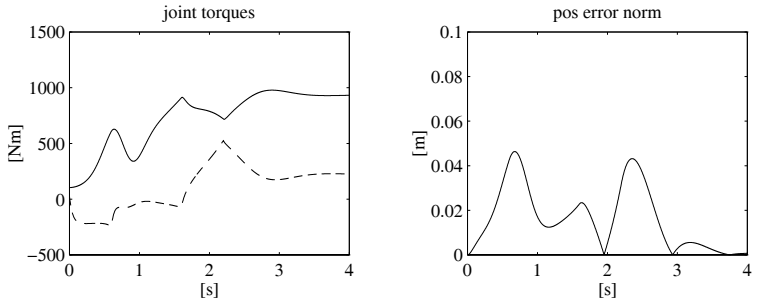

Fig. 8.38. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **K**

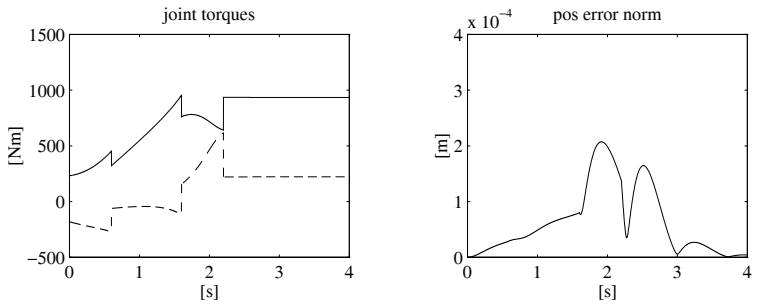

Fig. 8.39. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **L**

Deviation of the actual joint trajectories from the desired ones shows that tracking performance of scheme **A** is quite poor (Fig. 8.31). It should be noticed, however, that the largest contribution to the error is caused by a time lag of the actual trajectory behind the desired one, while the distance

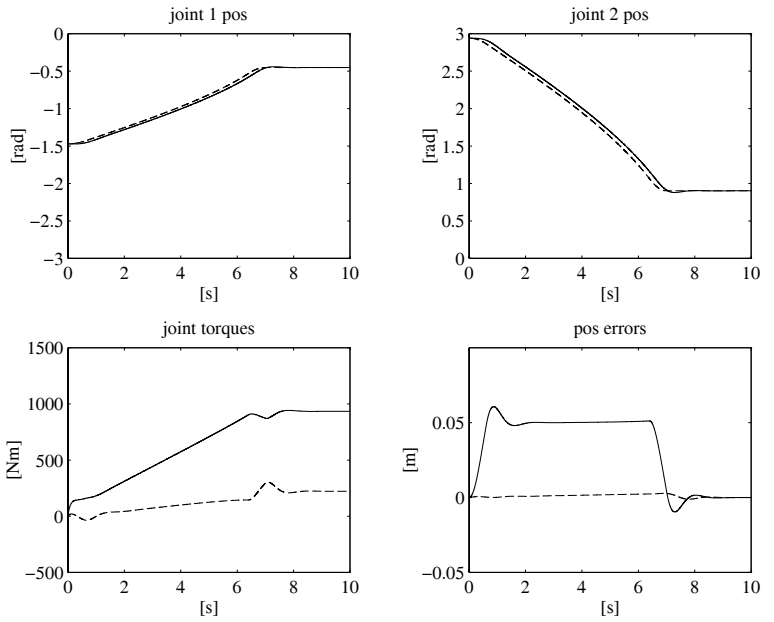

Fig. 8.40. Time history of the joint positions and torques and of the tip position errors for the *slow* trajectory with control scheme **A**

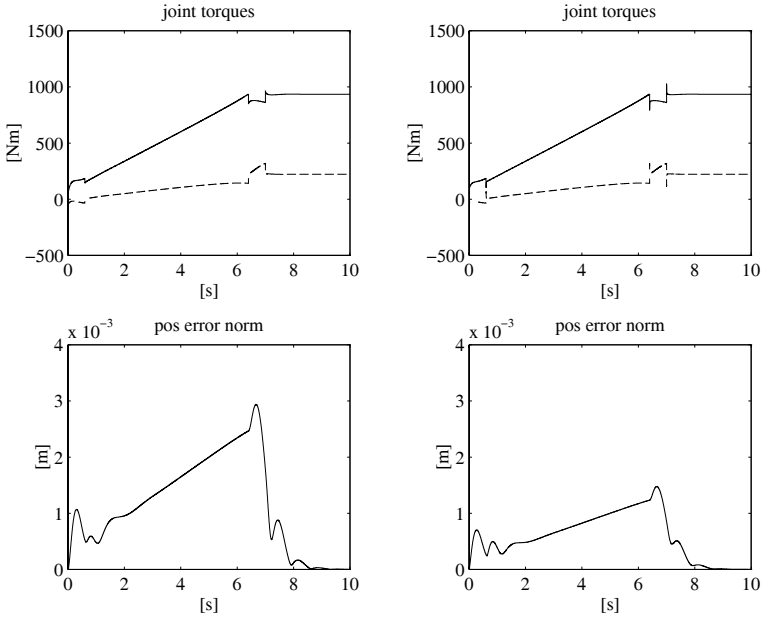

Fig. 8.41. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory; *left*: with control scheme **C**, *right*: with control scheme **D**

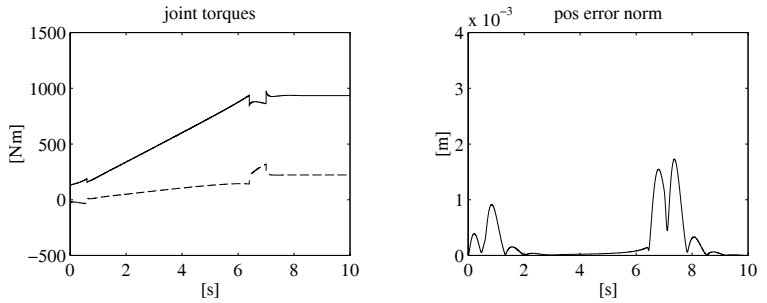

Fig. 8.42. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **E**

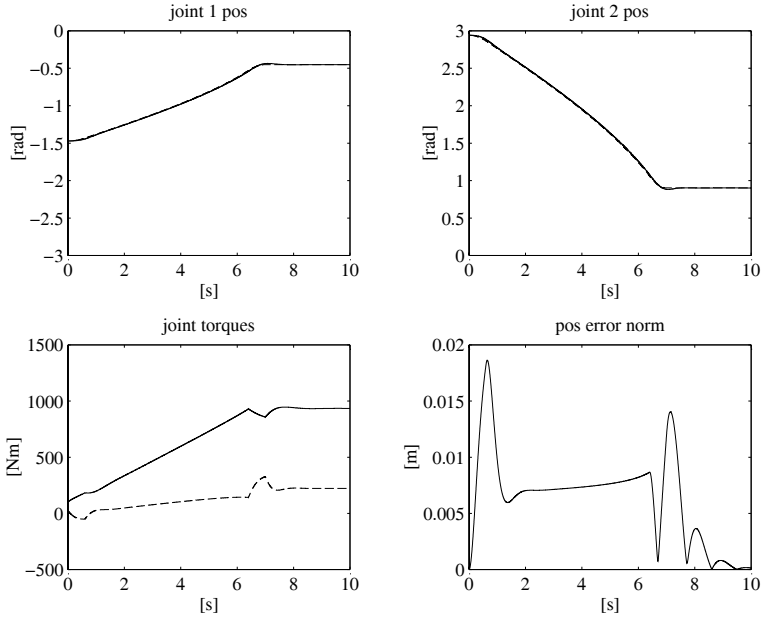

Fig. 8.43. Time history of the joint positions and torques and of the norm of tip position error for the *slow* trajectory with control scheme **F**

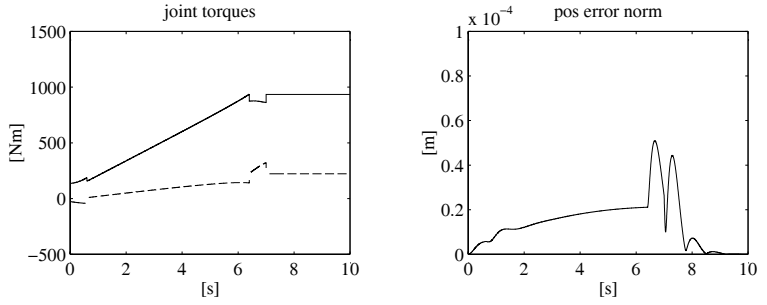

Fig. 8.44. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **G**

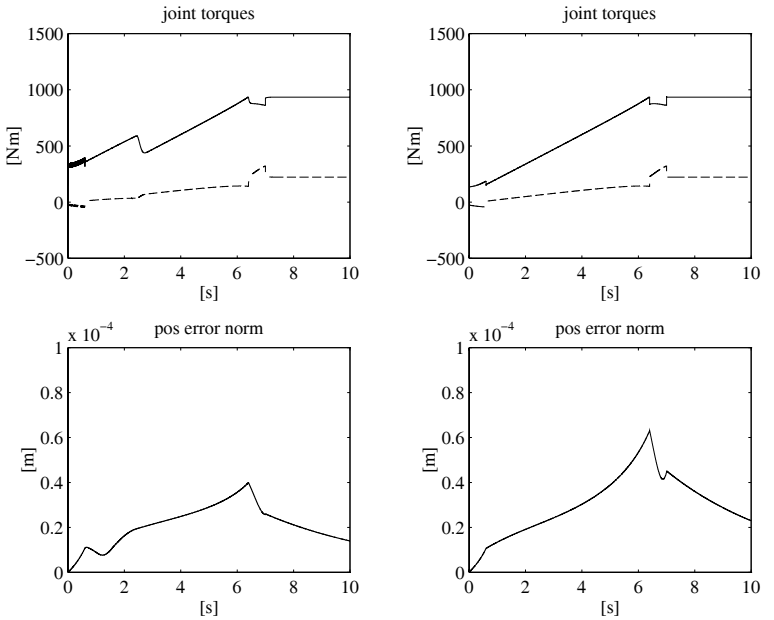

Fig. 8.45. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory; *left*: with control scheme **H**, *right*: with control scheme **I**

of the tip from the geometric path is quite contained. Similar results were obtained with scheme **B**, and then they have not been reported.

With schemes **C** and **D**, an appreciable tracking accuracy improvement is observed (Fig. 8.32), with better performance for the second scheme, thanks to the outer acceleration feedback loop that allows a disturbance rejection factor twice as much as for the first scheme. Notice that the feedforward action yields a set of torques which are closer to the nominal ones required to execute the desired trajectory; the torque time history has a discontinuity in correspondence of the acceleration and deceleration fronts.

The tracking error is further decreased with scheme **E** (Fig. 8.33), by virtue of the additional nonlinear feedforward compensation.

Scheme **F** guarantees stable convergence to the final arm posture with a tracking performance which is better than that of schemes **A** and **B**, thanks to the presence of a velocity feedforward action, but worse than that of schemes **C**–**E**, in view of lack of an acceleration feedforward action (Fig. 8.34).

As would be logical to expect, the best results are observed with scheme **G** for which the tracking error is practically zero, and it is mainly due to numerical discretization of the controller (Fig. 8.35).

It is then worth comparing the performance of schemes **H** and **I** (Fig. 8.36). In fact, the choice of a small threshold value for ϵ (scheme **H**) induces high-

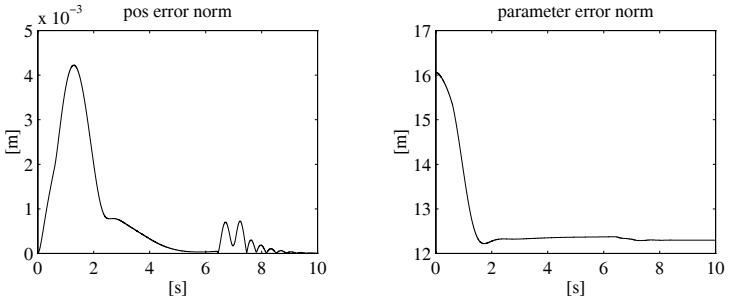

Fig. 8.46. Time history of the norm of tip position error and of the norm of parameter error vector for the *slow* trajectory with control scheme **J**

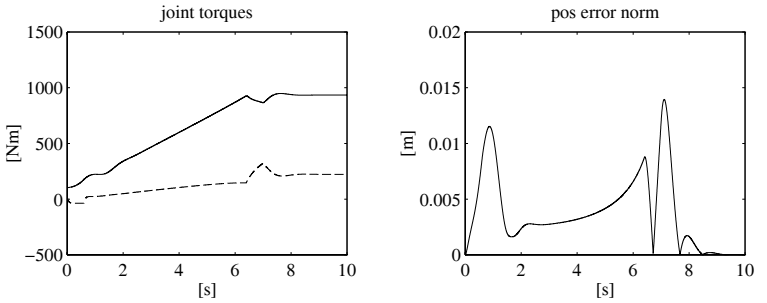

Fig. 8.47. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **K**

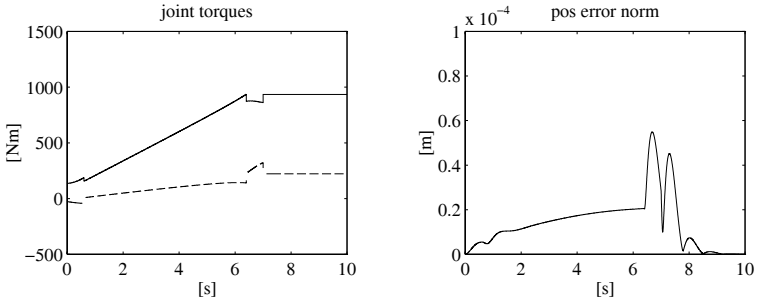

Fig. 8.48. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **L**

frequency components in Joint 1 torque (see the thick portions of the torque plot) at the advantage of a very limited tracking error. As the threshold value is increased (scheme **I**), the torque assumes a smoother behaviour at the expense of a doubled norm of tracking error, though.

For scheme **J**, a lower tracking error than that of scheme **F** is observed, thanks to the effectiveness of the adaptive action on the parameters of the dynamic model. Nonetheless, the parameters do not converge to their nominal values, as confirmed by the time history of the norm of the parameter error vector that reaches a non-null steady-state value (Fig. 8.37).

Finally, the performance of schemes **K** and **L** is substantially comparable to that of corresponding schemes **F** and **G** (Figs. 8.38 and 8.39).

Performance of the various control schemes for the *slow* trajectory is globally better than that for the *fast* trajectory. Such improvement is particularly evident for the decentralized control schemes (Figs. 8.40–8.42), whereas the tracking error reduction for the centralized control schemes is less dramatic (Figs. 8.43–8.48), in view of the small order of magnitude of the errors already obtained for the *fast* trajectory. In any case, as regards performance of each single scheme, it is possible to make a number of remarks analogous to those previously made.

Bibliography

The independent joint control is analyzed in classical texts [180, 120, 200], and scientific articles [19, 127, 141, 101, 39]. Stability of PD control with gravity compensation is proved in [7], on the basis of the notable properties of the dynamic model in [226].

Computed torque control and inverse dynamics control were developed at the beginning of the 1970s. One of the first experimental works is [149]. Other articles on the topic are [83, 4, 117, 121, 126, 227, 29].

The main approaches of robust control are inspired to the work [50]. Among them it is worth citing [212, 84, 130, 219, 205, 216]. Robust controllers based on the high gain concept are presented in [192, 222]. A survey on robust control is [1].

One of the first approaches to adaptive control, based on the assumption of decoupled joint dynamics, is presented in [67]. The first works on adaptive control accounting for the manipultor nonlinear dynamics are [15, 167, 100], yet they exploit the notable properties of the dynamic model only to some extent. The adaptive version of inverse dynamics control is analyzed in [52, 157]. The approach based on the energy properties of the dynamic model has been proposed in [214] and further analyzed in [218]. An interesting tutorial on adaptive control is [175].

Operational space control has been proposed in [114], on the basis of the resolved acceleration control concept [143]. Inverse dynamics control schemes in the operational space are given in [30]. For the extension to redundant manipulators see [102].

Problems

8.1. With reference to the block scheme with position feedback in Fig. 5.10, find the transfer functions of the forward path, the return path, and the closed-loop system.

8.2. With reference to the block scheme with position and velocity feedback in Fig. 5.11, find the transfer functions of the forward path, the return path, and the closed-loop system.

8.3. With reference to the block scheme with position, velocity and acceleration feedback in Fig. 8.9, find the transfer functions of the forward path, the return path, and the closed-loop system.

8.4. For a single joint drive system with the data: $I = 6\,\text{kg}\cdot\text{m}^2$, $R_a = 0.3\,\text{ohm}$, $k_t = 0.5\,\text{N}\cdot\text{m/A}$, $k_v = 0.5\,\text{V}\cdot\text{s/rad}$, $F_m = 0.001\,\text{N}\cdot\text{m}\cdot\text{s/rad}$, find the parameters of the controller with position feedback (unit transducer constant) that yield a closed-loop response with damping ratio $\zeta \geq 0.4$. Discuss disturbance rejection properties.

8.5. For the drive system of Problem 8.4, find the parameters of the controller with position and velocity feedback (unit transducer constants) that yield a closed-loop response with damping ratio $\zeta \geq 0.4$ and natural frequency $\omega_n = 20\,\text{rad/s}$. Discuss disturbance rejection properties.

8.6. For the drive system of Problem 8.4, find the parameters of the controller with position, velocity and acceleration feedback (unit transducer constants) that yield a closed-loop response with damping ratio $\zeta \geq 0.4$, natural frequency $\omega_n = 20\,\text{rad/s}$ and disturbance rejection factor $X_R = 400$. Also, design a first-order filter that allows acceleration measurement reconstruction.

8.7. Verify that the control schemes in Figs. 8.12, 8.13, 8.14 correspond to realizing (8.42), (8.43), (8.44), respectively.

8.8. Verify that the standard regulation schemes in Figs. 8.15, 8.16, 8.17 are equivalent to the schemes in Figs. 8.12, 8.13, 8.14, respectively.

8.9. Prove inequality (8.76).

8.10. For the two-link planar arm with the same data as in Sect. 8.7, design a joint control of PD type with gravity compensation. By means of a computer simulation, verify stability for the following postures $\boldsymbol{q} = [\,\pi/4 \quad -\pi/2\,]^T$ and $\boldsymbol{q} = [\,-\pi \quad -3\pi/4\,]^T$, respectively. Implement the control in discrete-time with a sampling time of 1 ms.

8.11. For the two-link planar arm with the same data as in Sect. 8.7, under the assumption of a concentrated tip payload of mass $m_L = 10\,\text{kg}$, design an independent joint control with feedforward computed torque. Perform a

computer simulation of the motion of the controlled arm along the joint space rectilinear path from $\boldsymbol{q}_i = [\,0 \quad \pi/4\,]^T$ to $\boldsymbol{q}_f = [\,\pi/2 \quad \pi/2\,]^T$ with a trapezoidal velocity profile and a trajectory duration $t_f = 1\,\text{s}$. Implement the control in discrete-time with a sampling time of 1 ms.

8.12. For the two-link planar arm of Problem 8.11, design an inverse dynamics joint control. Perform a computer simulation of the motion of the controlled arm along the trajectory specified in Problem 8.11. Implement the control in discrete-time with a sampling time of 1 ms.

8.13. For the two-link planar arm of Problem 8.11, design a robust joint control. Perform a computer simulation of the motion of the controlled arm along the trajectory specified in Problem 8.11. Implement the control in discrete-time with a sampling time of 1 ms.

8.14. For the two-link planar arm of Problem 8.11, design an adaptive joint control, on the basis of a suitable parameterization of the arm dynamic model. Perform a computer simulation of the motion of the controlled arm along the trajectory specified in Problem 8.11. Implement the control in discrete-time with a sampling time of 1 ms.

8.15. For the two-link planar of Problem 8.11, design a PD control in the operational space with gravity compensation. By means of a computer simulation, verify stability for the following postures $\boldsymbol{p} = [\,0.5 \quad 0.5\,]^T$ and $\boldsymbol{p} = [\,0.6 \quad -0.2\,]^T$, respectively. Implement the control in discrete-time with a sampling time of 1 ms.

8.16. For the two-link planar arm of Problem 8.11, design an inverse dynamics control in the operational space. Perform a computer simulation of the motion of the controlled arm along the operational space rectlinear path from $\boldsymbol{p}(0) = [\,0.7 \quad 0.2\,]^T$ to $\boldsymbol{p}(1) = [\,0.1 \quad -0.6\,]^T$ with a trapezoidal velocity profile and a trajectory duration $t_f = 1\,\text{s}$. Implement the control in discrete-time with a sampling time of 1 ms.