Bruno Siciliano • Lorenzo Sciavicco
Luigi Villani • Giuseppe Oriolo

Robotics

Modelling, Planning and Control

Springer

Bruno Siciliano • Lorenzo Sciavicco
Luigi Villani • Giuseppe Oriolo

Robotics

Modelling, Planning and Control

Springer

5

Actuators and Sensors

In this chapter, two basic robot components are treated: *actuators* and *sensors*. In the first part, the features of an *actuating system* are presented in terms of the power supply, power amplifier, servomotor and transmission. In view of their control versatility, two types of servomotors are used, namely, *electric servomotors* for actuating the joints of small and medium size manipulators, and *hydraulic servomotors* for actuating the joints of large size manipulators. The models describing the input/output relationship for such servomotors are derived, together with the control schemes of the *drives*. The electric servomotors are also employed to actuate the wheels of the mobile robots, which will be dealt with in Chap. 11. Successively, *proprioceptive sensors* are presented which allow measurement of the quantities characterizing the internal state of the manipulator, namely, *encoders* and *resolvers* for joint position measurement, *tachometers* for joint velocity measurement; further, *exteroceptive sensors* are presented including *force sensors* for end-effector force measurement, *distance sensors* for detection of objects in the workspace, and *vision sensors* for the measurement of the characteristic parameters of such objects, whenever the manipulator interacts with the environment.

5.1 Joint Actuating System

The motion imposed to a manipulator's joint is realized by an *actuating system* which in general consists of:

- a *power supply*,
- a *power amplifier*,
- a *servomotor*,
- a *transmission*.

The connection between the various components is illustrated in Fig. 5.1 where the exchanged powers are shown. To this end, recall that power can always be expressed as the product of a flow and a force quantity, whose physical context allows the specification of the nature of the power (mechanical, electric, hydraulic, or pneumatic).

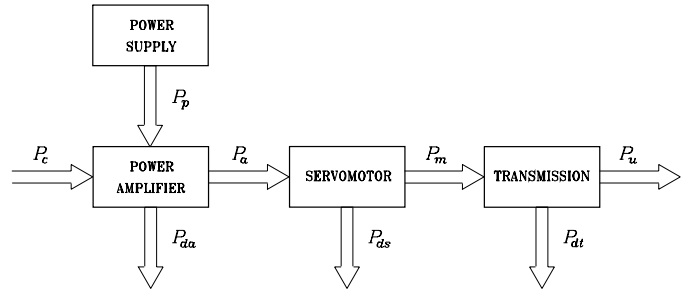

Fig. 5.1. Components of a joint actuating system

In terms of a global input/output relationship, P_c denotes the (usually electric) power associated with the control law signal, whereas P_u represents the mechanical power required to the joint to actuate the motion. The intermediate connections characterize the supply power P_a of the motor (of electric, hydraulic, or pneumatic type), the power provided by the primary source P_p of the same physical nature as that of P_a, and the mechanical power P_m developed by the motor. Moreover, P_{da}, P_{ds} and P_{dt} denote the powers lost for dissipation in the conversions performed respectively by the amplifier, motor and transmission.

To choose the components of an actuating system, it is worth starting from the requirements imposed on the mechanical power P_u by the force and velocity that describe the joint motion.

5.1.1 Transmissions

The execution of joint motions of a manipulator demands *low speeds* with *high torques*. In general, such requirements do not allow an effective use of the mechanical features of servomotors, which typically provide high speeds with low torques in optimal operating conditions. It is then necessary to interpose a *transmission* (*gear*) to optimize the transfer of mechanical power from the motor (P_m) to the joint (P_u). During this transfer, the power P_{dt} is dissipated as a result of friction.

The choice of the transmission depends on the power requirements, the kind of desired motion, and the allocation of the motor with respect to the joint. In fact, the transmission allows the outputs of the motor to be transformed both quantitatively (velocity and torque) and qualitatively (a rotational motion about the motor axis into a translational motion of the joint). Also, it allows the static and dynamic performance of a manipulator to be optimized, by reducing the effective loads when the motor is located upstream

of the joint; for instance, if some motors are mounted to the base of the robot, the total weight of the manipulator is decreased and the power-to-weight ratio is increased.

The following transmissions are typically used for industrial robots:

- *Spur gears* that modify the characteristics of the rotational motion of the motor by changing the axis of rotation and/or by translating the application point; spur gears are usually constructed with wide cross-section teeth and squat shafts.
- *Lead screws* that convert rotational motion of the motor into translational motion, as needed for actuation of prismatic joints; in order to reduce friction, ball screws are usually employed that are preloaded so as to increase stiffness and decrease backlash.
- *Timing belts* and *chains* which are equivalent from a kinematic viewpoint and are employed to locate the motor remotely from the axis of the actuated joint. The stress on timing belts may cause strain, and then these are used in applications requiring high speeds and low forces. On the other hand, chains are used in applications requiring low speeds, since their large mass may induce vibration at high speeds.

On the assumption of rigid transmissions with no backlash, the relationship between input forces (velocities) and output forces (velocities) is purely proportional.

The mechanical features of the motor used for an actuating system may sometimes allow a direct connection of the motor to the joint without the use of any transmission element (*direct drive*). The drawbacks due to transmission elasticity and backlash are thus eliminated, although more sophisticated control algorithms are required, since the absence of reduction gears does not allow the nonlinear coupling terms in the dynamic model to be neglected. The use of direct-drive actuating systems is not yet popular for industrial manipulators, in view of the cost and size of the motors as well as of control complexity.

5.1.2 Servomotors

Actuation of joint motions is entrusted to *motors* which allow the realization of a desired motion for the mechanical system. Concerning the kind of input power P_a, motors can be classified into three groups:

- *Pneumatic motors* which utilize the pneumatic energy provided by a compressor and transform it into mechanical energy by means of pistons or turbines.
- *Hydraulic motors* which transform the hydraulic energy stored in a reservoir into mechanical energy by means of suitable pumps.
- *Electric motors* whose primary supply is the electric energy available from the electric distribution system.

A portion of the input power P_a is converted to output as mechanical power P_m, and the rest (P_{ds}) is dissipated because of mechanical, electric, hydraulic, or pneumatic loss.

The motors employed in robotics are the evolution of the motors employed in industrial automation having powers ranging from about 10 W to about 10 kW. For the typical performance required, such motors should have the following requirements with respect to those employed in conventional applications:

- low inertia and high power-to-weight ratio,
- possibility of overload and delivery of impulse torques,
- capability to develop high accelerations,
- wide velocity range (from 1 to 1000 revolutes/min),
- high positioning accuracy (at least 1/1000 of a circle),
- low torque ripple so as to guarantee continuous rotation even at low speed.

These requirements are enhanced by the good trajectory tracking and positioning accuracy demanded for an actuating system for robots, and thus the motor must play the role of a *servomotor*. In this respect, pneumatic motors are difficult to control accurately, in view of the unavoidable fluid compressibility errors. Therefore, they are not widely employed, if not for the actuation of the typical opening and closing motions of the jaws in a gripper tool, then for the actuation of simple arms used in applications where continuous motion control is not of concern.

The most employed motors in robotics applications are *electric servomotors*. Among them, the most popular are permanent-magnet direct-current (DC) servomotors and brushless DC servomotors, in view of their good control flexibility.

The *permanent-magnet DC servomotor* consists of:

- A stator coil that generates magnetic flux; this generator is always a permanent magnet made by ferromagnetic ceramics or rare earths (high fields in contained space).
- An armature that includes the current-carrying winding that surrounds a rotary ferromagnetic core (rotor).
- A commutator that provides an electric connection by means of brushes between the rotating armature winding and the external feed winding, according to a commutation logic determined by the rotor motion.

The *brushless DC servomotor* consists of:

- A rotating coil (rotor) that generates magnetic flux; this generator is a permanent magnet made by ferromagnetic ceramics or rare earths.
- A stationary armature (stator) made by a polyphase winding.
- A static commutator that, on the basis of the signals provided by a position sensor located on the motor shaft, generates the feed sequence of the armature winding phases as a function of the rotor motion.

With reference to the above details of constructions, a comparison between the operating principle of a permanent-magnet DC and a brushless DC servomotor leads to the following considerations.

In the brushless DC motor, by means of the rotor position sensor, the winding orthogonal to the magnetic field of the coil is found; then, feeding the winding makes the rotor rotate. As a consequence of rotation, the electronic control module commutes the feeding on the winding of the various phases in such a way that the resulting field at the armature is always kept orthogonal to that of the coil. As regards electromagnetic interaction, such a motor operates in a way similar to that of a permanent-magnet DC motor where the brushes are at an angle of $\pi/2$ with respect to the direction of the excitation flux. In fact, feeding the armature coil makes the rotor rotate, and commutation of brushes from one plate of the commutator to the other allows the rotor to be maintained in rotation. The role played by the brushes and commutator in a permanent-magnet DC motor is analogous to that played by the position sensor and electronic control module in a brushless DC motor.

The main reason for using a brushless DC motor is to eliminate the problems due to mechanical commutation of the brushes in a permanent-magnet DC motor. In fact, the presence of the commutator limits the performance of a permanent-magnet DC motor, since this provokes electric loss due to voltage drops at the contact between the brushes and plates, and mechanical loss due to friction and arcing during commutation from one plate to the next one caused by the inductance of the winding. The elimination of the causes provoking such inconveniences, i.e., the brushes and plates, allows an improvement of motor performance in terms of higher speeds and less material wear.

The inversion between the functions of stator and rotor leads to further advantages. The presence of a winding on the stator instead of the rotor facilitates heat disposal. The absence of a rotor winding, together with the possibility of using rare-earth permanent magnets, allows construction of more compact rotors which are, in turn, characterized by a low moment of inertia. Therefore, the size of a brushless DC motor is smaller than that of a permanent-magnet DC motor of the same power; an improvement of dynamic performance can also be obtained by using a brushless DC motor. For the choice of the most suitable servomotor for a specific application, the cost factor plays a relevant role.

Not uncommon are also stepper motors. These actuators are controlled by suitable excitation sequences and their operating principle does not require measurement of motor shaft angular position. The dynamic behaviour of stepper motors is greatly influenced by payload, though. Also, they induce vibration of the mechanical structure of the manipulator. Such inconveniences confine the use of stepper motors to the field of micromanipulators, for which low-cost implementation prevails over the need for high dynamic performance.

A certain number of applications features the employment of *hydraulic servomotors*, which are based on the simple operating principle of volume

variation under the action of compressed fluid. From a construction viewpoint, they are characterized by one or more chambers made by pistons (cylinders reciprocating in tubular housings). Linear servomotors have a limited range and are constituted by a single piston. Rotary servomotors have unlimited range and are constituted by several pistons (usually an odd number) with an axial or radial disposition with respect to the motor axis of rotation. These servomotors offer a static and dynamic performance comparable with that offered by electric servomotors.

The differences between electric and hydraulic servomotors can be fundamentally observed from a plant viewpoint. In this respect, *electric servomotors* present the following *advantages*:

- widespread availability of power supply,
- low cost and wide range of products,
- high power conversion efficiency,
- easy maintenance,
- no pollution of working environment.

Instead, they present the following *limitations*:

- burnout problems at static situations caused by the effect of gravity on the manipulator; emergency brakes are then required,
- need for special protection when operating in flammable environments.

Hydraulic servomotors present the following *drawbacks*:

- need for a hydraulic power station,
- high cost, narrow range of products, and difficulty of miniaturization,
- low power conversion efficiency,
- need for operational maintenance,
- pollution of working environment due to oil leakage.

In their *favour* it is worth pointing out that they:

- do not suffer from burnout in static situations,
- are self-lubricated and the circulating fluid facilitates heat disposal,
- are inherently safe in harmful environments,
- have excellent power-to-weight ratios.

From an operational viewpoint, it can be observed that:

- Both types of servomotors have a good dynamic behaviour, although the electric servomotor has greater control flexibility. The dynamic behaviour of a hydraulic servomotor depends on the temperature of the compressed fluid.
- The electric servomotor is typically characterized by high speeds and low torques, and as such it requires the use of gear transmissions (causing elasticity and backlash). On the other hand, the hydraulic servomotor is capable of generating high torques at low speeds.

In view of the above remarks, hydraulic servomotors are specifically employed for manipulators that have to carry heavy payloads; in this case, not only is the hydraulic servomotor the most suitable actuator, but also the cost of the plant accounts for a reduced percentage on the total cost of the manipulation system.

5.1.3 Power Amplifiers

The *power amplifier* has the task of modulating, under the action of a control signal, the power flow which is provided by the primary supply and has to be delivered to the actuators for the execution of the desired motion. In other words, the amplifier takes a fraction of the power available at the source which is proportional to the control signal; then it transmits this power to the motor in terms of suitable force and flow quantities.

The inputs to the amplifier are the power taken from the primary source P_p and the power associated with the control signal P_c. The total power is partly delivered to the actuator (P_a) and partly lost in dissipation (P_{da}).

Given the typical use of electric and hydraulic servomotors, the operational principles of the respective amplifiers are discussed.

To control an *electric servomotor*, it is necessary to provide it with a voltage or current of suitable form depending on the kind of servomotor employed. Voltage (or current) is direct for permanent-magnet DC servomotors, while it is alternating for brushless DC servomotors. The value of voltage for permanent-magnet DC servomotors or the values of voltage and frequency for brushless DC servomotors are determined by the control signal of the amplifier, so as to make the motor execute the desired motion.

For the power ranges typically required by joint motions (of the order of a few kilowatts), transistor amplifiers are employed which are suitably switched by using pulse-width modulation (PWM) techniques. They allow the achievement of a power conversion efficiency $P_a/(P_p + P_c)$ greater than 0.9 and a power gain P_a/P_c of the order of 10^6. The amplifiers employed to control permanent-magnet DC servomotors are DC-to-DC converters (*choppers*), whereas those employed to control brushless DC servomotors are DC-to-AC converters (*inverters*).

Control of a *hydraulic servomotor* is performed by varying the flow rate of the compressed fluid delivered to the motor. The task of modulating the flow rate is typically entrusted to an interface (electro-hydraulic servovalve). This allows a relationship to be established between the electric control signal and the position of a distributor which is able to vary the flow rate of the fluid transferred from the primary source to the motor. The electric control signal is usually current-amplified and feeds a solenoid which moves (directly or indirectly) the distributor, whose position is measured by a suitable transducer. In this way, a position servo on the valve stem is obtained which reduces occurrence of any stability problem that may arise on motor control. The magnitude of the control signal determines the flow rate of the compressed

fluid through the distributor, according to a characteristic which is possibly made linear by means of a keen mechanical design.

5.1.4 Power Supply

The task of the *power supply* is to supply the primary power to the amplifier which is needed for operation of the actuating system.

In the case of *electric servomotors*, the power supply consists of a transformer and a typically uncontrolled bridge rectifier. These allow the alternating voltage available from the distribution to be converted into a direct voltage of suitable magnitude which is required to feed the power amplifier.

In the case of *hydraulic servomotors*, the power supply is obviously more complex. In fact, a gear or piston pump is employed to compress the fluid which is driven by a primary motor operating at constant speed, typically a three-phase nonsynchronous motor. To reduce the unavoidable pressure oscillations provoked by a flow rate demand depending on operational conditions of the motor, a reservoir is interfaced to store hydraulic energy. Such a reservoir, in turn, plays the same role as the filter capacitor used at the output of a bridge rectifier. The hydraulic power station is completed by the use of various components (filters, pressure valves, and check valves) that ensure proper operation of the system. Finally, it can be inferred how the presence of complex hydraulic circuits operating at high pressures (of the order of 100 atm) causes an appreciable pollution of the working environment.

5.2 Drives

This section presents the operation of the *electric drives* and the *hydraulic drives* for the actuation of a manipulator's joints. Starting from the mathematical models describing the dynamic behaviour, the block schemes are derived which allow an emphasis on the control features and the effects of the use of a mechanical transmission.

5.2.1 Electric Drives

From a modelling viewpoint, a permanent-magnet DC motor and a brushless DC motor provided with the commutation module and position sensor can be described by the same differential equations. In the domain of the complex variable s, the electric balance of the armature is described by the equations

$$V_a = (R_a + sL_a)I_a + V_g \tag{5.1}$$

$$V_g = k_v \Omega_m \tag{5.2}$$

where V_a and I_a respectively denote armature voltage and current, R_a and L_a are respectively the armature resistance and inductance, and V_g denotes

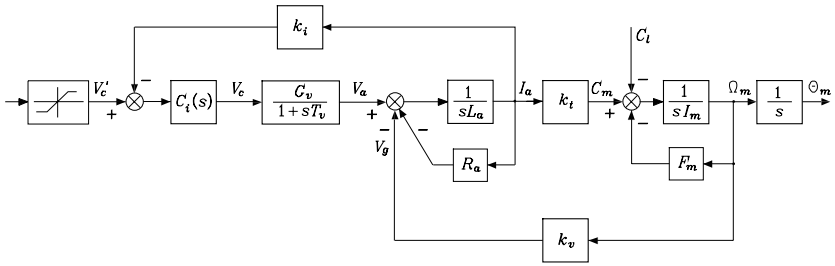

Fig. 5.2. Block scheme of an electric drive

the back electromotive force which is proportional to the angular velocity Ω_m through the voltage constant k_v that depends on the construction details of the motor as well as on the magnetic flux of the coil.

The mechanical balance is described by the equations

$$C_m = (sI_m + F_m)\Omega_m + C_l \tag{5.3}$$

$$C_m = k_t I_a \tag{5.4}$$

where C_m and C_l respectively denote the driving torque and load reaction torque, I_m and F_m are respectively the moment of inertia and viscous friction coefficient at the motor shaft, and the torque constant k_t is numerically equal to k_v in the SI unit system for a compensated motor.

Concerning the power amplifier, the input/output relationship between the control voltage V_c and the armature voltage V_a is given by the transfer function

$$\frac{V_a}{V_c} = \frac{G_v}{1 + sT_v} \tag{5.5}$$

where G_v denotes the voltage gain and T_v is a time constant that can be neglected with respect to the other time constants of the system. In fact, by using a modulation frequency in the range of 10 to 100 kHz, the time constant of the amplifier is in the range of 10^{-5} to 10^{-4}) s.

The block scheme of the servomotor with power amplifier (*electric drive*) is illustrated in Fig. 5.2. In such a scheme, besides the blocks corresponding to the above relations, there is an armature *current feedback* loop where current is thought of as measured by a transducer k_i between the power amplifier and the armature winding of the motor. Further, the scheme features a current regulator $C_i(s)$ as well as an element with a nonlinear saturation characteristic. The aim of such feedback is twofold. On one hand, the voltage V_c' plays the role of a current reference and thus, by means of a suitable choice of the regulator $C_i(s)$, the lag between the current I_a and the voltage V_c' can be reduced with respect to the lag between I_a and V_c. On the other hand, the introduction of a saturation nonlinearity allows the limitation of the magni-

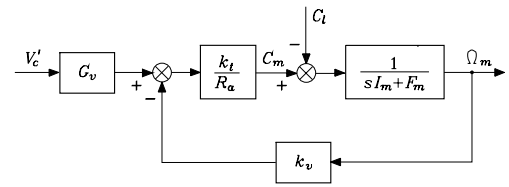

Fig. 5.3. Block scheme of an electric drive as a velocity-controlled generator

tude of V_c', and then it works like a current limit which ensures protection of the power amplifier whenever abnormal operating conditions occur.

The choice of the regulator $C_i(s)$ of the current loop allows a velocity-controlled or torque-controlled behaviour to be obtained from the electric drive, depending on the values attained by the loop gain. In fact, in the case of $k_i = 0$, recalling that the mechanical viscous friction coefficient is negligible with respect to the electrical friction coefficient

$$F_m \ll \frac{k_v k_t}{R_a}, \tag{5.6}$$

assuming a unit gain constant for $C_i(s)$[1] and $C_l = 0$ yields

$$\omega_m \approx \frac{G_v}{k_v} v_c' \tag{5.7}$$

and thus the drive system behaves like a *velocity-controlled generator*.

Instead, when $k_i \neq 0$, choosing a large loop gain for the current loop ($Kk_i \gg R_a$) leads at steady state to

$$c_m \approx \frac{k_t}{k_i}\left(v_c' - \frac{k_v}{G_v}\omega_m\right); \tag{5.8}$$

the drive behaves like a *torque-controlled generator* since, in view of the large value of G_v, the driving torque is practically independent of the angular velocity.

As regards the dynamic behaviour, it is worth considering a *reduced-order model* which can be obtained by neglecting the electric time constant L_a/R_a with respect to the mechanical time constant I_m/F_m, assuming $T_v \approx 0$ and a purely proportional controller. These assumptions, together with $k_i = 0$, lead to the block scheme in Fig. 5.3 for the velocity-controlled generator. On the other hand, if it is assumed $Kk_i \gg R_a$ and $k_v\Omega/Kk_i \approx 0$, the resulting block scheme of the torque-controlled generator is that in Fig. 5.4. From the

[1] It is assumed $C_i(0) = 1$; in the case of presence of an integral action in $C_i(s)$, it should be $\lim_{s\to 0} sC(s) = 1$.

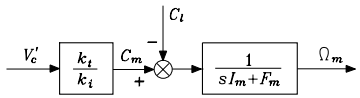

Fig. 5.4. Block scheme of an electric drive as a torque-controlled generator

above schemes, the following input/output relations between control voltage, reaction torque, and angular velocity can be derived:

$$\Omega_m = \frac{\frac{1}{k_v}}{1 + s\frac{R_a I_m}{k_v k_t}} G_v V_c' - \frac{\frac{R_a}{k_v k_t}}{1 + s\frac{R_a I_m}{k_v k_t}} C_l \tag{5.9}$$

for the velocity-controlled generator, and

$$\Omega_m = \frac{\frac{k_t}{k_i F_m}}{1 + s\frac{I_m}{F_m}} V_c' - \frac{\frac{1}{F_m}}{1 + s\frac{I_m}{F_m}} C_l \tag{5.10}$$

for the torque-controlled generator. These transfer functions show how, without current feedback, the system has a better rejection of disturbance torques in terms of both equivalent gain ($R_a/k_v k_t \ll 1/F_m$) and time response ($R_a I_m/k_v k_t \ll I_m/F_m$).

The relationship between the control input and the actuator position output can be expressed in a unified manner by the transfer function

$$M(s) = \frac{k_m}{s(1 + sT_m)} \tag{5.11}$$

where

$$k_m = \frac{1}{k_v} \qquad T_m = \frac{R_a I_m}{k_v k_t} \tag{5.12}$$

for the velocity-controlled generator, while for the torque-controlled generator it is

$$k_m = \frac{k_t}{k_i F_m} \qquad T_m = \frac{I_m}{F_m}. \tag{5.13}$$

Notice how the power amplifier, in the velocity control case, contributes to the input/output relation with the constant G_v, while in the case of current control the amplifier, being inside a local feedback loop, does not appear as a stand alone but rather in the expression of k_m with a factor $1/k_i$.

These considerations lead to the following conclusions. In all such applications where the drive system has to provide high rejection of disturbance torques (as in the case of independent joint control, see Sect. 8.3) it is not advisable to have a current feedback in the loop, at least when all quantities

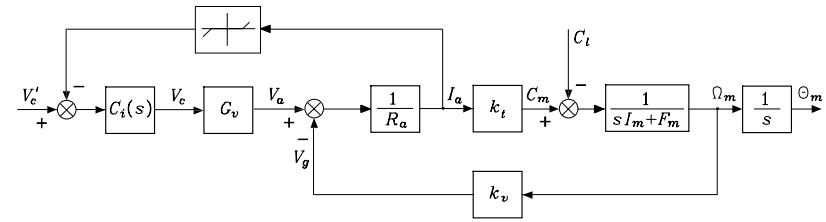

Fig. 5.5. Block scheme of an electric drive with nonlinear current feedback

are within their nominal values. In this case, the problem of setting a protection can be solved by introducing a current limit that is not performed by a saturation on the control signal but it exploits a current feedback with a dead-zone nonlinearity on the feedback path, as shown in Fig. 5.5. Therefore, an actual current limit is obtained whose precision is as high as the slope of the dead zone; it is understood that stability of the current loop is to be addressed when operating in this way.

As will be shown in Sect. 8.5, centralized control schemes, instead, demand the drive system to behave as a torque-controlled generator. It is then clear that a current feedback with a suitable regulator $C_i(s)$ should be used so as to confer a good static and dynamic behaviour to the current loop. In this case, servoing of the driving torque is achieved indirectly, since it is based on a current measurement which is related to the driving torque by means of gain $1/k_t$.

5.2.2 Hydraulic Drives

No matter how a hydraulic servomotor is constructed, the derivation of its input/output mathematical model refers to the basic equations describing the relationship between flow rate and pressure, the relationship between the fluid and the parts in motion, and the mechanical balance of the parts in motion. Let Q represent the volume flow rate supplied by the distributor; the flow rate balance is given by the equation

$$Q = Q_m + Q_l + Q_c \tag{5.14}$$

where Q_m is the flow rate transferred to the motor, Q_l is the flow rate due to leakage, and Q_c is the flow rate related to fluid compressibility. The terms Q_l and Q_c are taken into account in view of the high operating pressures (of the order of 100 atm).

Let P denote the differential pressure of the servomotor due to the load; then it can be assumed that

$$Q_l = k_l P. \tag{5.15}$$

Regarding the loss for compressibility, if V denotes the instantaneous volume of the fluid, one has

$$Q_c = \gamma V s P \tag{5.16}$$

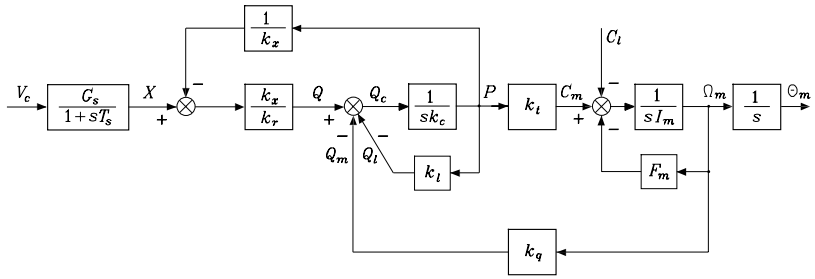

Fig. 5.6. Block scheme of a hydraulic drive

where γ is the uniform compressibility coefficient of the fluid. Notice that the proportional factor $k_c = \gamma V$ between the time derivative of the pressure and the flow rate due to compressibility depends on the volume of the fluid; therefore, in the case of rotary servomotors, k_c is a constant, whereas in the case of a linear servomotor, the volume of fluid varies and thus the characteristic of the response depends on the operating point.

The volume flow rate transferred to the motor is proportional to the volume variation in the chambers per time unit; with reference from now on to a rotary servomotor, such variation is proportional to the angular velocity, and then

$$Q_m = k_q \Omega_m. \tag{5.17}$$

The mechanical balance of the parts in motion is described by

$$C_m = (sI_m + F_m)\Omega_m + C_l \tag{5.18}$$

with obvious meaning of the symbols. Finally, the driving torque is proportional to the differential pressure of the servomotor due to the load, i.e.,

$$C_m = k_t P. \tag{5.19}$$

Concerning the servovalve, the transfer function between the stem position X and the control voltage V_c is expressed by

$$\frac{X}{V_c} = \frac{G_s}{1 + sT_s} \tag{5.20}$$

thanks to the linearizing effect achieved by position feedback; G_s is the equivalent gain of the servovalve, whereas its time constant T_s is of the order of ms and thus it can be neglected with respect to the other time constants of the system.

Finally, regarding the distributor, the relationship between the differential pressure, the flow rate, and the stem displacement is highly nonlinear; linearization about an operating point leads to the equation

$$P = k_x X - k_r Q. \tag{5.21}$$

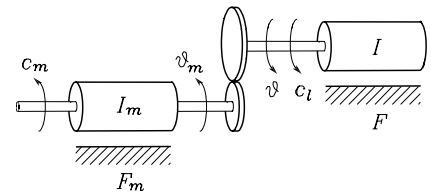

Fig. 5.7. Schematic representation of a mechanical gear

By virtue of (5.14)–(5.21), the servovalve/distributor/motor complex (*hydraulic drive*) is represented by the block scheme of Fig. 5.6. A comparison between the schemes in Figs. 5.2 and 5.6 clearly shows the formal analogy in the dynamic behaviour of an electric and a hydraulic servomotor. Nevertheless, such analogy should not induce one to believe that it is possible to make a hydraulic drive play the role of a velocity- or torque-controlled generator, as for an electric drive. In this case, the pressure feedback loop (formally analogous to the current feedback loop) is indeed a structural characteristic of the system and, as such, it cannot be modified but with the introduction of suitable transducers and the realization of the relative control circuitry.

5.2.3 Transmission Effects

In order to describe quantitatively the effects introduced by the use of a transmission (*mechanical gear*) between the servomotor and the actuated joint, it is worth referring to the mechanical coupling realized by a pair of spur gears of radius r_m and r, which is schematically represented in Fig. 5.7; the kinematic pair is assumed to be ideal (without backlash) and connects the rotation axis of the servomotor with the axis of the corresponding joint.

With reference to an electric servomotor, it is assumed that the rotor of the servomotor is characterized by an inertia moment I_m about its rotation axis and a viscous friction coefficient F_m; likewise, I and F denote the inertia moment and the viscous friction coefficient of the load. The inertia moments and the friction coefficients of the gears are assumed to have been included in the corresponding parameters of the motor (for the gear of radius r_m) and of the load (for the gear of radius r). Let c_m denote the driving torque of the motor and c_l the reaction torque applied to the load axis. Also let ω_m and ϑ_m denote the angular velocity and position of the motor axis, while ω and ϑ denote the corresponding quantities at the load side. Finally, f indicates the force exchanged at the contact between the teeth of the two gears.[2]

[2] In the case considered, it has been assumed that both the motor and the load are characterized by revolute motions; if the load should exhibit a translation motion, the following arguments can be easily extended, with analogous results, by replacing the angular displacements with linear displacements and the inertia moments with masses at the load side.

The gear reduction ratio is defined as

$$k_r = \frac{r}{r_m} = \frac{\vartheta_m}{\vartheta} = \frac{\omega_m}{\omega} \tag{5.22}$$

since, in the absence of slipping in the kinematic coupling, it is $r_m \vartheta_m = r\vartheta$.

The gear reduction ratio, in the case when it is representative of the coupling between a servomotor and the joint of a robot manipulator, attains values much larger than unity ($r_m \ll r$) — typically from a few tens to a few hundreds.

The force f exchanged between the two gears generates a reaction torque $f \cdot r_m$ for the motion at the motor axis and a driving torque $f \cdot r$ for the rotation motion of the load.

The mechanical balances at the motor side and the load side are respectively:

$$c_m = I_m \dot{\omega}_m + F_m \omega_m + f r_m \tag{5.23}$$

$$f r = I \dot{\omega} + F \omega + c_l. \tag{5.24}$$

To describe the motion with reference to the motor angular velocity, in view of (5.22), combining the two equations gives at the motor side

$$c_m = I_{eq} \dot{\omega}_m + F_{eq} \omega_m + \frac{c_l}{k_r} \tag{5.25}$$

where

$$I_{eq} = \left(I_m + \frac{I}{k_r^2} \right) \qquad F_{eq} = \left(F_m + \frac{F}{k_r^2} \right). \tag{5.26}$$

The expressions (5.25), (5.26) show how, in the case of a gear with large reduction ratio, the inertia moment and the viscous friction coefficient of the load are reflected at the motor axis with a reduction of a factor $1/k_r^2$; the reaction torque, instead, is reduced by a factor $1/k_r$. If this torque depends on ϑ in a nonlinear fashion, then the presence of a large reduction ratio tends to linearize the dynamic equation.

Example 5.1

In Fig. 5.8 a rigid pendulum is represented, which is actuated by the torque $f \cdot r$ to the load axis after the gear. In this case, the dynamic equations of the system are

$$c_m = I_m \dot{\omega}_m + F_m \omega_m + f r_m \tag{5.27}$$

$$f r = I \dot{\omega} + F \omega + mg\ell \sin \vartheta \tag{5.28}$$

where I is the inertia moment of the pendulum at the load axis, F is the viscous friction coefficient, m is the pendulum mass, ℓ its length and g the gravity acceleration. Reporting (5.28) to the motor axis gives

$$c_m = I_{eq} \dot{\omega}_m + F_{eq} \omega_m + \left(\frac{mg\ell}{k_r} \right) \sin\left(\frac{\vartheta_m}{k_r} \right) \tag{5.29}$$

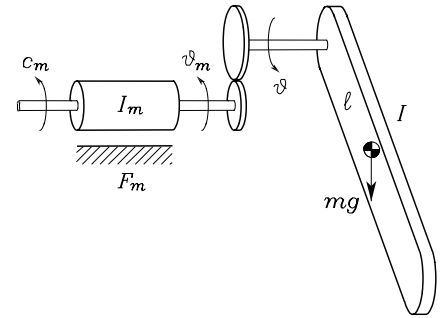

Fig. 5.8. Pendulum actuated via mechanical gear

from which it is clear how the contribution of the nonlinear term is reduced by the factor k_r.

The example of the pendulum has been considered to represent an n-link manipulator with revolute joints, for which each link, considered as isolated from the others, can be considered as a simple rigid pendulum. The connection with other links introduces, in reality, other nonlinear effects which complicate the input/output model; in this regard, it is sufficient to notice that, in the case of a double pendulum, the inertia moment at the motor side of the first link depends also on the angular position of the second link.

In Chap. 7 the effect introduced by the presence of transmissions in a generic n-link manipulator structure will be studied in detail. Nevertheless, it can already be understood how the nonlinear couplings between the motors of the various links will be reduced by the presence of transmissions with large reduction ratios.

5.2.4 Position Control

After having examined the modalities to control the angular velocity of an electric or hydraulic drive, the motion control problem for a link of a generic manipulator is to be solved. A structure is sought which must be capable of determining, in an automatic way, the time evolution of the quantity chosen to control the drive, so that the actuated joint executes the required motion allowing the end-effector to execute a given task.

Once a trajectory has been specified for the end-effector pose, the solution of the inverse kinematics problem allows the computation of the desired trajectories for the various joints, which thus can be considered as available.

Several control techniques can be adopted to control the manipulator motion; the choice of a particular solution depends on the required dynamic performance, the kind of motion to execute, the kinematic structure, and the

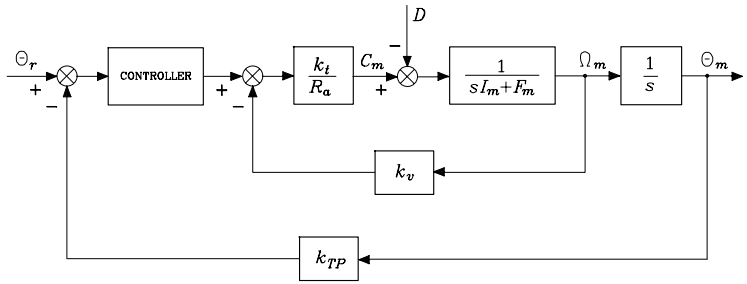

Fig. 5.9. General block scheme of electric drive control

choice to utilize either servomotors with transmissions or torque motors with joint direct drive.

The simplest solution is to consider, at first approximation, the motion of a joint independent of the motion of the other joints, i.e., the interaction can be regarded as a disturbance. Assume the reference trajectory $\vartheta_r(t)$ is available. According to classical automatic control theory, to ensure that the angular motor position ϑ_m, properly measured by means of a transducer with constant k_{TP}, follows ϑ_r, it is worth resorting to a feedback control system providing 'robustness' with respect to both model uncertainty on the motor and the load, and the presence of a disturbance. A more detailed treatment is deferred to Chap. 8, where the most congenial solutions to solve the above problems will be presented.

In the following, the problem of joint *position control* is tackled by assuming an electric DC servomotor; the choice is motivated by the diffusion of this technology, due to the high flexibility of these actuators providing optimal responses in the large majority of motion control applications.

The choice of a feedback control system to realize a position servo at the motor axis requires the adoption of a *controller*; this device generates a signal which, applied to the power amplifier, automatically generates the driving torque producing an axis motion very close to the desired motion ϑ_r. Its structure should be so that the error between the reference input and the measured output is minimized, even in the case of inaccurate knowledge of the dynamics of the motor, the load, and a disturbance. The *rejection* action of the disturbance is the more efficient, the smaller the magnitude of the disturbance.

On the other hand, according to (5.9), the disturbance is minimized, provided the drive is velocity-controlled. In this case, in view of (5.6), the reaction torque influences the motor axis velocity with a coefficient equal to $R_a/k_v k_t$ which is much smaller than $1/F_m$, which represents instead the weight on the reaction torque in the case when the drive is torque-controlled. Therefore, with reference to Fig. 5.3, the general scheme of drive control with *position feedback* is illustrated in Fig. 5.9, where the disturbance d represents the load

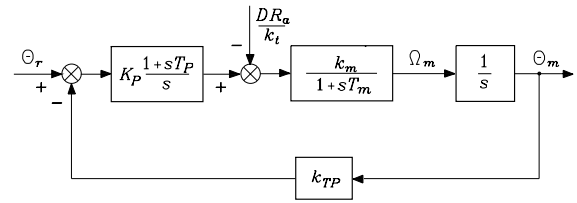

Fig. 5.10. Block scheme of drive control with position feedback

torque and the value of the power amplifier gain has been included in the control action.

Besides reducing the effects of the disturbance on the output, the structure of the controller must ensure an optimal trade-off between the stability of the feedback control system and the capability of the output to dynamically track the reference with a reduced error.

The reduction of the disturbance effects on the output can be achieved by conferring a large value of the gain before the point of intervention of the disturbance, without affecting stability. If, at steady state (ϑ_r = cost, c_l = cost), it is desired to cancel the disturbance effect on the output, the controller must act an *integral action* on the error given by the difference between ϑ_r and $k_{TP}\vartheta_m$.

The above requirements suggest the use of a simple controller with an integral and a proportional action on the error; the *proportional action* is added to realize a stabilizing action, which, however, cannot confer to the closed-loop system a damped transient response with a sufficiently short sampling time. This behaviour is due to the presence of a double pole at the origin of the transfer function of the forward path.

The resulting control scheme is illustrated in Fig. 5.10, where k_m and T_m are respectively the voltage-to-velocity gain constant and the characteristic time constant of the motor in (5.12). The parameters of the controller K_P and T_P should be keenly chosen so as to ensure stability of the feedback control system and obtain a good dynamic behaviour.

To improve the transient response, the industrial drives employed for position servoing may also include a local feedback loop based on the angular velocity measurement (tachometer feedback). The general scheme with *position and velocity feedback* is illustrated in Fig. 5.11; besides the position transducer, a velocity transducer is used with constant k_{TV}, as well as a simple proportional controller with gain K_P. With the adoption of the tachometer feedback, the proportional-integral controller with parameters K_V and T_V is retained in the internal velocity loop so as to cancel the effects of the disturbance on the position ϑ_m at steady state. The presence of two feedback loops, in lieu of one, around the intervention point of the disturbance is expected to lead to a further reduction of the disturbance effects on the output also during the transients.

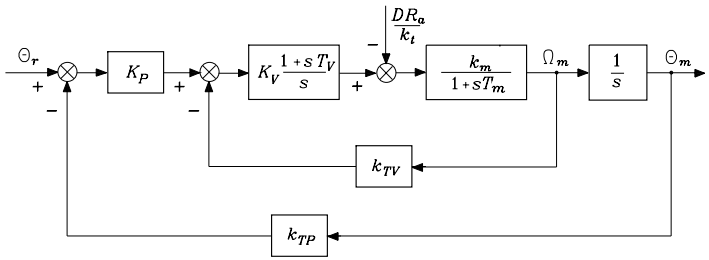

Fig. 5.11. Block scheme of drive control with position and velocity feedback

The adoption of tachometer feedback may also improve the transient response of the whole control system with respect to the previous case. With a keen choice of the controller parameters, indeed, it is possible to achieve a transfer function between ϑ_m and ϑ_r with a larger bandwidth and reduced resonance phenomena. The result is a faster transient response with reduced oscillations, thus improving the capability of $\vartheta_m(t)$ to track more demanding reference trajectories $\vartheta_r(t)$.

The above analysis will be further detailed in Sect. 8.3.

The position servo may also utilize a current-controller motor; the schemes in Figs. 5.9–5.11 can be adopted, provided that the constants in (5.13) are used in the transfer function (5.11) and the disturbance D is weighed with the quantity k_i/k_t in lieu of R_a/k_t. In that case, the voltage gain G_v of the power amplifier will not contribute to the control action.

As a final consideration, the general control structure presented above may be extended to the case when the motor is coupled to a load via a gear reduction. In such a case, it is sufficient to account for (5.25) and (5.26), i.e., replace I_m and F_m with the quantities I_{eq} and F_{eq}, and scale the disturbance by the factor $1/k_r$.

5.3 Proprioceptive Sensors

The adoption of *sensors* is of crucial importance to achieve high-performance robotic systems. It is worth classifying sensors into *proprioceptive* sensors that measure the internal state of the manipulator, and *exteroceptive* sensors that provide the robot with knowledge of the surrounding environment.

In order to guarantee that a coordinated motion of the mechanical structure is obtained in correspondence of the task planning, suitable parameter identification and control algorithms are used which require the on-line measurement, by means of proprioceptive sensors, of the quantities characterizing the internal state of the manipulator, i.e.:

- joint positions,
- joint velocities,
- joint torques.

On the other hand, typical exteroceptive sensors include:

- force sensors,
- tactile sensors,
- proximity sensors,
- range sensors,
- vision sensors.

The goal of such sensors is to extract the features characterizing the interaction of the robot with the objects in the environment, so as to enhance the degree of autonomy of the system. To this class also belong those sensors which are specific for the robotic application, such as sound, humidity, smoke, pressure, and temperature sensors. Fusion of the available sensory data can be used for (high-level) task planning, which in turn characterizes a *robot* as the *intelligent connection of perception to action*.

In the following, the main features of the proprioceptive sensors are illustrated, while those of the exteroceptive sensors will be presented in the next section.

5.3.1 Position Transducers

The aim of *position transducers* is to provide an electric signal proportional to the linear or angular displacement of a mechanical apparatus with respect to a given reference position. They are mostly utilized for control of machine tools, and thus their range is wide. Potentiometers, linear variable-differential transformers (LVDT), and inductosyns may be used to measure linear displacements. Potentiometers, encoders, resolvers and synchros may be used to measure angular displacements.

Angular displacement transducers are typically employed in robotics applications since, also for prismatic joints, the servomotor is of a rotary type. In view of their precision, robustness and reliability, the most common transducers are the *encoders* and *resolvers*, whose operating principles are detailed in what follows.

On the other hand, linear displacement transducers (LVDT's and inductosyns) are mainly employed in measuring robots.

Encoder

There are two types of encoder: absolute and incremental. The *absolute encoder* consists of an optical-glass disk on which concentric circles (tracks) are disposed; each track has an alternating sequence of transparent sectors and matte sectors obtained by deposit of a metallic film. A light beam is emitted in

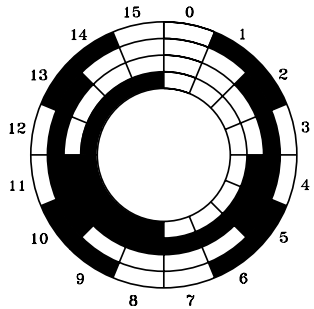

Fig. 5.12. Schematic representation of an absolute encoder

correspondence of each track which is intercepted by a photodiode or a phototransistor located on the opposite side of the disk. By a suitable arrangement of the transparent and matte sectors, it is possible to convert a finite number of angular positions into corresponding digital data. The number of tracks determines the length of the word, and thus the resolution of the encoder.

To avoid problems of incorrect measurement in correspondence of a simultaneous multiple transition between matte and transparent sectors, it is worth utilizing a Gray-code encoder whose schematic representation is given in Fig. 5.12 with reference to the implementation of 4 tracks that allow the discrimination of 16 angular positions. It can be noticed that measurement ambiguity is eliminated, since only one change of contrast occurs at each transition (Table 5.1). For the typical resolution required for joint control, absolute encoders with a minimum number of 12 tracks (bits) are employed (resolution of 1/4096 per circle). Such encoders can provide unambiguous measurements only in a circle. If a gear reduction is present, a circle at the joint side corresponds to several circles at the motor side, and thus a simple electronics is needed to count and store the number of actual circles.

Table 5.1. Coding table with Gray-code

#	Code	#	Code
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

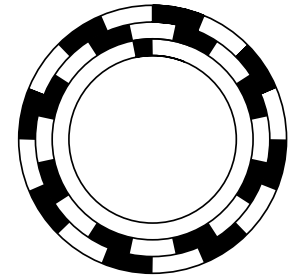

Fig. 5.13. Schematic representation of an incremental encoder

Incremental encoders have a wider use than absolute encoders, since they are simpler from a construction viewpoint and thus cheaper. Like the absolute one, the incremental encoder consists of an optical disk on which two tracks are disposed, whose transparent and matte sectors (in equal number on the two tracks) are mutually in quadrature. The presence of two tracks also allows, besides the number of transitions associated with any angular rotation, the detection of the sign of rotation. Often a third track is present with one single matte sector which allows the definition of an absolute mechanical zero as a reference for angular position. A schematic representation is illustrated in Fig. 5.13.

The use of an incremental encoder for a joint actuating system clearly demands the evaluation of absolute positions. This is performed by means of suitable counting and storing electronic circuits. To this end, it is worth noticing that the position information is available on volatile memories, and thus it can be corrupted due to the effect of disturbances acting on the electronic circuit, or else fluctuations in the supply voltage. Such limitation obviously does not occur for absolute encoders, since the angular position information is coded directly on the optical disk.

The optical encoder has its own signal processing electronics inside the case, which provides direct digital position measurements to be interfaced with the control computer. If an external circuitry is employed, velocity measurements can be reconstructed from position measurements. In fact, if a pulse is generated at each transition, a velocity measurement can be obtained in three possible ways, namely, by using a voltage-to-frequency converter (with analog output), by (digitally) measuring the frequency of the pulse train, or by (digitally) measuring the sampling time of the pulse train. Between these last two techniques, the former is suitable for high-speed measurements while the latter is suitable for low-speed measurements.

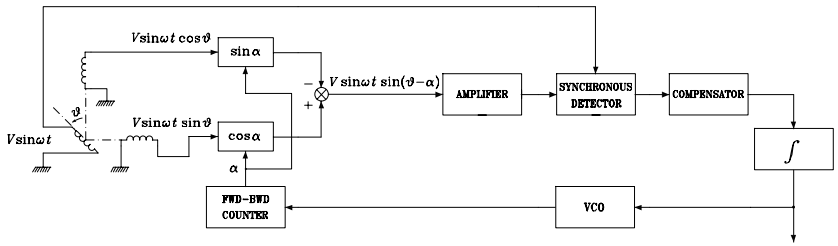

Fig. 5.14. Electric scheme of a resolver with functional diagram of a tracking-type RDC

Resolver

The resolver is an electromechanical position transducer which is compact and robust. Its operating principle is based on the mutual induction between two electric circuits which allow continuous transmission of angular position without mechanical limits. The information on the angular position is associated with the magnitude of two sinusoidal voltages, which are treated by a suitable resolver-to-digital converter (RDC) to obtain the digital data corresponding to the position measurement. The electric scheme of a resolver with the functional diagram of a tracking-type RDC is illustrated in Fig. 5.14.

From a construction viewpoint, the resolver is a small electric machine with a rotor and a stator; the inductance coil is on the rotor while the stator has two windings at 90 electrical degrees one from the other. By feeding the rotor with a sinusoidal voltage $V \sin \omega t$ (with typical frequencies in the range of 0.4 to 10 kHz), a voltage is induced on the stator windings whose magnitude depends on the rotation angle θ. The two voltages are fed to two digital multipliers, whose input is α and whose outputs are algebraically summed to achieve $V \sin \omega t \sin(\theta - \alpha)$; this signal is then amplified and sent to the input of a synchronous detector, whose filtered output is proportional to the quantity $\sin(\theta - \alpha)$. The resulting signal, after a suitable compensating action, is integrated and then sent to the input of a voltage-controlled oscillator (VCO) (a voltage-to-frequency converter) whose output pulses are input to a forward-backward counter. Digital data of the quantity α are available on the output register of the counter, which represent a measurement of the angle θ.

It can be recognized that the converter works according to a feedback principle. The presence of two integrators (one is represented by the forward-backward counter) in the loop ensures that the (digital) position and (analog) velocity measurements are error-free as long as the rotor rotates at constant speed; actually, a round-off error occurs on the word α and thus affects the position measurement. The compensating action is needed to confer suitable stability properties and bandwidth to the system. Whenever digital data are wished also for velocity measurements, it is necessary to use an analog-to-

digital converter. Since the resolver is a very precise transducer, a resolution of 1 bit out of 16 can be obtained at the output of the RDC.

5.3.2 Velocity Transducers

Even though velocity measurements can be reconstructed from position transducers, it is often preferred to resort to direct measurements of velocity, by means of suitable transducers. *Velocity transducers* are employed in a wide number of applications and are termed *tachometers*. The most common devices of this kind are based on the operating principles of electric machines. The two basic types of tachometers are the *direct-current (DC) tachometer* and the *alternating-current (AC) tachometer*.

DC tachometer

The direct-current tachometer is the most used transducer in the applications. It is a small DC generator whose magnetic field is provided by a permanent magnet. Special care is paid to its construction, so as to achieve a linear input/output relationship and to reduce the effects of magnetic hysteresis and temperature. Since the field flux is constant, when the rotor is set in rotation, its output voltage is proportional to angular speed according to the constant characteristic of the machine.

Because of the presence of a commutator, the output voltage has a residual ripple which cannot be eliminated by proper filtering, since its frequency depends on angular speed. A linearity range of 0.1 to 1% can be obtained, whereas the residual ripple coefficient is of 2 to 5% of the mean value of the output signal.

AC tachometer

In order to avoid the drawbacks caused by the presence of a residual ripple in the output of a DC tachometer, one may resort to an AC tachometer. While the DC tachometer is a true DC generator, the AC tachometer differs from a generator. In fact, if a synchronous generator would be used, the frequency of the output signal would be proportional to the angular speed.

To obtain an alternating voltage whose magnitude is proportional to speed, one may resort to an electric machine that is structurally different from the synchronous generator. The AC tachometer has two windings on the stator mutually in quadrature and a cup rotor. If one of the windings is fed by a constant-magnitude sinusoidal voltage, a sinusoidal voltage is induced on the other winding which has the same frequency, a magnitude proportional to angular speed, and a phase equal or opposite to that of the input voltage according to the sign of rotation; the exciting frequency is usually set to 400 Hz. The use of a synchronous detector then yields an analog measurement

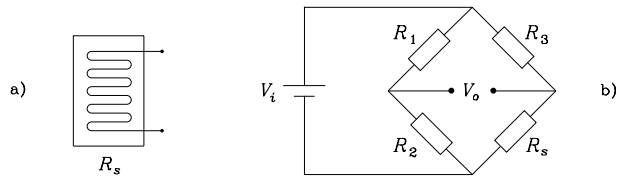

Fig. 5.15. a) Schematic representation of a strain gauge. **b)** Its insertion in a Wheatstone bridge

of angular velocity. In this case, the output ripple can be eliminated by a proper filter, since its fundamental frequency is twice as much as the supply frequency.

The performance of AC tachometers is comparable to that of DC tachometers. Two further advantages of AC tachometers are the lack of wiping contacts and the presence of a low moment of inertia, in view of the use of a lightweight cup rotor. However, a residual voltage occurs, even when the rotor is still, because of the unavoidable parasitic couplings between the stator coil and the measurement circuitry.

5.4 Exteroceptive Sensors

5.4.1 Force Sensors

Measurement of a force or torque is usually reduced to measurement of the strain induced by the force (torque) applied to an extensible element of suitable features. Therefore, an indirect measurement of force is obtained by means of measurements of small displacements. The basic component of a force sensor is the *strain gauge* which uses the change of electric resistance of a wire under strain.

Strain gauge

The strain gauge consists of a wire of low temperature coefficient. The wire is disposed on an insulated support (Fig. 5.15a) which is glued to the element subject to strain under the action of a stress. Dimensions of the wire change and then they cause a change of electric resistance.

The strain gauge is chosen in such a way that the resistance R_s changes linearly in the range of admissible strain for the extensible element. To transform changes of resistance into an electric signal, the strain gauge is inserted in one arm of a Wheatstone bridge which is balanced in the absence of stress on the strain gauge itself. From Fig. 5.15b it can be understood that the voltage balance in the bridge is described by

$$V_o = \left(\frac{R_2}{R_1 + R_2} - \frac{R_s}{R_3 + R_s} \right) V_i. \tag{5.30}$$

If temperature variations occur, the wire changes its dimension without application of any external stress. To reduce the effect of temperature variations on the measurement output, it is worth inserting another strain gauge in an adjacent arm of the bridge, which is glued on a portion of the extensible element not subject to strain.

Finally, to increase bridge sensitivity, two strain gauges may be used which have to be glued on the extensible element in such a way that one strain gauge is subject to traction and the other to compression; the two strain gauges then have to be inserted in two adjacent arms of the bridge.

Shaft torque sensor

In order to employ a servomotor as a torque-controlled generator, an indirect measurement of the driving torque is typically used, e.g., through the measurement of armature current in a permanent-magnet DC servomotor. If it is desired to guarantee insensitivity to change of parameters relating torque to the measured physical quantities, it is necessary to resort to a direct torque measurement.

The torque delivered by the servomotor to the joint can be measured by strain gauges mounted on an extensible apparatus interposed between the motor and the joint, e.g., a hollow shafting. Such apparatus must have low torsional stiffness and high bending stiffness, and it must ensure a proportional relationship between the applied torque and the induced strain.

By connecting the strain gauges mounted on the hollow shafting (in a Wheatstone bridge configuration) to a slip ring by means of graphite brushes, it is possible to feed the bridge and measure the resulting unbalanced signal which is proportional to the applied torque.

The measured torque is that delivered by the servomotor to the joint, and thus it does not coincide with the driving torque C_m in the block schemes of the actuating systems in Fig. 5.2 and in Fig. 5.6. In fact, such measurement does not account for the inertial and friction torque contributions as well as for the transmission located upstream of the measurement point.

Wrist force sensor

When the manipulator's end-effector is in contact with the working environment, the *force sensor* allows the measurement of the three components of a force and the three components of a moment with respect to a frame attached to it.

As illustrated in Fig. 5.16, the sensor is employed as a connecting apparatus at the wrist between the outer link of the manipulator and the end-effector. The connection is made by means of a suitable number of extensible elements subject to strain under the action of a force and a moment. Strain gauges are glued on each element which provide strain measurements. The elements

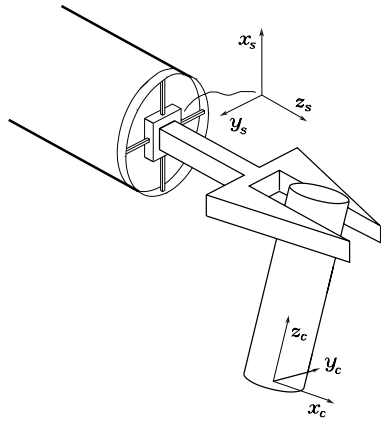

Fig. 5.16. Use of a force sensor on the outer link of a manipulator

have to be disposed in a keen way so that at least one element is appreciably deformed for any possible orientation of forces and moments.

Furthermore, the single force component with respect to the frame attached to the sensor should induce the least possible number of deformations, so as to obtain good structural decoupling of force components. Since a complete decoupling cannot be achieved, the number of significant deformations to reconstruct the six components of the force and moment vector is greater than six.

A typical force sensor is that where the extensible elements are disposed as in a Maltese cross; this is schematically indicated in Fig. 5.17. The elements connecting the outer link with the end-effector are four bars with a rectangular parallelepiped shape. On the opposite sides of each bar, a pair of strain gauges is glued that constitute two arms of a Wheatstone bridge; there is a total of eight bridges and thus the possibility of measuring eight strains.

The matrix relating strain measurements to the force components expressed in a Frame s attached to the sensor is termed sensor *calibration matrix*. Let w_i, for $i = 1, \ldots, 8$, denote the outputs of the eight bridges providing measurement of the strains induced by the applied forces on the bars according

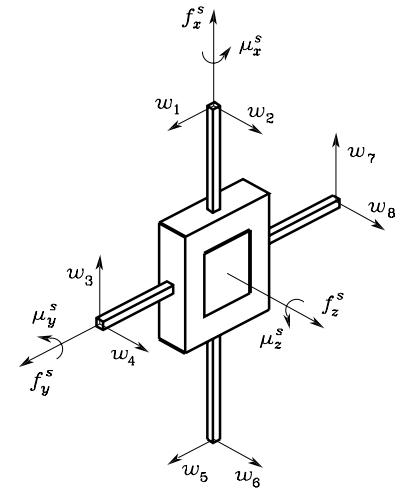

Fig. 5.17. Schematic representation of a Maltese-cross force sensor

to the directions specified in Fig. 5.17. Then, the calibration matrix is given by the transformation

$$\begin{bmatrix} f_x^s \\ f_y^s \\ f_z^s \\ \mu_x^s \\ \mu_y^s \\ \mu_z^s \end{bmatrix} = \begin{bmatrix} 0 & 0 & c_{13} & 0 & 0 & 0 & c_{17} & 0 \\ c_{21} & 0 & 0 & 0 & c_{25} & 0 & 0 & 0 \\ 0 & c_{32} & 0 & c_{34} & 0 & c_{36} & 0 & c_{38} \\ 0 & 0 & 0 & c_{44} & 0 & 0 & 0 & c_{48} \\ 0 & c_{52} & 0 & 0 & 0 & c_{56} & 0 & 0 \\ c_{61} & 0 & c_{63} & 0 & c_{65} & 0 & c_{67} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \end{bmatrix} . \tag{5.31}$$

Reconstruction of force measurements through the calibration matrix is entrusted to suitable signal processing circuitry available in the sensor.

Typical sensors have a diameter of about 10 cm and a height of about 5 cm, with a measurement range of 50 to 500 N for the forces and of 5 to 70 N·m for the torques, and a resolution of the order of 0.1% of the maximum force and of 0.05% of the maximum torque, respectively; the sampling frequency at the output of the processing circuitry is of the order of 1 kHz.

Finally, it is worth noticing that force sensor measurements cannot be directly used by a force/motion control algorithm, since they describe the equivalent forces acting on the sensors which differ from the forces applied to the manipulator's end-effector (Fig. 5.16). It is therefore necessary to trans-

form those forces from the sensor Frame s into the constraint Frame c; in view of the transformation in (3.116), one has

$$\begin{bmatrix} \boldsymbol{f}_c^c \\ \boldsymbol{\mu}_c^c \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_s^c & \boldsymbol{O} \\ \boldsymbol{S}(\boldsymbol{r}_{cs}^c)\boldsymbol{R}_s^c & \boldsymbol{R}_s^c \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_s^s \\ \boldsymbol{\mu}_s^s \end{bmatrix} \tag{5.32}$$

which requires knowledge of the position $\boldsymbol{r}_{cs}^c$ of the origin of Frame s with respect to Frame c as well as of the orientation $\boldsymbol{R}_s^c$ of Frame s with respect to Frame c. Both such quantities are expressed in Frame c, and thus they are constant only if the end-effector is still, once contact has been achieved.

5.4.2 Range Sensors

The primary function of the exteroceptive sensors is to provide the robot with the information needed to execute 'intelligent' actions in an autonomous way. To this end, it is crucial to detect the presence of an object in the workspace and eventually to measure its range from the robot along a given direction.

The former kind of data is provided by the *proximity sensors*, a simplified type of *range sensors*, capable of detecting only the presence of objects nearby the sensitive part of the sensor, without a physical contact. The distance within which such sensors detect objects is defined *sensitive range*.

In the more general case, range sensors are capable of providing structured data, given by the distance of the measured object and the corresponding measurement direction, i.e., the position in space of the detected object with respect to the sensor.

The data provided by the range sensors are used in robotics to avoid obstacles, build maps of the environment, recognize objects.

The most popular range sensors in robotics applications are those based on sound propagation through an elastic fluid, the so-called *sonars* (SOund NAvigation and Ranging), and those exploiting light propagation features, the so-called *lasers* (Light Amplification by Stimulated Emission of Radiation). In the following, the main features of these two sensors are illustrated.

Sonars

The sonars employ acoustic pulses and their echoes to measure the range to an object. Since the sound speed is usually known for a given media (air, water), the range to an object is proportional to the echo travel time, commonly called *time-of-flight*, i.e., the time which the acoustic wave takes to cover the distance sensor-object-sensor. Sonars are widely utilized in robotics, and especially in mobile and underwater robotics. Their popularity is due to their low cost, light weight, low power consumption, and low computational effort, compared to other ranging sensors. In some applications, such as in underwater and low-visibility environments, the sonar is often the only viable sensing modality.

Despite a few rare examples of sonars operating at audible frequencies for human ears (about 20 Hz to 20 KHz), the ultrasound frequencies (higher

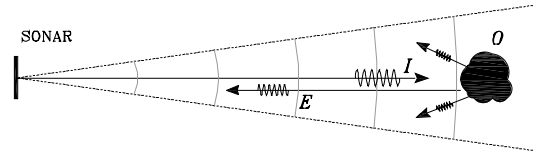

Fig. 5.18. Sonar ranging principle

than 20 KHz) are the most widely used to realize this type of sensor. Typical frequencies in robotics range from 20 KHz to 200 KHz, even though higher values (of the order of MHz) can be achieved utilizing piezoelectric quartz crystals. In this range, the energy of the wave emitted by the sonar can be regarded as concentrated in a conical volume whose beamwidth depends on the frequency as well as on the transducer diameter. Further to measuring range, sonars provide qualitative directional data on the object which has generated the echo. For the most common sensors in robotics, the beamwidth of the energy beam is typically not smaller than 15 deg. Obviously, for smaller beamwidths, higher angular resolutions can be obtained.

The main components of a sonar measurement system are a transducer, which is vibrated and transforms acoustic energy into electric energy and vice versa, and a circuitry for the excitation of the transducer and the detection of the reflected signal. Figure 5.18 schematically illustrates the operating principle: the pulse I emitted by the transducer, after hitting the object O found in the emission cone of the sensor, is partly reflected (echo E) towards the sound source and thus detected. The time-of-flight t_v is the time between the emission of the ultrasound pulse and the reception of the echo. The object range d_O can be computed from t_v using the relation

$$d_O = \frac{c_s t_v}{2} \tag{5.33}$$

where c_s is sound speed, which in low-humidity air depends on the temperature T (measured in centigrade) according to the expression

$$c_s \approx 20.05\sqrt{T + 273.16} \text{ m/s}. \tag{5.34}$$

In the scheme of Fig. 5.18 the use of a sole transducer is represented for the transmission of the pulse and the reception of the echo. This configuration requires that the commutation from transmitter to receiver takes place after a certain latency time which depends not only on the duration of the transmitted pulse but also on the mechanical inertia of the transducer.

Despite the low cost and ease of use, however, these sensors have non-negligible limits with respect to the angular and radial resolution, as well as to the minimum and maximum measurement range that can be achieved. In particular, the width of the radiation cone decreases as frequency increases with improved angular resolution. A higher frequency leads to greater radial

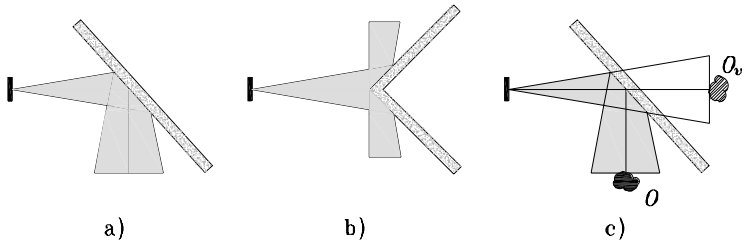

Fig. 5.19. Reflector models on smooth surfaces: **a)** non-detected plane. **b)** non-detected corner. **c)** plane with false detection (O real object, O_v virtual object detected)

resolution and contributes to reducing the minimum range that can be detected by the sonar. Nevertheless, there is a lower limit because of the lapse time when reception is inhibited to avoid interference with the reflected signal — in certain cases better performance can be obtained by employing two distinct transducers for the emission and the detection. On the other hand, too high frequencies may exasperate absorbtion phenomena, depending on the features of the surface generating the echo. Such phenomena further reduce the power of the transmitted signal — decreasing with the square of the range covered by the ultrasound wave — thus reducing the maximum limit of the measurement time.

Piezoelectric and electrostatic transducers are the two major types available that operate in air and can in principle operate both as a transmitter and receiver.

The *piezoelectric transducers* exploit the property of some crystal materials to deform under the action of an electric field and vibrate when a voltage is applied at the resonant frequency of the crystal. The efficiency of the acoustic match of these transducers with compressible fluids such as air is rather low. Often a conical concave horn is mounted on the crystal to match acoustically the crystal acoustic impedance to that of air. Being of resonant type, these transducers are characterized by a rather low bandwidth and show a significant mechanical inertia which severely limits the minimum detectable range, thus justifying the use of two distinct transducers as transmitter and receiver.

The *electrostatic transducers* operate as capacitors whose capacitance varies moving and/or deforming one of its plates. A typical construction consists of a gold-coated plastic foil membrane (*mobile* plate) stretched across a round grooved aluminium back plate (*fixed* plate). When the transducer operates as receiver, the change of capacitance, induced by the deformation of the membrane under the acoustic pressure, produces a proportional change of the voltage across the capacitor, assuming that the foil charge is constant. As a transmitter, the transducer membrane is vibrated by applying a sequence of electric pulses across the capacitor. The electric oscillations generate, as

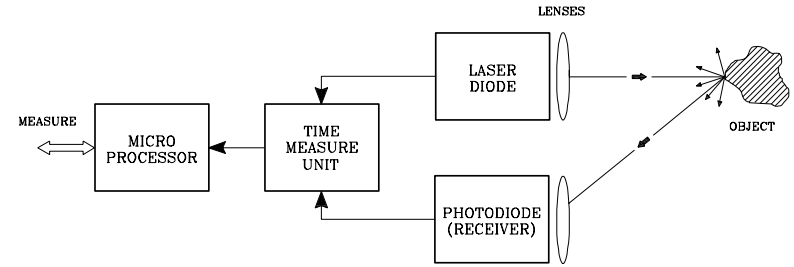

Fig. 5.20. Time-of-flight laser sensor operating principle

a result of the induced electric field, a mechanical force which vibrates the mobile plate.

Since the electrostatic transducers can operate at different frequencies, they are characterized by large bandwidth and high sensitivity, low mechanical inertia and rather efficient acoustic match with air. As compared to the piezoelectric transducers, however, they can operate at lower maximum frequencies (a few hundreds kHz vs a few MHz) and require a bias voltage which complicates the control electronics. Among the ultrasound measurement systems with capacitive transducers, it is worth mentioning the Polaroid sonar, initially developed for autofocus systems and later widely employed as range sensors in several robotic applications. The 600 series sensor utilizes a capacitive transducer of the type described above with a diameter of almost 4 cm, operates at 50 kHz frequency and is characterized by a beamwidth of 15 deg, can detect a maximum range of about 10 m and a mimimum range of about 15 cm with an accuracy of $\pm 1\%$ across the measurement range. The bias voltage is 200 V with current absorbtion peaks of 2 A in transmission.

Accuracy of ultrasound range sensors depends on the features of the transducer and the excitation/detection circuitry, as well as on the reflective properties of the surfaces hit by the acoustic waves.

Smooth surfaces, i.e., those characterized by irregularities of comparable size to that of the wavelength corresponding to the employed frequency, may produce a non-detectable echo at the sensor (Figura 5.19a,b) if the incident angle of the ultrasound beam exceeds a given critical angle which depends on the operational frequency and the reflective material. In the case of the Polaroid sensors, this angle is equal to 65 deg, i.e., 25 deg from the normal to the reflective surface, for a smooth surface in plywood. When operating in complex environments, such mirror reflections may give rise to multiple reflections, thus causing range measurement errors or false detection (Fig. 5.19c).

Lasers

In the construction of optical measurement systems, the laser beam is usually preferred to other light sources for the following reasons:

- They can easily generate bright beams with lightweight sources.
- The infrared beams can be used unobtrusively.
- They focus well to give narrow beams.
- Single-frequency sources allow easier rejection filtering of unwanted frequencies, and do not disperse from refraction as much as full spectrum sources.

There are two types of laser-based range sensors in common use: the time-of-flight sensors and the triangulation sensors.

The *time-of-flight sensors* compute distance by measuring the time that a pulse of light takes to travel from the source to the observed target and then to the detector (usually collocated with the source). The travel time multiplied by the speed of light (properly adjusted for the air temperature) gives the distance measurement. The operating principle of a time-of-flight laser sensor is illustrated in Fig. 5.20.

Limitations on the accuracy of these sensors are based on the minimum observation time — and thus the minimum distance observable, the temporal accuracy (or quantization) of the receiver, and the temporal width of the laser pulse. Such limitations are not only of a technological nature. In many cases, cost is the limiting factor of these measurement devices. For instance, to obtain 1 mm resolution, a time accuracy of about 3 ps, which can be achieved only by using rather expensive technology.

Many time-of-flight sensors used have what is called an *ambiguity interval*. The sensor emits pulses of light periodically, and computes an average target distance from the time of the returning pulses. Typically, to simplify the detection electronics of these sensors, the receiver only accepts signals that arrive within time Δt, but this time window might also observe previous pulses reflected by more distant surfaces. This means that a measurement is ambiguous to the multiple of $\frac{1}{2}c\Delta t$, where c is the speed of light. Typical values of $\frac{1}{2}c\Delta t$ are 20–40 m.

In certain conditions, suitable algorithms can be employed to recover the true depth by assuming that the distances should be changing smoothly.

The time-of-flight sensors transmit only a single beam, thus range measurements are only obtained from a single surface point. In order to obtain more information, the range data is usually supplied as a vector of range to surfaces lying in a plane or as an image. To obtain these denser representations, the laser beam is swept across the scene. Normally the beam is swept by a set of mirrors rather than moving the laser and detector themselves — mirrors are lighter and less prone to motion damage.

Typical time-of-flight sensors suitable for mobile robotics applications have a range of 5–100 m, an accuracy of 5–10 mm, and a frequency of data acquisition per second of 1000–25000 Hz.

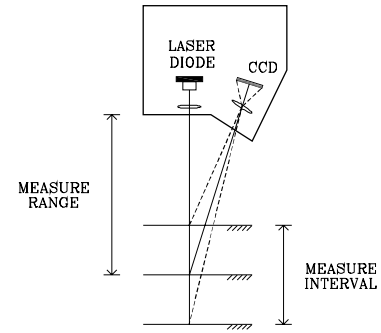

Fig. 5.21. Triangulation laser sensor operating principle

The operating principle of *triangulation laser sensors*[3] is illustrated in Fig. 5.21.

The laser beam emitted by a photodiode is projected onto the observed surface. The reflected beam is focused on a CCD sensor by means of a suitable lens. Obviously, reflection must be diffused. The position of the focused beam reflected to the receiver gives rise to a signal which is proportional to the distance of the transmitter from the object. In fact, from the measurement of the CCD sensor it is possible to resort to the angle at which the reflected energy hits the sensor. Once the relative position and orientation of the CCD sensor with respect to the photodiode are known, as e.g. through a suitable calibration procedure, it is possible to compute the distance from the object with simple geometry.

Accuracy can be influenced by certain object surfaces not favouring reflection, differences or changes of colour. Such occurrences can be mitigated or even eliminated with modern electronic technology and automatic regulation of light intensity.

The possibility of controlling the laser beam light brings the following advantages:

[3] The triangulation method is based on the trigonometric properties of triangles and in particular on the cosine theorem. The method allows the computation of the distance between two non-directly accessible points, i.e., once two angles and one side of a triangle are known, it is possible to determine the other two sides. For the case at issue, one side is given by the distance between the emitter (laser) and the receiver (the CCD sensor), one angle is given by the orientation of the emitter with respect to that side and the other angle can be computed from the position of the laser beam on the image plane. In practice, it is not easy to compute the above quantities, and suitable calibration techniques are to be employed which avoid such computation to determine the distance measurement.

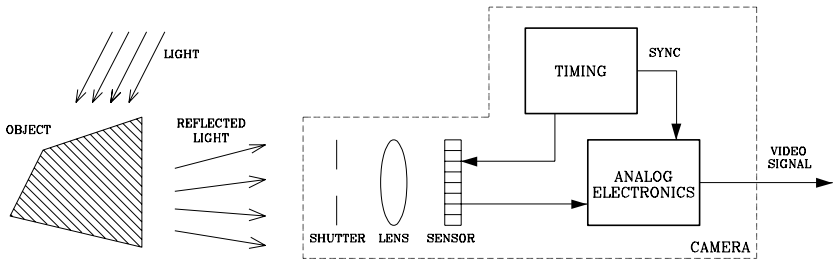

Fig. 5.22. Schematic representation of a vision system

- If the laser beam wavelength is known, e.g. that of the visible red 670 nm, highly selective filters can be used which are set to the same frequency to reduce the effects of other light sources.
- The laser beam may be remodelled through lenses and mirrors so as to create multiple beams or laser strips to measure multiple 3D points simultaneously.
- The direction of the laser beam can be controlled directly by the control system to observe selectively only those portions of the scene of interest.

The main limitations of this type of sensors are the potential eye safety risks from the power of lasers, particularly when invisible laser frequencies are used (commonly infrared), as well as the false specular reflections from metallic and polished objects.

5.4.3 Vision Sensors

The task of a camera as a *vision sensor* is to measure the intensity of the light reflected by an object. To this end, a photosensitive element, termed *pixel* (or *photosite*), is employed, which is capable of transforming light energy into electric energy. Different types of sensors are available depending on the physical principle exploited to realize the energy transformation. The most widely used devices are CCD and CMOS sensors based on the photoelectric effect of semiconductors.

CCD

A CCD (Charge Coupled Device) sensor consists of a rectangular array of photosites. Due to the photoelectric effect, when a photon hits the semiconductor surface, a number of free electrons are created, so that each element accumulates a charge depending on the time integral of the incident illumination over the photosensitive element. This charge is then passed by a transport mechanism (similar to an analog shift register) to the output amplifier, while at

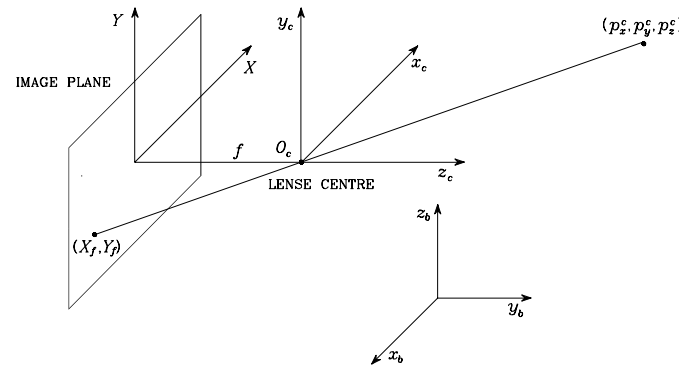

Fig. 5.23. Perspective transformation

the same time the photosite is discharged. The electric signal is to be further processed in order to produce the real *video signal*.

CMOS

A CMOS (Complementary Metal Oxide Semiconductor) sensor consists of a rectangular array of photodiodes. The junction of each photodiode is precharged and it is discharged when hit by photons. An amplifier integrated in each pixel can transform this charge into a voltage or current level. The main difference with the CCD sensor is that the pixels of a CMOS sensor are non-integrating devices; after being activated they measure throughput, not volume. In this manner, a saturated pixel will never overflow and influence a neighboring pixel. This prevents the effect of *blooming*, which indeed affects CCD sensors.

Camera

As sketched in Fig. 5.22, a camera is a complex system comprising several devices other than the photosensitive sensor, i.e., a *shutter*, a *lens* and *analog preprocessing electronics*. The lens is responsible for focusing the light reflected by the object on the plane where the photosensitive sensor lies, called the *image plane*.

With reference to Fig. 5.23, consider a frame O_c–$x_c y_c z_c$ attached to the camera, whose location with respect to the base frame is identified by the homogeneous transformation matrix $\boldsymbol{T}_c^b$. Take a point of the object of coordinates $\boldsymbol{p}^c = [\, p_x^c \quad p_y^c \quad p_z^c \,]^T$; typically, the centroid of the object is chosen. Then, the coordinate transformation from the base frame to the camera frame is described as

$$\widetilde{\boldsymbol{p}}^c = \boldsymbol{T}_b^c \widetilde{\boldsymbol{p}}, \tag{5.35}$$

where $\boldsymbol{p}$ denotes the object position with respect to the base frame and homogeneous representations of vectors have been used.

A reference frame can be introduced on the image plane, whose axes X and Y are parallel to the axes x_c and y_c of the camera frame, and the origin is at the intersection of the optical axis with the image plane, termed principal point. Due to the refraction phenomenon, the point in the camera frame is transformed into a point in the image plane via the *perspective transformation*, i.e.,

$$X_f = -\frac{f p_x^c}{p_z^c}$$
$$Y_f = -\frac{f p_y^c}{p_z^c}$$

where (X_f, Y_f) are the new coordinates in the frame defined on the image plane, and f is the *focal length* of the lens. Notice that these coordinates are expressed in metric units and the above transformation is singular at $p_z^c = 0$.

The presence of the minus sign in the equations of the perspective transformation is consistent with the fact that the image of an object appears upside down on the image plane of the camera. Such an effect can be avoided, for computational ease, by considering a virtual image plane positioned before the lens, in correspondence of the plane $z_c = f$ of the camera frame. In this way, the model represented in Fig. 5.24 is obtained, which is characterized by the *frontal* perspective transformation

$$X_f = \frac{f p_x^c}{p_z^c} \tag{5.36}$$
$$Y_f = \frac{f p_y^c}{p_z^c} \tag{5.37}$$

where, with abuse of notation, the name of the variables on the virtual plane has not been changed.

These relationships hold only in theory, since the real lenses are always affected by imperfections, which cause image quality degradation. Two types of distortions can be recognized, namely, *aberrations* and *geometric distortion*. The former can be reduced by restricting the light rays to a small central region of the lens; the effects of the latter can be compensated on the basis of a suitable model whose parameters are to be identified.

A visual information is typically elaborated by a digital processor, and thus the measurement principle is to transform the light intensity $I(X, Y)$ of each point in the image plane into a number. It is clear that a *spatial sampling* is needed since an infinite number of points in the image plane exist, as well as a *temporal sampling* since the image can change during time. The CCD or CMOS sensors play the role of spatial samplers, while the shutter in front of the lens plays the role of the temporal sampler.

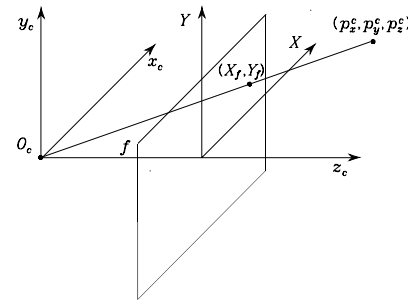

Fig. 5.24. Frontal perspective transformation

The spatial sampling unit is the pixel, and thus the coordinates (X, Y) of a point in the image plane are to be expressed in pixels, i.e., (X_I, Y_I). Due to the photosite finite dimensions, the pixel coordinates of the point are related to the coordinates in metric units through two scale factors α_x and α_y, namely,

$$X_I = \frac{\alpha_x f p_x^c}{p_z^c} + X_0 \tag{5.38}$$
$$Y_I = \frac{\alpha_y f p_y^c}{p_z^c} + Y_0, \tag{5.39}$$

where X_0 and Y_0 are the offsets which take into account the position of the origin of the pixel coordinate system with respect to the optical axis. This nonlinear transformation can be written in a linear form by resorting to the homogeneous representation of the point (x_I, y_I, z_I) via the relationships

$$X_I = \frac{x_I}{\lambda}$$
$$Y_I = \frac{y_I}{\lambda}$$

where $\lambda > 0$. As a consequence, (5.38), (5.39) can be rewritten as

$$\begin{bmatrix} x_I \\ y_I \\ \lambda \end{bmatrix} = \lambda \begin{bmatrix} X_I \\ Y_I \\ 1 \end{bmatrix} = \boldsymbol{\Omega\Pi} \begin{bmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{bmatrix} \tag{5.40}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} f\alpha_x & 0 & X_0 \\ 0 & f\alpha_y & Y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.41}$$

$$\boldsymbol{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.42}$$

At this point, the overall transformation from the Cartesian space of the observed object to the *image space* of its image in pixels is characterized by composing the transformations in (5.35), (5.40) as

$$\Xi = \boldsymbol{\Omega \Pi T}_b^c \tag{5.43}$$

which represents the so-called *camera calibration* matrix. It is worth pointing out that such a matrix contains *intrinsic parameters* $(\alpha_x, \alpha_y, X_0, Y_0, f)$ in $\boldsymbol{\Omega}$ depending on the sensor and lens characteristics as well as *extrinsic parameters* in $\boldsymbol{T}_c^b$ depending on the relative position and orientation of the camera with respect to the base frame. Several calibration techniques exist to identify these parameters in order to compute the transformation between the Cartesian space and the image space as accurately as possible.

If the intrinsic parameters of a camera are known, from a computationally viewpoint, it is convenient to refer to the *normalized coordinates* (X, Y), defined by the normalized perspective transformation

$$\lambda \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \boldsymbol{\Pi} \begin{bmatrix} p_x^c \\ p_y^c \\ p_z^c \\ 1 \end{bmatrix}. \tag{5.44}$$

These coordinates are defined in metrical units and coincide with the coordinates (5.36), (5.37) in the case when $f = 1$. Comparing (5.40) with (5.44) yields the invertible transformation

$$\begin{bmatrix} X_I \\ Y_I \\ 1 \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{5.45}$$

relating the normalized coordinates to those expressed in pixels through the matrix of intrinsic parameters.

If a monochrome CCD camera[4] is of concern, the output amplifier of the sensor produces a signal which is processed by a timing analog electronics in order to generate an electric signal according to one of the existing *video standards*, i.e., the CCIR European and Australian standard, or the RS170 American and Japanese standard. In any case, the video signal is a voltage of 1 V peak-to-peak whose amplitude represents sequentially the image intensity.

The entire image is divided into a number of lines (625 for the CCIR standard and 525 for the RS170 standard) to be sequentially scanned. The raster scan proceeds horizontally across each line and each line from top to bottom, but first all the even lines, forming the first *field*, and then all the odd lines, forming the second *field*, so that a *frame* is composed of two successive

[4] Colour cameras are equipped with special CCDs sensitive to three basic colours (RGB); the most sophisticated cameras have three separate sensors, one per each basic colour.

fields. This technique, called *interlacing*, allows the image to be updated either at frame rate or at field rate; in the former case the update frequency is that of the entire frame (25 Hz for the CCIR standard and 30 Hz for the RS170 standard), while in the latter case the update frequency can be doubled as long as half the vertical resolution can be tolerated.

The last step of the measurement process is to digitize the analog video signal. The special analog-to-digital converters adopted for video signal acquisition are called *frame grabbers*. By connecting the output of the camera to the frame grabber, the video waveform is sampled and quantized and the values stored in a two-dimensional memory array representing the spatial sample of the image, known as *framestore*; this array is then updated at field or frame rate.

In the case of CMOS cameras (currently available only for monochrome images), thanks to CMOS technology which allows the integration of the analog-to-digital converter in each pixel, the output of the camera is directly a two-dimensional array, whose elements can be accessed randomly. Such advantage, with respect to CCD cameras, leads to the possibility of higher frame rates if only parts of the entire frame are accessed.

The sequence of steps from image formation to image acquisition described above can be classified as a process of *low-level vision*; this includes the extraction of elementary image features, e.g., centroid and intensity discontinuities. On the other hand, a robotic system can be considered really autonomous only if procedures for emulating cognition are available, e.g., recognizing an observed object among a set of CAD models stored into a data base. In this case, the artificial vision process can be referred to as *high-level vision*.

Bibliography

Scientific literature on actuating systems and sensors is wide and continuously updated. The mechanical aspects on the joint actuating systems can be probed further in e.g. [186]. Details about electric servomotors can be found in [22], while in [156] construction and control problems for hydraulic motors are extensively treated. Control of electric drives is discussed in [128]; for direct drives see [12]. Joint control problems are discussed in [89].

A wide and detailed survey on sensors and in particular on proprioceptive sensors is given in [81]. In [220] force sensors are accurately described, with special attention to wrist force sensors. Further details about range sensors, with reference to mobile robotics applications, are available in [210]. Finally, a general introduction on vision sensors is contained in [48], while in [233] one of the most common calibration techniques for vision systems is described.

Problems

5.1. Prove (5.7)–(5.10).

5.2. Consider the DC servomotor with the data: $I_m = 0.0014\,\text{kg·m}^2$, $F_m = 0.01\,\text{N·m·s/rad}$, $L_a = 2\,\text{mH}$, $R_a = 0.2\,\text{ohm}$, $k_t = 0.2\,\text{N·m/A}$, $k_v = 0.2\,\text{V·s/rad}$, $C_i G_v = 1$, $T_v = 0.1\,\text{ms}$, $k_i = 0$. Perform a computer simulation of the current and velocity response to a unit step voltage input V_c'. Adopt a sampling time of 1 ms.

5.3. For the servomotor of the previous problem, design the controller of the current loop $C_i(s)$ so that the current response to a unit step voltage input V_c' is characterized by a settling time of 2 ms. Compare the velocity response with that obtained in Problem 5.2.

5.4. Find the control voltage/output position and reaction torque/output position transfer functions for the scheme of Fig. 5.6.

5.5. For a Gray-code optical encoder, find the interconversion logic circuit which yields a binary-coded output word.

5.6. With reference to a contact situation of the kind illustrated in Fig. 5.16, let

$$\boldsymbol{r}_{cs}^{c} = [\,-0.3 \quad 0 \quad 0.2\,]^T\ \text{m} \qquad \boldsymbol{R}_{s}^{c} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and let the force sensor measurement be

$$\boldsymbol{f}_{s}^{s} = [\,20 \quad 0 \quad 0\,]^T\ \text{N} \qquad \boldsymbol{\mu}_{s}^{s} = [\,0 \quad 6 \quad 0\,]^T\ \text{N·m}.$$

Compute the equivalent force and moment in the contact frame.

5.7. Consider the SCARA manipulator in Fig. 2.34 with link lengths $a_1 = a_2 = 0.5\,\text{m}$. Let the base frame be located at the intersection between the first link and the base link with axis z pointing downward and axis x in the direction of the first link when $\vartheta_1 = 0$. Assume that a CCD camera is mounted on the wrist so that the camera frame is aligned with the end-effector frame. The camera parameters are $f = 8\,\text{mm}$, $\alpha_x = 79.2\,\text{pixel/mm}$, $\alpha_y = 120.5\,\text{pixel/mm}$, $X_0 = 250$, $Y_0 = 250$. An object is observed by the camera and is described by the point of coordinates $\boldsymbol{p} = [\,0.8 \quad 0.5 \quad 0.9\,]^T\,\text{m}$. Compute the pixel coordinates of the point when the manipulator is at the configuration $\boldsymbol{q} = [\,0 \quad \pi/4 \quad 0.1 \quad 0\,]^T$.

6

Control Architecture

This chapter is devoted to presenting a reference model for the *functional architecture* of an industrial robot's *control system*. The *hierarchical structure* and its articulation into *functional modules* allows the determination of the requirements and characteristics of the *programming environment* and the *hardware architecture*. The architecture refers to robot manipulators, yet its articulation in *levels* also holds for mobile robots.

6.1 Functional Architecture

The *control system* to supervise the activities of a robotic system should be endowed with a number of tools providing the following functions:

- capability of moving physical objects in the working environment, i.e., *manipulation* ability;
- capability of obtaining information on the state of the system and working environment, i.e., *sensory* ability;
- capability of exploiting information to modify system behaviour in a pre-programmed manner, i.e., *intelligence* ability;
- capability of storing, elaborating and providing data on system activity, i.e., *data processing* ability.

An effective implementation of these functions can be obtained by means of a *functional architecture* which is thought of as the superposition of several *activity levels* arranged in a *hierarchical structure*. The lower levels of the structure are oriented to physical motion execution, whereas the higher levels are oriented to logical action planning. The levels are connected by data flows; those directed towards the higher levels regard measurements and/or results of actions, while those directed towards the lower levels regard transmission of directions.

With reference to the control system functions implementing management of the above listed system activities, in general it is worth allocating three

functional models at each level. A first module is devoted to sensory data management (sensory module). A second module is devoted to provide knowledge of the relevant world (modelling module). A third module is devoted to decide the policy of the action (decision module).

More specifically, the *sensory modules* acquire, elaborate, correlate and integrate sensory data in time and space, in order to recognize and measure the system state and environment characteristic; clearly, the functions of each module are oriented to the management of the relevant sensory data for that level.

On the other hand, the *modelling modules* contain models derived on the basis of a priori knowledge of system and environment; these models are updated by the information coming from the sensory modules, while the activation of the required functions is entrusted to the decision modules.

Finally, the *decision modules* perform breakdown of high-level tasks into low-level actions; such task breakdown concerns both breakdown in time of sequential actions and breakdown in space of concurrent actions. Each decision module is entrusted with the functions concerning management of elementary action assignments, task planning and execution.

The functions of a decision module characterize the level of the hierarchy and determine the functions required to the modelling and sensory modules operating at the same level. This implies that the contents of these two modules do not uniquely allow the determination of the hierarchical level, since the same function may be present at more levels depending on the needs of the decision modules at the relative levels.

The functional architecture needs an *operator interface* at each level of the hierarchy, so as to allow an operator to perform supervisory and intervention functions on the robotic system.

The instructions imparted to the decision module at a certain level may be provided either by the decision module at the next higher level or by the operator interface, or else by a combination of the two. Moreover, the operator, by means of suitable communication tools, can be informed on the system state and thus can contribute his/her own knowledge and decisions to the modelling and sensory modules.

In view of the high data flow concerning the exchange of information between the various levels and modules of the functional architecture, it is worth allocating a shared *global memory* which contains the updated estimates on the state of the whole system and environment.

The structure of the reference model for the functional architecture is represented in Fig. 6.1, where the *four hierarchical levels* potentially relevant for robotic systems in industrial applications are illustrated. Such levels regard definition of the *task*, its breakdown into elementary *actions*, assignment of *primitives* to the actions, and implementation of control actions on the *servo*-manipulator. In the following, the general functions of the three modules at each level are described.

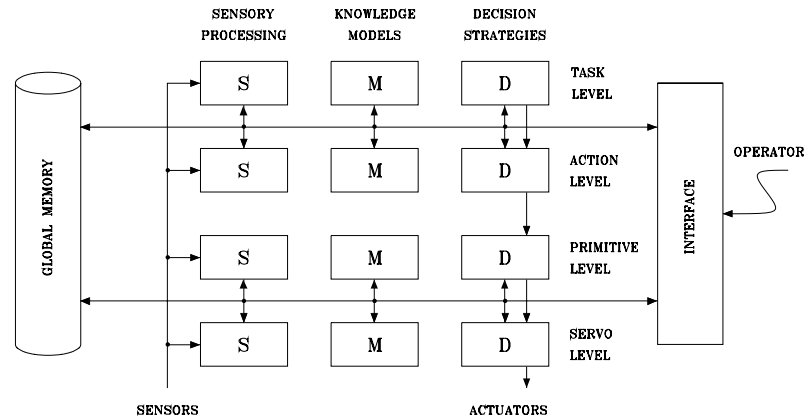

Fig. 6.1. Reference model for a control system functional architecture

At the *task level*, the user specifies the task which the robotic system should execute; this specification is performed at a high level of abstraction. The goal of the desired task is analyzed and broken down into a sequence of actions which are coordinated in space and time and allow implementation of the task. The choice of actions is performed on the basis of knowledge models as well as of the scene of interest for the task. For instance, consider the application of a robot installed in an assembly line which is required to perform a specific assembly task. To define the elementary actions that have to be transmitted to the decision module at the next lower level, the decision module should consult its knowledge base available in the modelling module, e.g., type of assembly, components of the object to assembly, assembly sequence, and choice of tools. This knowledge base should be continuously updated by the information provided by the sensory module concerning location of the parts to assembly; such information is available by means of a high-level vision system operating in a scarcely structured environment, or else by means of simple sensors detecting the presence of an object in a structured environment.

At the *action level*, the symbolic commands coming from the task level are translated into sequences of intermediate configurations which characterize a motion path for each elementary action. The choice of the sequences is performed on the basis of models of the manipulator and environment where the action is to take place. With reference to one of the actions generated by the above assembly task, the decision module chooses the most appropriate coordinate system to compute manipulator's end-effector poses, by separating translation from rotation if needed; it decides whether to operate in the joint or operational space, it computes the path or via points, and for the latter it defines the interpolation functions. By doing so, the decision module should compare the sequence of configurations with a model of the manipulator as

well as with a geometric description of the environment, which are both available in the modelling model. In this way, action feasibility is ascertained in terms of obstacle-collision avoidance, motion in the neighbourhood of kinematically singular configurations, occurrence of mechanical joint limits, and eventually utilization of available redundant DOFs. The knowledge base is updated by the information on the portion of scene where the single action takes place which is provided by the sensory module, e.g., by means of a low-level vision system or range sensors.

At the *primitive level*, on the basis of the sequence of configurations received by the action level, admissible motion trajectories are computed and the control strategy is decided. The motion trajectory is interpolated so as to generate the references for the servo level. The choice of motion and control primitives is conditioned by the features of the mechanical structure and its degree of interaction with the environment. Still with reference to the above case study, the decision module computes the geometric path and the relative trajectory on the basis of the knowledge of the manipulator dynamic model available in the modelling module. Moreover, it defines the type of control algorithm, e.g., decentralized control, centralized control, or interaction control; it specifies the relative gains; and it performs proper coordinate transformations, e.g., kinematic inversion if needed. The sensory module provides information on the occurrence of conflicts between motion planning and execution, by means of, e.g., force sensors, low-level vision systems and proximity sensors.

At the *servo level*, on the basis of the motion trajectories and control strategies imparted by the primitive level, control algorithms are implemented which provide the driving signals to the joint servomotors. The control algorithm operates on error signals between the reference and the actual values of the controlled quantities, by utilizing knowledge of manipulator dynamic model, and of kinematics if needed. In particular, the decision module performs a microinterpolation on the reference trajectory to exploit fully the dynamic characteristic of the drives; it computes the control law, and it generates the (voltage or current) signals for controlling the specific drives. The modelling module elaborates the terms of the control law depending on the manipulator current configuration and pass them to the decision module; such terms are computed on the basis of knowledge of manipulator dynamic model. Finally, the sensory module provides measurements of the proprioceptive sensors (position, velocity and contact force if needed); these measurements are used by the decision module to compute the servo errors and, if required, by the modelling module to update the configuration-dependent terms in the model.

The specification of the functions associated with each level points out that the implementation of such functions should be performed at different time rates, in view of their complexity and requirements. On one hand, the functions associated with the higher levels are not subject to demanding real-time constraints, since they regard planning activities. On the other hand, their

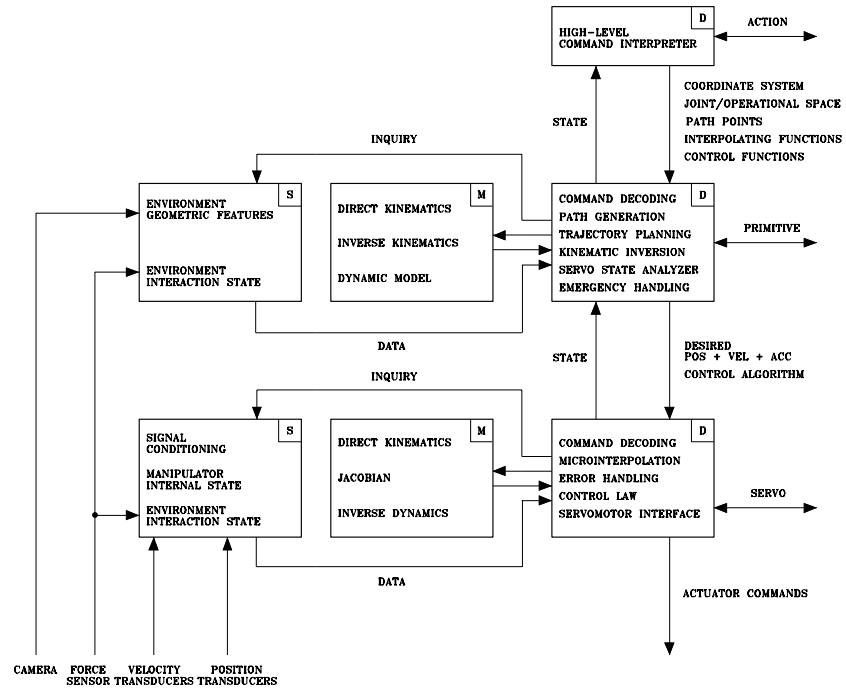

Fig. 6.2. Hierarchical levels of a functional architecture for industrial robots

complexity is notable, since scheduling, optimization, resource management and high-level sensory system data processing are required to update complex models.

At the lowest level, demanding real-time operation prevails in order to obtain high dynamic performance of the mechanical structure. The above remarks lead to the conclusion that, at the servo level, it is necessary to provide the driving commands to the motors and to detect the proprioceptive sensory data at sampling rates of the order of the millisecond, while sampling rates of the order of the minute are admissible at the task level.

With respect to this reference model of functional architecture, current industrial robot's control systems are not endowed with all the functions illustrated, because of both technology and cost limitations. In this regard, the task level is not implemented at all since there do not yet exist effective and reliable application software packages allowing support of the complex functions required at this level.

It is worth characterizing those functional levels of the reference models which are typically implemented in *advanced industrial robot's control systems*. The details of Fig. 6.2 show what follows:

- The modelling and sensory modules are always present at the lowest level, because of demanding requirements at the servo level for high dynamic performance robots to be employed even in relatively simple applications.
- At the primitive level, the modelling module is usually present while the sensory module is present only in a reduced number of applications that require robot interaction with a less structured environment.
- At the action level, the decision module is present only as an interpreter of the high-level commands imparted by the operator. All the task breakdown functions are entrusted to the operator, and thus the modelling and sensory module are absent at this level. Possible checking of action feasibility is moved down to the primitive level where a modelling module exists.

In view of the highly-structured reference model of functional architecture illustrated above, evolution of the control system towards more and more powerful capabilities is possible. In fact, one may foresee that information technology progress may allow the addition of hierarchically higher levels than the task level. These should functionally characterize complex tasks to be broken down into elementary tasks and yet, at an even higher level, missions to be broken down into complex tasks. A six-level hierarchical structure of the above kind has been proposed as the reference model for the functional architecture of the control system of a service robotic system for space applications (NASREM). In this framework, one may allocate the functions required to *advanced robotics* systems devoted to field or service applications, as discussed in Sect. 1.4.

6.2 Programming Environment

Programming a robotic system requires definition of a *programming environment* supported by suitable *languages*, which allows the operator imparting the task directions that the robot should execute. The programming environment is entrusted not only with the function of translating statements by means of a suitable language, but also with the function of checking correct execution of a task being executed by the robot. Therefore, robot programming environments, besides having some features in common with computer programming environments, present a number of issues related to the observation that program execution produces effects on the physical world. In other words, even if a very accurate description of physical reality is available in the programming environment, a number of situations will unavoidably occur which have not been or cannot be predicted.

As a consequence, a robot programming environment should be endowed with the following features:

- real-time operating system,
- world modelling,
- motion control,

- sensory data reading,
- interaction with physical system,
- error detection capability,
- recovery of correct operational functions,
- specific language structure.

Therefore, the requirements on a programming environment may naturally stem from the articulation into models of the preceding reference model of functional architecture. Such an environment will be clearly conditioned by the level of the architecture at which operator access is allowed. In the following, the requirements imposed on the programming environment by the functions respectively characterizing the sensory, modelling and decision modules are presented with reference to the hierarchical levels of the functional architecture.

Sensory data handling is the determining factor which qualifies a programming environment. At the servo level, real-time proprioceptive sensory data conditioning is required. At the primitive level, sensory data have to be expressed in the relevant reference frames. At the action level, geometric features of the objects interested to the action have to be extracted by high-level sensory data. At the task level, tools allowing recognition of the objects present in the scene are required.

The ability of *consulting knowledge models* is a support for a programming environment. At the servo level, on-line numerical computation of the models utilized by control algorithms is to be performed on the basis of sensory data. At the primitive level, coordinate transformations have to be operated. At the action level, it is crucial to have tools allowing system simulation and CAD modelling of elementary objects. At the task level, the programming environment should assume the functions of an expert system.

Decision functions play a fundamental role in a programming environment, since they allow the definition of the flow charts. At the servo level, on-line computation ability is required to generate the driving signals for the mechanical system. At the primitive level, logic conditioning is to be present. At the action level, process synchronization options should be available in order to implement nested loops, parallel computation and interrupt system. At the task level, the programming environment should allow management of concurrent processes, and it should be endowed with tools to test for, locate and remove mistakes from a program (debuggers) at a high-interactive level.

The evolution of programming environments has been conditioned by technology development of computer science. An analysis of this evolution leads to finding three generations of environments with respect to their functional characteristics, namely, *teaching-by-showing*, *robot-oriented programming*, and *object-oriented programming*. In the evolution of the environments, the next generation usually incorporates the functional characteristics of the previous generation.

This classification regards those features of the programming environment relative to the operator interface, and thus it has a direct correspondence with the hierarchical levels of the reference model of functional architecture. The functions associated with the servo level lead to understanding that a programming environment problem does not really exist for the operator. In fact, low-level programming concerns the use of traditional programming languages (Assembly, C) for development of real-time systems. The operator is only left with the possibility of intervening by means of simple command actuation (point-to-point, reset), reading of proprioceptive sensory data, and limited editing capability.

6.2.1 Teaching-by-Showing

The first generation has been characterized by programming techniques of *teaching-by-showing* type. The operator guides the manipulator manually or by means of a teach pendant along the desired motion path. During motion execution, the data read by joint position transducers are stored and thus they can be utilized later as references for the joint drive servos; in this way, the mechanical structure is capable of executing (playing back) the motion taught by a direct acquisition on the spot.

The programming environment does not allow implementation of logic conditioning and queuing, and thus the associated computational hardware plays elementary functions. The operator is not required to have special programming skill, and thus he/she can be a plant technician. The set-up of a working program obviously requires the robot to be available to the operator at the time of teaching, and thus the robot itself has to be taken off production. Typical applications that can be solved by this programming technique include spot welding, spray painting and, in general, simple palletizing.

With regard to the reference model of functional architecture, a programming environment based on the teaching-by-showing technique allows operator access at the primitive level.

The drawbacks of such an environment may be partially overcome by the adoption of simple programming languages which allow:

- the acquisition of a meaningful posture by teaching,
- the computation of the end-effector pose with respect to a reference frame, by means of a direct kinematics transformation,
- the assignment of a motion primitive and the trajectory parameters (usually, velocity as a percentage of the maximum velocity),
- the computation of the servo references, by means of an inverse kinematics transformation,
- the teaching sequences to be conditioned to the use of simple external sensors (presence of an object at the gripper),
- the correction of motion sequences by using simple text editors,
- simple connections to be made between subsets of elementary sequences.

Providing a teaching-by-showing environment with the the above-listed functions can be framed as an attempt to develop a structured programming environment.

6.2.2 Robot-oriented Programming

Following the advent of efficient low-cost computational means, *robot-oriented* programming environments have been developed. The need for interaction of the environment with physical reality has imposed integration of several functions, typical of high-level programming languages (BASIC, PASCAL), with those specifically required by robotic applications. In fact, many robot-oriented languages have retained the teaching-by-showing programming mode, in view of its natural characteristic of accurate interface with the physical world.

Since the general framework is that of a computer programming environment, two alternatives have been considered:

- to develop *ad hoc languages* for robotic applications,
- to develop robot *program libraries* supporting standard programming languages.

The current situation features the existence of numerous new proprietary languages, whereas it would be desirable to develop either robotic libraries to be used in the context of consolidated standards, or new general-purpose languages for industrial automation applications.

Robot-oriented languages are *structured programming* languages which incorporate high-level statements and have the characteristic of an interpreted language, in order to obtain an interactive environment allowing the programmer to check the execution of each source program statement before proceeding to the next one. Common features of such languages are:

- text editor,
- complex data representation structures,
- extensive use of predefined state variable,
- execution of matrix algebra operations,
- extensive use of symbolic representations for coordinate frames,
- possibility to specify the coordinated motion of more frames rigidly attached to objects by means of a single frame,
- inclusion of subroutines with data and parameter exchange,
- use of logic conditioning and queuing by means of flags,
- capability of parallel computing,
- functions of programmable logic controller (PLC).

With respect to the reference model of functional architecture, it can be recognized that a robot-oriented programming environment allows operator access at the action level.

In view of the structured language characteristic, the operator in this case should be an expert language programmer. Editing an application program may be performed off line, i.e., without physical availability of the robot to the operator; off-line programming demands a perfectly structured environment, though. A robotic system endowed with a robot-oriented programming language allows execution of complex applications where the robot is inserted in a work cell and interacts with other machines and devices to perform complex tasks, such as part assembly.

Finally, a programming environment that allows access at the task level of a reference model of functional architecture is characterized by an *object-oriented* language. Such an environment should have the capability of specifying a task by means of high-level statements allowing automatic execution of a number of actions on the objects present in the scene. Robot programming languages belonging to this generation are currently under development and thus they are not yet available on the market. They can be framed in the field of expert systems and artificial intelligence.

6.3 Hardware Architecture

The hierarchical structure of the functional architecture adopted as a reference model for an industrial robot's control system, together with its articulation into different functional modules, suggests hardware implementation which exploits distributed computational resources interconnected by means of suitable communication channels. To this end, it is worth recalling that the functions implemented in current control systems regard the three levels from servo to action, with a typically limited development of the functions implemented at the action level. At the servo and primitive levels, computational capabilities are required with demanding real-time constraints.

A general model of the *hardware architecture* for the control system of an industrial robot is illustrated in Fig. 6.3. In this figure, proper *boards* with autonomous computational capabilities have been associated with the functions indicated in the reference model of functional architecture of Fig. 9.2. The boards are connected to a *bus*, e.g., a VME bus, which allows support of the communication data flow; the bus bandwidth should be wide enough so as to satisfy the requirements imposed by real-time constraints.

The *system* board is typically a CPU endowed with:

- a microprocessor with mathematical coprocessor,
- a bootstrap EPROM memory,
- a local RAM memory,
- a RAM memory shared with the other boards through the bus,
- a number of serial and parallel ports interfacing the bus and the external world,
- counters, registers and timers,

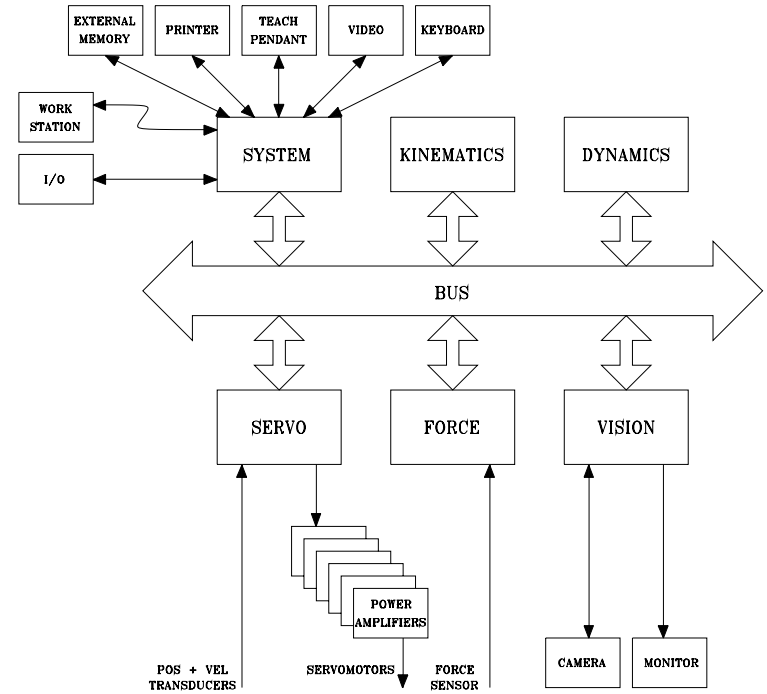

Fig. 6.3. General model of the hardware architecture of an industrial robot's control system

- an interrupt system.

The following functions are to be implemented in the system board:

- operator interface through teach pendant, keyboard, video and printer,
- interface with an external memory (hard disk) used to store data and application programs,
- interface with workstations and other control systems by means of a local communication network, e.g., Ethernet,
- I/O interface with peripheral devices in the working area, e.g., feeders, conveyors and ON/OFF sensors,
- system bootstrap,
- programming language interpreter,
- bus arbiter.

The other boards facing the bus may be endowed, besides the basic components of the system board, with a supplementary or alternative processor

(DSP, Transputer) for implementation of computationally demanding or dedicated functions. With reference to the architecture in Fig. 6.3, the following functions are implemented in the *kinematics* board:

- computation of motion primitives,
- computation of direct kinematics, inverse kinematics and Jacobian,
- test for trajectory feasibility,
- handling of kinematic redundancy.

The *dynamics* board is devoted to

- computation of inverse dynamics.

The *servo* board has the functions of:

- microinterpolation of references,
- computation of control algorithm,
- digital-to-analog conversion and interface with power amplifiers,
- handling of position and velocity transducer data,
- motion interruption in case of malfunction.

The remaining boards in the figure have been considered for the sake of an example to illustrate how the use of sensors may require local processing capabilities to retrieve significant information from the given data which can be effectively used in the sensory system. The *force* board performs the following operations:

- conditioning of data provided by the force sensor,
- representation of forces in a given coordinate frame.

The *vision* board is in charge of:

- processing data provided by the camera,
- extracting geometric features of the scene,
- localizing objects in given coordinate frames.

Although the boards face the same bus, the frequency at which data are exchanged needs not to be the same for each board. Those boards connected to the proprioceptive sensors indeed need to exchange date with the robot at the highest possible frequency (from 100 to 1000 Hz) to ensure high dynamic performance to motion control as well as to reveal end-effector contact in a very short time.

On the other hand, the kinematics and dynamics boards implement modelling functions and, as such, they do not require data update at a rate as high as that required by the servo board. In fact, manipulator postures do not vary appreciably in a very short time, at least with respect to typical operational velocities and/or accelerations of current industrial robots. Common sampling frequencies are in the range of 10 to 100 Hz.

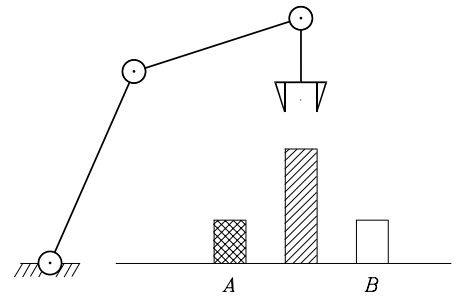

Fig. 6.4. Object pick-and-place task

Also the vision board does not require a high update rate, both because the scene is generally quasi-static, and because processing of interpretive functions are typically complex. Typical frequencies are in the range of 1 to 10 Hz.

In summary, the board access to the communication bus of a hardware control architecture may be performed according to a multirate logic which allows the solution of bus data overflow problems.

Bibliography

The features of robot control architectures are presented in [230, 25]. The NASREM architecture model has been proposed in [3]. For robot programming see [225, 139, 91]. More advanced control architectures based on artificial intelligence concepts are discussed in [8, 158].

Problems

6.1. With reference to the situation illustrated in Fig. 6.4, describe the sequence of actions required from the manipulator to pick up an object at location A and place it at location B.

6.2. For the situation of Problem 6.1, find the motion primitives in the cases of given via points and given path points.

6.3. The planar arm indicated in Fig. 6.5 is endowed with a wrist force sensor which allows the measurement of the relevant force and moment components for the execution of a peg-in-hole task. Draw the flow chart for writing a program to execute the described task.

6.4. A palletizing problem is represented in Fig. 6.6. Sixteen equal objects have to be loaded on the pallet. The manipulator's end-effector has to pick

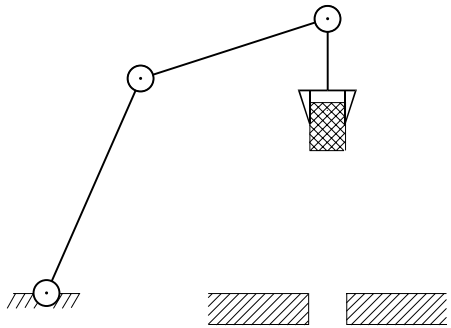

Fig. 6.5. Peg-in-hole task

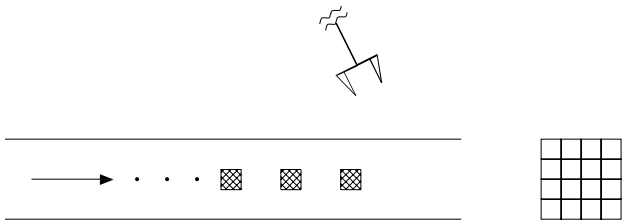

Fig. 6.6. Palletizing task of objects available on a conveyor

up the objects from a conveyor, whose feeding is commanded by the robot in such a way that the objects are always found in the same location to be picked. Write a PASCAL program to execute the task.

7

Dynamics

Derivation of the *dynamic model* of a manipulator plays an important role for simulation of motion, analysis of manipulator structures, and design of control algorithms. Simulating manipulator motion allows control strategies and motion planning techniques to be tested without the need to use a physically available system. The analysis of the dynamic model can be helpful for mechanical design of prototype arms. Computation of the forces and torques required for the execution of typical motions provides useful information for designing joints, transmissions and actuators. The goal of this chapter is to present two methods for derivation of the equations of motion of a manipulator in the *joint space*. The first method is based on the *Lagrange formulation* and is conceptually simple and systematic. The second method is based on the *Newton–Euler formulation* and yields the model in a recursive form; it is computationally more efficient since it exploits the typically open structure of the manipulator kinematic chain. Then, a technique for *dynamic parameter identification* is presented. Further, the problems of *direct dynamics* and *inverse dynamics* are formalized, and a technique for trajectory *dynamic scaling* is introduced, which adapts trajectory planning to the dynamic characteristics of the manipulator. The chapter ends with the derivation of the *dynamic model* of a manipulator in the *operational space* and the definition of the *dynamic manipulability ellipsoid*.

7.1 Lagrange Formulation

The dynamic model of a manipulator provides a description of the relationship between the joint actuator torques and the motion of the structure.

With *Lagrange formulation*, the equations of motion can be derived in a systematic way independently of the reference coordinate frame. Once a set of variables q_i, $i = 1, \ldots, n$, termed *generalized coordinates*, are chosen which effectively describe the link positions of an n-DOF manipulator, the

Lagrangian of the mechanical system can be defined as a function of the generalized coordinates:

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{7.1}$$

where T and U respectively denote the total *kinetic energy* and *potential energy* of the system.

The Lagrange equations are expressed by

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \qquad i = 1, \ldots, n \tag{7.2}$$

where ξ_i is the *generalized force* associated with the generalized coordinate q_i. Equations (7.2) can be written in compact form as

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^T = \boldsymbol{\xi} \tag{7.3}$$

where, for a manipulator with an open kinematic chain, the generalized coordinates are gathered in the vector of *joint variables* $\boldsymbol{q}$. The contributions to the generalized forces are given by the nonconservative forces, i.e., the joint actuator torques, the joint friction torques, as well as the joint torques induced by end-effector forces at the contact with the environment.[1]

The equations in (7.2) establish the relations existing between the generalized forces applied to the manipulator and the joint positions, velocities and accelerations. Hence, they allow the derivation of the dynamic model of the manipulator starting from the determination of kinetic energy and potential energy of the mechanical system.

Example 7.1

In order to understand the Lagrange formulation technique for deriving the dynamic model, consider again the simple case of the pendulum in Example 5.1. With reference to Fig. 5.8, let ϑ denote the angle with respect to the reference position of the body hanging down ($\vartheta = 0$). By choosing ϑ as the generalized coordinate, the kinetic energy of the system is given by

$$\mathcal{T} = \frac{1}{2} I \dot{\vartheta}^2 + \frac{1}{2} I_m k_r^2 \dot{\vartheta}^2.$$

The system potential energy, defined at less than a constant, is expressed by

$$\mathcal{U} = mg\ell(1 - \cos\vartheta).$$

Therefore, the Lagrangian of the system is

$$\mathcal{L} = \frac{1}{2} I \dot{\vartheta}^2 + \frac{1}{2} I_m k_r^2 \dot{\vartheta}^2 - mg\ell(1 - \cos\vartheta).$$

[1] The term *torque* is used as a synonym of joint *generalized force*.

Substituting this expression in the Lagrange equation in (7.2) yields

$$(I + I_m k_r^2)\ddot{\vartheta} + mg\ell \sin\vartheta = \xi.$$

The generalized force ξ is given by the contributions of the actuation torque τ at the joint and of the viscous friction torques $-F\dot{\vartheta}$ and $-F_m k_r^2 \dot{\vartheta}$, where the latter has been reported to the joint side. Hence, it is

$$\xi = \tau - F\dot{\vartheta} - F_m k_r^2 \dot{\vartheta}$$

leading to the complete dynamic model of the system as the second-order differential equation

$$(I + I_m k_r^2)\ddot{\vartheta} + (F + F_m k_r^2)\dot{\vartheta} + mg\ell \sin\vartheta = \tau.$$

It is easy to verify how this equation is equivalent to (5.25) when reported to the joint side.

7.1.1 Computation of Kinetic Energy

Consider a manipulator with n *rigid links*. The total kinetic energy is given by the sum of the contributions relative to the motion of each link and the contributions relative to the motion of each joint actuator:[2]

$$\mathcal{T} = \sum_{i=1}^{n} (\mathcal{T}_{\ell_i} + \mathcal{T}_{m_i}), \tag{7.4}$$

where $\mathcal{T}_{\ell_i}$ is the kinetic energy of Link i and $\mathcal{T}_{m_i}$ is the kinetic energy of the motor actuating Joint i.

The kinetic energy contribution of Link i is given by

$$\mathcal{T}_{\ell_i} = \frac{1}{2} \int_{V_{\ell_i}} \dot{\boldsymbol{p}}_i^{*T} \dot{\boldsymbol{p}}_i^* \rho dV, \tag{7.5}$$

where $\dot{\boldsymbol{p}}_i^*$ denotes the linear velocity vector and ρ is the density of the elementary particle of volume dV; V_{ℓ_i} is the volume of Link i.

Consider the position vector $\boldsymbol{p}_i^*$ of the elementary particle and the position vector $\boldsymbol{p}_{C_i}$ of the link centre of mass, both expressed in the *base frame*. One has

$$\boldsymbol{r}_i = [\, r_{ix} \quad r_{iy} \quad r_{iz} \,]^T = \boldsymbol{p}_i^* - \boldsymbol{p}_{\ell_i} \tag{7.6}$$

with

$$\boldsymbol{p}_{\ell_i} = \frac{1}{m_{\ell_i}} \int_{V_{\ell_i}} \boldsymbol{p}_i^* \rho dV \tag{7.7}$$

[2] Link 0 is fixed and thus gives no contribution.

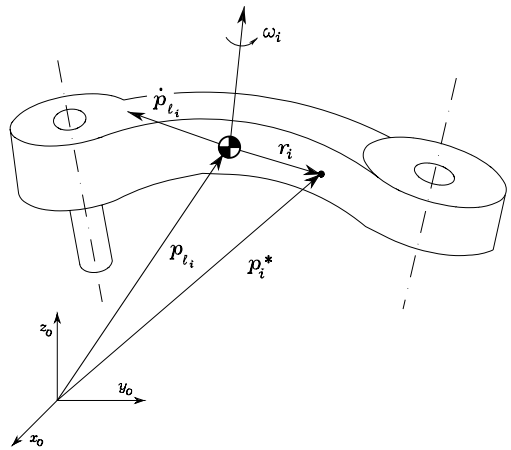

Fig. 7.1. Kinematic description of Link i for Lagrange formulation

where m_{ℓ_i} is the link mass. As a consequence, the link point velocity can be expressed as

$$\begin{aligned}\dot{\boldsymbol{p}}_i^* &= \dot{\boldsymbol{p}}_{\ell_i} + \boldsymbol{\omega}_i \times \boldsymbol{r}_i \\ &= \dot{\boldsymbol{p}}_{\ell_i} + \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i,\end{aligned} \tag{7.8}$$

where $\dot{\boldsymbol{p}}_{\ell_i}$ is the linear velocity of the centre of mass and $\boldsymbol{\omega}_i$ is the angular velocity of the link (Fig. 7.1).

By substituting the velocity expression (7.8) into (7.5), it can be recognized that the kinetic energy of each link is formed by the following contributions.

Translational

The contribution is

$$\frac{1}{2}\int_{V_{\ell_i}} \dot{\boldsymbol{p}}_{\ell_i}^T \dot{\boldsymbol{p}}_{\ell_i} \rho dV = \frac{1}{2} m_{\ell_i} \dot{\boldsymbol{p}}_{\ell_i}^T \dot{\boldsymbol{p}}_{\ell_i}. \tag{7.9}$$

Mutual

The contribution is

$$2\left(\frac{1}{2}\int_{V_{\ell_i}} \dot{\boldsymbol{p}}_{\ell_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i \rho dV\right) = 2\left(\frac{1}{2}\dot{\boldsymbol{p}}_{\ell_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i)\int_{V_{\ell_i}} (\boldsymbol{p}_i^* - \boldsymbol{p}_{\ell_i})\rho dV\right) = 0$$

since, by virtue of (7.7), it is

$$\int_{V_{\ell_i}} \boldsymbol{p}_i^* \rho dV = \boldsymbol{p}_{\ell_i} \int_{V_{\ell_i}} \rho dV.$$

Rotational

The contribution is

$$\frac{1}{2}\int_{V_{\ell_i}} \boldsymbol{r}_i^T \boldsymbol{S}^T(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i \rho dV = \frac{1}{2}\boldsymbol{\omega}_i^T \left(\int_{V_{\ell_i}} \boldsymbol{S}^T(\boldsymbol{r}_i)\boldsymbol{S}(\boldsymbol{r}_i)\rho dV\right)\boldsymbol{\omega}_i$$

where the property $\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i = -\boldsymbol{S}(\boldsymbol{r}_i)\boldsymbol{\omega}_i$ has been exploited. In view of the expression of the matrix operator $\boldsymbol{S}(\cdot)$

$$\boldsymbol{S}(\boldsymbol{r}_i) = \begin{bmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{bmatrix},$$

it is

$$\frac{1}{2}\int_{V_{\ell_i}} \boldsymbol{r}_i^T \boldsymbol{S}^T(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i \rho dV = \frac{1}{2}\boldsymbol{\omega}_i^T \boldsymbol{I}_{\ell_i}\boldsymbol{\omega}_i. \tag{7.10}$$

The matrix

$$\begin{aligned}\boldsymbol{I}_{\ell_i} &= \begin{bmatrix} \int(r_{iy}^2 + r_{iz}^2)\rho dV & -\int r_{ix}r_{iy}\rho dV & -\int r_{ix}r_{iz}\rho dV \\ * & \int(r_{ix}^2 + r_{iz}^2)\rho dV & -\int r_{iy}r_{iz}\rho dV \\ * & * & \int(r_{ix}^2 + r_{iy}^2)\rho dV \end{bmatrix} \\ &= \begin{bmatrix} I_{\ell_i xx} & -I_{\ell_i xy} & -I_{\ell_i xz} \\ * & I_{\ell_i yy} & -I_{\ell_i yz} \\ * & * & I_{\ell_i zz} \end{bmatrix}.\end{aligned} \tag{7.11}$$

is symmetric[3] and represents the *inertia tensor* relative to the centre of mass of Link i when expressed in the base frame. Notice that the position of Link i depends on the manipulator configuration and thus the inertia tensor, when expressed in the base frame, is configuration-dependent. If the angular velocity of Link i is expressed with reference to a frame attached to the link (as in the Denavit–Hartenberg convention), it is

$$\boldsymbol{\omega}_i^i = \boldsymbol{R}_i^T \boldsymbol{\omega}_i$$

where $\boldsymbol{R}_i$ is the rotation matrix from Link i frame to the base frame. When referred to the link frame, the inertia tensor is constant. Let $\boldsymbol{I}_{\ell_i}^i$ denote such tensor; then it is easy to verify the following relation:

$$\boldsymbol{I}_{\ell_i} = \boldsymbol{R}_i \boldsymbol{I}_{\ell_i}^i \boldsymbol{R}_i^T. \tag{7.12}$$

If the axes of Link i frame coincide with the central axes of inertia, then the inertia products are null and the inertia tensor relative to the centre of mass is a diagonal matrix.

[3] The symbol '$*$' has been used to avoid rewriting the symmetric elements.

By summing the translational and rotational contributions (7.9) and (7.10), the kinetic energy of Link i is

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell_i} \dot{\boldsymbol{p}}_{\ell_i}^T \dot{\boldsymbol{p}}_{\ell_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \boldsymbol{R}_i \boldsymbol{I}_{\ell_i}^i \boldsymbol{R}_i^T \boldsymbol{\omega}_i. \tag{7.13}$$

At this point, it is necessary to express the kinetic energy as a function of the generalized coordinates of the system, that are the joint variables. To this end, the geometric method for Jacobian computation can be applied to the intermediate link other than the end-effector, yielding

$$\dot{\boldsymbol{p}}_{\ell_i} = \boldsymbol{\jmath}_{P1}^{(\ell_i)} \dot{q}_1 + \ldots + \boldsymbol{\jmath}_{Pi}^{(\ell_i)} \dot{q}_i = \boldsymbol{J}_P^{(\ell_i)} \dot{\boldsymbol{q}} \tag{7.14}$$

$$\boldsymbol{\omega}_i = \boldsymbol{\jmath}_{O1}^{(\ell_i)} \dot{q}_1 + \ldots + \boldsymbol{\jmath}_{Oi}^{(\ell_i)} \dot{q}_i = \boldsymbol{J}_O^{(\ell_i)} \dot{\boldsymbol{q}}, \tag{7.15}$$

where the contributions of the Jacobian columns relative to the joint velocities have been taken into account up to current Link i. The Jacobians to consider are then:

$$\boldsymbol{J}_P^{(\ell_i)} = \begin{bmatrix} \boldsymbol{\jmath}_{P1}^{(\ell_i)} & \ldots & \boldsymbol{\jmath}_{Pi}^{(\ell_i)} & \boldsymbol{0} & \ldots & \boldsymbol{0} \end{bmatrix} \tag{7.16}$$

$$\boldsymbol{J}_O^{(\ell_i)} = \begin{bmatrix} \boldsymbol{\jmath}_{O1}^{(\ell_i)} & \ldots & \boldsymbol{\jmath}_{Oi}^{(\ell_i)} & \boldsymbol{0} & \ldots & \boldsymbol{0} \end{bmatrix}; \tag{7.17}$$

the columns of the matrices in (7.16) and (7.17) can be computed according to (3.30), giving

$$\boldsymbol{\jmath}_{Pj}^{(\ell_i)} = \begin{cases} \boldsymbol{z}_{j-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{z}_{j-1} \times (\boldsymbol{p}_{\ell_i} - \boldsymbol{p}_{j-1}) & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.18}$$

$$\boldsymbol{\jmath}_{Oj}^{(\ell_i)} = \begin{cases} \boldsymbol{0} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{z}_{j-1} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{7.19}$$

where $\boldsymbol{p}_{j-1}$ is the position vector of the origin of Frame $j-1$ and $\boldsymbol{z}_{j-1}$ is the unit vector of axis z of Frame $j-1$. It follows that the kinetic energy of Link i in (7.13) can be written as

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell_i} \dot{\boldsymbol{q}}^T \boldsymbol{J}_P^{(\ell_i)T} \boldsymbol{J}_P^{(\ell_i)} \dot{\boldsymbol{q}} + \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{J}_O^{(\ell_i)T} \boldsymbol{R}_i \boldsymbol{I}_{\ell_i}^i \boldsymbol{R}_i^T \boldsymbol{J}_O^{(\ell_i)} \dot{\boldsymbol{q}}. \tag{7.20}$$

The kinetic energy contribution of the motor of Joint i can be computed in a formally analogous way to that of the link. Consider the typical case of rotary electric motors (that can actuate both revolute and prismatic joints by means of suitable transmissions). It can be assumed that the contribution of the fixed part (stator) is included in that of the link on which such motor is located, and thus the sole contribution of the rotor is to be computed.

With reference to Fig. 7.2, the motor of Joint i is assumed to be located on Link $i-1$. In practice, in the design of the mechanical structure of an open kinematic chain manipulator one attempts to locate the motors as close as possible to the base of the manipulator so as to lighten the dynamic load of

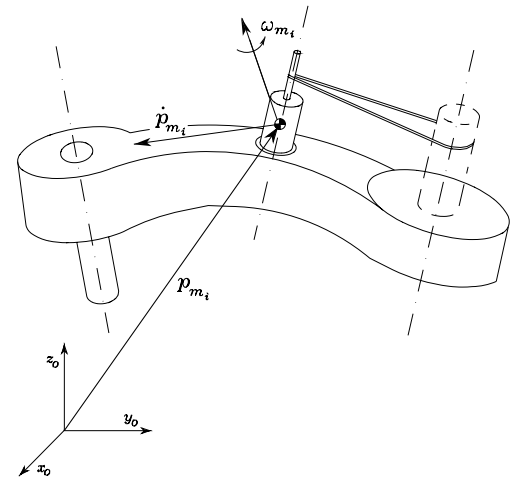

Fig. 7.2. Kinematic description of Motor i

the first joints of the chain. The joint actuator torques are delivered by the motors by means of mechanical transmissions (gears).[4] The contribution of the gears to the kinetic energy can be suitably included in that of the motor. It is assumed that no induced motion occurs, i.e., the motion of Joint i does not actuate the motion of other joints.

The kinetic energy of Rotor i can be written as

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} \dot{\boldsymbol{p}}_{m_i}^T \dot{\boldsymbol{p}}_{m_i} + \frac{1}{2} \boldsymbol{\omega}_{m_i}^T \boldsymbol{I}_{m_i} \boldsymbol{\omega}_{m_i}, \tag{7.21}$$

where m_{m_i} is the mass of the rotor, $\dot{\boldsymbol{p}}_{m_i}$ denotes the linear velocity of the centre of mass of the rotor, $\boldsymbol{I}_{m_i}$ is the inertia tensor of the rotor relative to its centre of mass, and $\boldsymbol{\omega}_{m_i}$ denotes the angular velocity of the rotor.

Let ϑ_{m_i} denote the angular position of the rotor. On the assumption of a *rigid transmission*, one has

$$k_{ri} \dot{q}_i = \dot{\vartheta}_{m_i} \tag{7.22}$$

where k_{ri} is the gear reduction ratio. Notice that, in the case of actuation of a prismatic joint, the gear reduction ratio is a dimensional quantity.

According to the angular velocity composition rule (3.18) and the relation (7.22), the total angular velocity of the rotor is

$$\boldsymbol{\omega}_{m_i} = \boldsymbol{\omega}_{i-1} + k_{ri} \dot{q}_i \boldsymbol{z}_{m_i} \tag{7.23}$$

where $\boldsymbol{\omega}_{i-1}$ is the angular velocity of Link $i-1$ on which the motor is located, and $\boldsymbol{z}_{m_i}$ denotes the unit vector along the rotor axis.

[4] Alternatively, the joints may be actuated by torque motors directly coupled to the rotation axis without gears.

To express the rotor kinetic energy as a function of the joint variables, it is worth expressing the linear velocity of the rotor centre of mass — similarly to (7.14) — as

$$\dot{\boldsymbol{p}}_{m_i} = \boldsymbol{J}_P^{(m_i)} \dot{\boldsymbol{q}}. \tag{7.24}$$

The Jacobian to compute is then

$$\boldsymbol{J}_P^{(m_i)} = \begin{bmatrix} \boldsymbol{\jmath}_{P1}^{(m_i)} & \dots & \boldsymbol{\jmath}_{P,i-1}^{(m_i)} & \boldsymbol{0} & \dots & \boldsymbol{0} \end{bmatrix} \tag{7.25}$$

whose columns are given by

$$\boldsymbol{\jmath}_{Pj}^{(m_i)} = \begin{cases} \boldsymbol{z}_{j-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{z}_{j-1} \times (\boldsymbol{p}_{m_i} - \boldsymbol{p}_{j-1}) & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.26}$$

where $\boldsymbol{p}_{j-1}$ is the position vector of the origin of Frame $j-1$. Notice that $\boldsymbol{\jmath}_{Pi}^{(m_i)} = \boldsymbol{0}$ in (7.25), since the centre of mass of the rotor has been taken along its axis of rotation.

The angular velocity in (7.23) can be expressed as a function of the joint variables, i.e.,

$$\boldsymbol{\omega}_{m_i} = \boldsymbol{J}_O^{(m_i)} \dot{\boldsymbol{q}}. \tag{7.27}$$

The Jacobian to compute is then

$$\boldsymbol{J}_O^{(m_i)} = \begin{bmatrix} \boldsymbol{\jmath}_{O1}^{(m_i)} & \dots & \boldsymbol{\jmath}_{O,i-1}^{(m_i)} & \boldsymbol{\jmath}_{Oi}^{(m_i)} & \boldsymbol{0} & \dots & \boldsymbol{0} \end{bmatrix} \tag{7.28}$$

whose columns, in view of (7.23), (7.15), are respectively given by

$$\boldsymbol{\jmath}_{Oj}^{(m_i)} = \begin{cases} \boldsymbol{\jmath}_{Oj}^{(\ell_i)} & j = 1, \dots, i-1 \\ k_{ri} \boldsymbol{z}_{m_i} & j = i. \end{cases} \tag{7.29}$$

To compute the second relation in (7.29), it is sufficient to know the components of the unit vector of the rotor rotation axis $\boldsymbol{z}_{m_i}$ with respect to the base frame. Hence, the kinetic energy of Rotor i can be written as

$$\mathcal{T}_{m_i} = \frac{1}{2} m_{m_i} \dot{\boldsymbol{q}}^T \boldsymbol{J}_P^{(m_i)T} \boldsymbol{J}_P^{(m_i)} \dot{\boldsymbol{q}} + \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{J}_O^{(m_i)T} \boldsymbol{R}_{m_i} \boldsymbol{I}_{m_i}^{m_i} \boldsymbol{R}_{m_i}^T \boldsymbol{J}_O^{(m_i)} \dot{\boldsymbol{q}}. \tag{7.30}$$

Finally, by summing the various contributions relative to the single links (7.20) and single rotors (7.30) as in (7.4), the total kinetic energy of the manipulator with actuators is given by the quadratic form

$$\mathcal{T} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}(\boldsymbol{q}) \dot{q}_i \dot{q}_j = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{B}(\boldsymbol{q}) \dot{\boldsymbol{q}} \tag{7.31}$$

where

$$\boldsymbol{B}(\boldsymbol{q}) = \sum_{i=1}^{n} \Big(m_{\ell_i} \boldsymbol{J}_P^{(\ell_i)T} \boldsymbol{J}_P^{(\ell_i)} + \boldsymbol{J}_O^{(\ell_i)T} \boldsymbol{R}_i \boldsymbol{I}_{\ell_i}^i \boldsymbol{R}_i^T \boldsymbol{J}_O^{(\ell_i)} + m_{m_i} \boldsymbol{J}_P^{(m_i)T} \boldsymbol{J}_P^{(m_i)} + \boldsymbol{J}_O^{(m_i)T} \boldsymbol{R}_{m_i} \boldsymbol{I}_{m_i}^{m_i} \boldsymbol{R}_{m_i}^T \boldsymbol{J}_O^{(m_i)} \Big) \tag{7.32}$$

is the $(n \times n)$ *inertia matrix* which is:

- *symmetric*,
- *positive definite*,
- (in general) *configuration-dependent*.

7.1.2 Computation of Potential Energy

As done for kinetic energy, the potential energy stored in the manipulator is given by the sum of the contributions relative to each link as well as to each rotor:

$$\mathcal{U} = \sum_{i=1}^{n} (\mathcal{U}_{\ell_i} + \mathcal{U}_{m_i}). \tag{7.33}$$

On the assumption of *rigid links*, the contribution due only to gravitational forces[5] is expressed by

$$\mathcal{U}_{\ell_i} = -\int_{V_{\ell_i}} \boldsymbol{g}_0^T \boldsymbol{p}_i^* \rho dV = -m_{\ell_i} \boldsymbol{g}_0^T \boldsymbol{p}_{\ell_i} \tag{7.34}$$

where $\boldsymbol{g}_0$ is the gravity acceleration vector in the base frame (e.g., $\boldsymbol{g}_0 = [0 \quad 0 \quad -g]^T$ if z is the vertical axis), and (7.7) has been utilized for the coordinates of the centre of mass of Link i. As regards the contribution of Rotor i, similarly to (7.34), one has

$$\mathcal{U}_{m_i} = -m_{m_i} \boldsymbol{g}_0^T \boldsymbol{p}_{m_i}. \tag{7.35}$$

By substituting (7.34), (7.35) into (7.33), the *potential energy* is given by

$$\mathcal{U} = -\sum_{i=1}^{n} (m_{\ell_i} \boldsymbol{g}_0^T \boldsymbol{p}_{\ell_i} + m_{m_i} \boldsymbol{g}_0^T \boldsymbol{p}_{m_i}) \tag{7.36}$$

which reveals that potential energy, through the vectors $\boldsymbol{p}_{\ell_i}$ and $\boldsymbol{p}_{m_i}$ is a function only of the joint variables $\boldsymbol{q}$, and *not* of the joint velocities $\dot{\boldsymbol{q}}$.

7.1.3 Equations of Motion

Having computed the total kinetic and potential energy of the system as in (7.31), (7.36), the Lagrangian (7.1) for the manipulator can be written as

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q}). \tag{7.37}$$

Taking the derivatives required by Lagrange equations in (7.3) and recalling that U does not depend on $\dot{\boldsymbol{q}}$ yields

$$\boldsymbol{B}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\xi} \tag{7.38}$$

[5] In the case of link flexibility, one would have an additional contribution due to elastic forces.

where

$$\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right)^T + \left(\frac{\partial \mathcal{U}(\boldsymbol{q})}{\partial \boldsymbol{q}}\right)^T.$$

In detail, noticing that $\mathcal{U}$ in (7.36) does not depend on $\dot{\boldsymbol{q}}$ and accounting for (7.31) yields

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) = \frac{d}{dt}\left(\frac{\partial \mathcal{T}}{\partial \dot{q}_i}\right) &= \sum_{j=1}^{n} b_{ij}(\boldsymbol{q})\ddot{q}_j + \sum_{j=1}^{n}\frac{db_{ij}(\boldsymbol{q})}{dt}\dot{q}_j \\ &= \sum_{j=1}^{n} b_{ij}(\boldsymbol{q})\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{ij}(\boldsymbol{q})}{\partial q_k}\dot{q}_k\dot{q}_j\end{aligned}$$

and

$$\frac{\partial \mathcal{T}}{\partial q_i} = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{jk}(\boldsymbol{q})}{\partial q_i}\dot{q}_k\dot{q}_j$$

where the indices of summation have been conveniently switched. Further, in view of (7.14), (7.24), it is

$$\begin{aligned}\frac{\partial \mathcal{U}}{\partial q_i} &= -\sum_{j=1}^{n}\left(m_{\ell_j}\boldsymbol{g}_0^T\frac{\partial \boldsymbol{p}_{\ell_j}}{\partial q_i} + m_{m_j}\boldsymbol{g}_0^T\frac{\partial \boldsymbol{p}_{m_j}}{\partial q_i}\right) \qquad (7.39)\\ &= -\sum_{j=1}^{n}\left(m_{\ell_j}\boldsymbol{g}_0^T\boldsymbol{J}_{Pi}^{(\ell_j)}(\boldsymbol{q}) + m_{m_j}\boldsymbol{g}_0^T\boldsymbol{J}_{Pi}^{(m_j)}(\boldsymbol{q})\right) = g_i(\boldsymbol{q})\end{aligned}$$

where, again, the index of summation has been changed.

As a result, the equations of motion are

$$\sum_{j=1}^{n} b_{ij}(\boldsymbol{q})\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}(\boldsymbol{q})\dot{q}_k\dot{q}_j + g_i(\boldsymbol{q}) = \xi_i \qquad i = 1,\dots,n. \qquad (7.40)$$

where

$$h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}. \qquad (7.41)$$

A physical interpretation of (7.40) reveals that:

- For the *acceleration terms*:
  - The coefficient b_{ii} represents the moment of inertia at Joint i axis, in the current manipulator configuration, when the other joints are blocked.
  - The coefficient b_{ij} accounts for the effect of acceleration of Joint j on Joint j.
- For the *quadratic velocity terms*:
  - The term $h_{iij}\dot{q}_j^2$ represents the *centrifugal* effect induced on Joint i by velocity of Joint j; notice that $h_{iii} = 0$, since $\partial b_{ii}/\partial q_i = 0$.
  - The term $h_{ijk}\dot{q}_j\dot{q}_k$ represents the *Coriolis* effect induced on Joint i by velocities of Joints j and k.
- For the *configuration-dependent terms*:
  - The term g_i represents the moment generated at Joint i axis of the manipulator, in the current configuration, by the presence of gravity.

Some joint dynamic couplings, e.g., coefficients b_{ij} and h_{ijk}, may be reduced or zeroed when designing the structure, so as to simplify the control problem.

Regarding the nonconservative forces doing work at the manipulator joints, these are given by the *actuation torques* $\boldsymbol{\tau}$ minus the *viscous friction* torques $\boldsymbol{F}_v\dot{\boldsymbol{q}}$ and the static friction torques $\boldsymbol{f}_s(\boldsymbol{q},\dot{\boldsymbol{q}})$: $\boldsymbol{F}_v$ denotes the $(n\times n)$ diagonal matrix of viscous friction coefficients. As a simplified model of static friction torques, one may consider the *Coulomb friction* torques $\boldsymbol{F}_s\,\mathbf{sgn}\,(\dot{\boldsymbol{q}})$, where $\boldsymbol{F}_s$ is an $(n\times n)$ diagonal matrix and $\mathbf{sgn}\,(\dot{\boldsymbol{q}})$ denotes the $(n\times 1)$ vector whose components are given by the sign functions of the single joint velocities.

If the manipulator's end-effector is in contact with an environment, a portion of the actuation torques is used to balance the torques induced at the joints by the contact forces. According to a relation formally analogous to (3.111), such torques are given by $\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e$ where $\boldsymbol{h}_e$ denotes the vector of force and moment exerted by the end-effector on the environment.

In summary, the equations of motion (7.38) can be rewritten in the compact matrix form which represents the *joint space dynamic model*:

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{F}_s\,\mathbf{sgn}\,(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e \qquad (7.42)$$

where $\boldsymbol{C}$ is a suitable $(n\times n)$ matrix such that its elements c_{ij} satisfy the equation

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_k\dot{q}_j. \qquad (7.43)$$

7.2 Notable Properties of Dynamic Model

In the following, two *notable properties* of the dynamic model are presented which will be useful for dynamic parameter identification as well as for deriving control algorithms.

7.2.1 Skew-symmetry of Matrix $\dot{\boldsymbol{B}} - 2\boldsymbol{C}$

The choice of the matrix $\boldsymbol{C}$ is not unique, since there exist several matrices $\boldsymbol{C}$ whose elements satisfy (7.43). A particular choice can be obtained by

elaborating the term on the right-hand side of (7.43) and accounting for the expressions of the coefficients h_{ijk} in (7.41). To this end, one has

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\dot{q}_k\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j.$$

Splitting the first term on the right-hand side by an opportune switch of summation between j and k yields

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\dot{q}_k\dot{q}_j + \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j.$$

As a consequence, the generic element of $\boldsymbol{C}$ is

$$c_{ij} = \sum_{k=1}^{n} c_{ijk}\dot{q}_k \tag{7.44}$$

where the coefficients

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right) \tag{7.45}$$

are termed *Christoffel symbols of the first type*. Notice that, in view of the symmetry of $\boldsymbol{B}$, it is

$$c_{ijk} = c_{ikj}. \tag{7.46}$$

This choice for the matrix $\boldsymbol{C}$ leads to deriving the following notable property of the equations of motion (7.42). The matrix

$$\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \tag{7.47}$$

is *skew-symmetric*; that is, given any $(n\times 1)$ vector $\boldsymbol{w}$, the following relation holds:

$$\boldsymbol{w}^T\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{w} = 0. \tag{7.48}$$

In fact, substituting the coefficient (7.45) into (7.44) gives

$$c_{ij} = \frac{1}{2}\sum_{k=1}^{n}\frac{\partial b_{ij}}{\partial q_k}\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k = \frac{1}{2}\dot{b}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k$$

and then the expression of the generic element of the matrix $\boldsymbol{N}$ in (7.47) is

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = \sum_{k=1}^{n}\left(\frac{\partial b_{jk}}{\partial q_i} - \frac{\partial b_{ik}}{\partial q_j}\right)\dot{q}_k.$$

The result follows by observing that

$$n_{ij} = -n_{ji}.$$

An interesting property which is a direct implication of the skew-symmetry of $\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is that, by setting $\boldsymbol{w} = \dot{\boldsymbol{q}}$,

$$\dot{\boldsymbol{q}}^T\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = 0; \tag{7.49}$$

notice that (7.49) does not imply (7.48), since $\boldsymbol{N}$ is a function of $\dot{\boldsymbol{q}}$, too.

It can be shown that (7.49) holds for any choice of the matrix $\boldsymbol{C}$, since it is a result of the principle of conservation of energy (*Hamilton*). By virtue of this principle, the total time derivative of kinetic energy is balanced by the power generated by all the forces acting on the manipulator joints. For the mechanical system at issue, one may write

$$\frac{1}{2}\frac{d}{dt}\left(\dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) = \dot{\boldsymbol{q}}^T\left(\boldsymbol{\tau} - \boldsymbol{F}_v\dot{\boldsymbol{q}} - \boldsymbol{F}_s\,\mathbf{sgn}\,(\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e\right). \tag{7.50}$$

Taking the derivative on the left-hand side of (7.50) gives

$$\frac{1}{2}\dot{\boldsymbol{q}}^T\dot{\boldsymbol{B}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}$$

and substituting the expression of $\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}}$ in (7.42) yields

$$\frac{1}{2}\frac{d}{dt}\left(\dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) = \frac{1}{2}\dot{\boldsymbol{q}}^T\left(\dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right)\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\left(\boldsymbol{\tau} - \boldsymbol{F}_v\dot{\boldsymbol{q}} - \boldsymbol{F}_s\,\mathbf{sgn}\,(\dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e\right). \tag{7.51}$$

A direct comparison of the right-hand sides of (7.50) and (7.51) leads to the result established by (7.49).

To summarize, the relation (7.49) holds for any choice of the matrix $\boldsymbol{C}$, since it is a direct consequence of the physical properties of the system, whereas the relation (7.48) holds only for the particular choice of the elements of $\boldsymbol{C}$ as in (7.44), (7.45).

7.2.2 Linearity in the Dynamic Parameters

An important property of the dynamic model is the *linearity* with respect to the *dynamic parameters* characterizing the manipulator links and rotors.

In order to determine such parameters, it is worth associating the kinetic and potential energy contributions of each rotor with those of the link on which it is located. Hence, by considering the union of Link i and Rotor $i+1$ (*augmented Link* i), the kinetic energy contribution is given by

$$\mathcal{T}_i = \mathcal{T}_{\ell_i} + \mathcal{T}_{m_{i+1}} \tag{7.52}$$

where

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell i} \dot{\boldsymbol{p}}_{\ell_i}^T \dot{\boldsymbol{p}}_{\ell_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \boldsymbol{I}_{\ell_i} \boldsymbol{\omega}_i \tag{7.53}$$

and

$$\mathcal{T}_{m_{i+1}} = \frac{1}{2} m_{m_{i+1}} \dot{\boldsymbol{p}}_{m_{i+1}}^T \dot{\boldsymbol{p}}_{m_{i+1}} + \frac{1}{2} \boldsymbol{\omega}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{\omega}_{m_{i+1}}. \tag{7.54}$$

With reference to the centre of mass of the augmented link, the linear velocities of the link and rotor can be expressed according to (3.26) as

$$\dot{\boldsymbol{p}}_{\ell_i} = \dot{\boldsymbol{p}}_{C_i} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{C_i,\ell_i} \tag{7.55}$$

$$\dot{\boldsymbol{p}}_{m_{i+1}} = \dot{\boldsymbol{p}}_{C_i} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{C_i,m_{i+1}} \tag{7.56}$$

with

$$\boldsymbol{r}_{C_i,\ell_i} = \boldsymbol{p}_{\ell_i} - \boldsymbol{p}_{C_i} \tag{7.57}$$

$$\boldsymbol{r}_{C_i,m_{i+1}} = \boldsymbol{p}_{m_{i+1}} - \boldsymbol{p}_{C_i}, \tag{7.58}$$

where $\boldsymbol{p}_{C_i}$ denotes the position vector of the centre of mass of augmented Link i.

Substituting (7.55) into (7.53) gives

$$\begin{aligned}\mathcal{T}_{\ell_i} = {} & \frac{1}{2} m_{\ell i} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{\ell_i} \boldsymbol{r}_{C_i,\ell_i} \\ & + \frac{1}{2} m_{\ell_i} \boldsymbol{\omega}_i^T \boldsymbol{S}^T(\boldsymbol{r}_{C_i,\ell_i}) \boldsymbol{S}(\boldsymbol{r}_{C_i,\ell_i}) \boldsymbol{\omega}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \boldsymbol{I}_{\ell_i} \boldsymbol{\omega}_i.\end{aligned} \tag{7.59}$$

By virtue of *Steiner theorem*, the matrix

$$\bar{\boldsymbol{I}}_{\ell_i} = \boldsymbol{I}_{\ell_i} + m_{\ell_i} \boldsymbol{S}^T(\boldsymbol{r}_{C_i,\ell_i}) \boldsymbol{S}(\boldsymbol{r}_{C_i,\ell_i}) \tag{7.60}$$

represents the inertia tensor relative to the overall centre of mass $\boldsymbol{p}_{C_i}$, which contains an additional contribution due to the translation of the pole with respect to which the tensor is evaluated, as in (7.57). Therefore, (7.59) can be written as

$$\mathcal{T}_{\ell_i} = \frac{1}{2} m_{\ell i} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{\ell_i} \boldsymbol{r}_{C_i,\ell_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_{\ell_i} \boldsymbol{\omega}_i. \tag{7.61}$$

In a similar fashion, substituting (7.56) into (7.54) and exploiting (7.23) yields

$$\begin{aligned}\mathcal{T}_{m_{i+1}} = {} & \frac{1}{2} m_{m_{i+1}} \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \dot{\boldsymbol{p}}_{C_i}^T \boldsymbol{S}(\boldsymbol{\omega}_i) m_{m_{i+1}} \boldsymbol{r}_{C_i,m_{i+1}} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_{m_{i+1}} \boldsymbol{\omega}_i \\ & + k_{r,i+1} \dot{q}_{i+1} \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{\omega}_i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{z}_{m_{i+1}},\end{aligned} \tag{7.62}$$

where

$$\bar{\boldsymbol{I}}_{m_{i+1}} = \boldsymbol{I}_{m_{i+1}} + m_{m_{i+1}} \boldsymbol{S}^T(\boldsymbol{r}_{C_i,m_{i+1}}) \boldsymbol{S}(\boldsymbol{r}_{C_i,m_{i+1}}). \tag{7.63}$$

Summing the contributions in (7.61), (7.62) as in (7.52) gives the expression of the kinetic energy of augmented Link i in the form

$$\begin{aligned}\mathcal{T}_i = {} & \frac{1}{2} m_i \dot{\boldsymbol{p}}_{C_i}^T \dot{\boldsymbol{p}}_{C_i} + \frac{1}{2} \boldsymbol{\omega}_i^T \bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i + k_{r,i+1} \dot{q}_{i+1} \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{\omega}_i \\ & + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 \boldsymbol{z}_{m_{i+1}}^T \boldsymbol{I}_{m_{i+1}} \boldsymbol{z}_{m_{i+1}},\end{aligned} \tag{7.64}$$

where $m_i = m_{\ell_i} + m_{m_{i+1}}$ and $\bar{\boldsymbol{I}}_i = \bar{\boldsymbol{I}}_{\ell_i} + \bar{\boldsymbol{I}}_{m_{i+1}}$ are respectively the overall mass and inertia tensor. In deriving (7.64), the relations in (7.57), (7.58) have been utilized as well as the following relation between the positions of the centres of mass:

$$m_{\ell_i} \boldsymbol{p}_{\ell_i} + m_{m_{i+1}} \boldsymbol{p}_{m_{i+1}} = m_i \boldsymbol{p}_{C_i}. \tag{7.65}$$

Notice that the first two terms on the right-hand side of (7.64) represent the kinetic energy contribution of the rotor when this is still, whereas the remaining two terms account for the rotor's own motion.

On the assumption that the rotor has a symmetric mass distribution about its axis of rotation, its inertia tensor expressed in a frame $\boldsymbol{R}_{m_i}$ with origin at the centre of mass and axis z_{m_i} aligned with the rotation axis can be written as

$$\boldsymbol{I}_{m_i}^{m_i} = \begin{bmatrix} I_{m_i xx} & 0 & 0 \\ 0 & I_{m_i yy} & 0 \\ 0 & 0 & I_{m_i zz} \end{bmatrix} \tag{7.66}$$

where $I_{m_i yy} = I_{m_i xx}$. As a consequence, the inertia tensor is invariant with respect to any rotation about axis z_{m_i} and is, anyhow, constant when referred to any frame attached to Link $i-1$.

Since the aim is to determine a set of dynamic parameters independent of the manipulator joint configuration, it is worth referring the inertia tensor of the link $\bar{\boldsymbol{I}}_i$ to frame $\boldsymbol{R}_i$ attached to the link and the inertia tensor $\boldsymbol{I}_{m_{i+1}}$ to frame $\boldsymbol{R}_{m_{i+1}}$ so that it is diagonal. In view of (7.66) one has

$$\boldsymbol{I}_{m_{i+1}} \boldsymbol{z}_{m_{i+1}} = \boldsymbol{R}_{m_{i+1}} \boldsymbol{I}_{m_{i+1}}^{m_{i+1}} \boldsymbol{R}_{m_{i+1}}^T \boldsymbol{z}_{m_{i+1}} = I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}} \tag{7.67}$$

where $I_{m_{i+1}} = I_{m_{i+1} zz}$ denotes the constant scalar moment of inertia of the rotor about its rotation axis.

Therefore, the kinetic energy (7.64) becomes

$$\begin{aligned}\mathcal{T}_i = {} & \frac{1}{2} m_i \dot{\boldsymbol{p}}_{C_i}^{iT} \dot{\boldsymbol{p}}_{C_i}^i + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \bar{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i \\ & + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 I_{m_{i+1}}.\end{aligned} \tag{7.68}$$

According to the linear velocity composition rule for Link i in (3.15), one may write

$$\dot{\boldsymbol{p}}_{C_i}^i = \dot{\boldsymbol{p}}_i^i + \boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i,C_i}^i, \tag{7.69}$$

where all the vectors have been referred to Frame i; note that $\boldsymbol{r}^i_{i,C_i}$ is fixed in such a frame. Substituting (7.69) into (7.68) gives

$$\mathcal{T}_i = \frac{1}{2} m_i \dot{\boldsymbol{p}}_i^{iT} \dot{\boldsymbol{p}}_i^i + \dot{\boldsymbol{p}}_i^{iT} \boldsymbol{S}(\boldsymbol{\omega}_i^i) m_i \boldsymbol{r}^i_{i,C_i} + \frac{1}{2} \boldsymbol{\omega}_i^{iT} \widehat{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}^{iT} \boldsymbol{\omega}_i^i + \frac{1}{2} k_{r,i+1}^2 \dot{q}_{i+1}^2 I_{m_{i+1}}, \tag{7.70}$$

where

$$\widehat{\boldsymbol{I}}_i^i = \bar{\boldsymbol{I}}_i^i + m_i \boldsymbol{S}^T(\boldsymbol{r}^i_{i,C_i}) \boldsymbol{S}(\boldsymbol{r}^i_{i,C_i}) \tag{7.71}$$

represents the inertia tensor with respect to the origin of Frame i according to Steiner theorem.

Let $\boldsymbol{r}^i_{i,C_i} = [\, \ell_{C_ix} \quad \ell_{C_iy} \quad \ell_{C_iz} \,]^T$. The *first moment of inertia* is

$$m_i \boldsymbol{r}^i_{i,C_i} = \begin{bmatrix} m_i \ell_{C_ix} \\ m_i \ell_{C_iy} \\ m_i \ell_{C_iz} \end{bmatrix}. \tag{7.72}$$

From (7.71) the inertia tensor of augmented Link i is

$$\widehat{\boldsymbol{I}}_i^i = \begin{bmatrix} \bar{I}_{ixx} + m_i(\ell_{C_iy}^2 + \ell_{C_iz}^2) & -\bar{I}_{ixy} - m_i \ell_{C_ix} \ell_{C_iy} & -\bar{I}_{ixz} - m_i \ell_{C_ix} \ell_{C_iz} \\ * & \bar{I}_{iyy} + m_i(\ell_{C_ix}^2 + \ell_{C_iz}^2) & -\bar{I}_{iyz} - m_i \ell_{C_iy} \ell_{C_iz} \\ * & * & \bar{I}_{izz} + m_i(\ell_{C_ix}^2 + \ell_{C_iy}^2) \end{bmatrix}$$
$$= \begin{bmatrix} \widehat{I}_{ixx} & -\widehat{I}_{ixy} & -\widehat{I}_{ixz} \\ * & \widehat{I}_{iyy} & -\widehat{I}_{iyz} \\ * & * & \widehat{I}_{izz} \end{bmatrix}. \tag{7.73}$$

Therefore, the kinetic energy of the augmented link is linear with respect to the dynamic parameters, namely, the *mass*, the *three components of the first moment of inertia* in (7.72), the *six components of the inertia tensor* in (7.73), and the *moment of inertia of the rotor*.

As regards potential energy, it is worth referring to the centre of mass of augmented Link i defined as in (7.65), and thus the single contribution of potential energy can be written as

$$\mathcal{U}_i = -m_i \boldsymbol{g}_0^{iT} \boldsymbol{p}^i_{C_i} \tag{7.74}$$

where the vectors have been referred to Frame i. According to the relation

$$\boldsymbol{p}^i_{C_i} = \boldsymbol{p}^i_i + \boldsymbol{r}^i_{i,C_i}.$$

The expression in (7.74) can be rewritten as

$$\mathcal{U}_i = -\boldsymbol{g}_0^{iT} (m_i \boldsymbol{p}^i_i + m_i \boldsymbol{r}^i_{i,C_i}) \tag{7.75}$$

that is, the potential energy of the augmented link is linear with respect to the mass and the three components of the first moment of inertia in (7.72).

By summing the contributions of kinetic energy and potential energy for all augmented links, the Lagrangian of the system (7.1) can be expressed in the form

$$\mathcal{L} = \sum_{i=1}^{n} (\boldsymbol{\beta}_{\mathcal{T}i}^T - \boldsymbol{\beta}_{\mathcal{U}i}^T) \boldsymbol{\pi}_i \tag{7.76}$$

where $\boldsymbol{\pi}_i$ is the (11×1) vector of dynamic parameters

$$\boldsymbol{\pi}_i = [\, m_i \;\; m_i \ell_{C_ix} \;\; m_i \ell_{C_iy} \;\; m_i \ell_{C_iz} \;\; \widehat{I}_{ixx} \;\; \widehat{I}_{ixy} \;\; \widehat{I}_{ixz} \;\; \widehat{I}_{iyy} \;\; \widehat{I}_{iyz} \;\; \widehat{I}_{izz} \;\; I_{m_i} \,]^T, \tag{7.77}$$

in which the moment of inertia of Rotor i has been associated with the parameters of Link i so as to simplify the notation.

In (7.76), $\boldsymbol{\beta}_{\mathcal{T}i}$ and $\boldsymbol{\beta}_{\mathcal{U}i}$ are two (11×1) vectors that allow the Lagrangian to be written as a function of $\boldsymbol{\pi}_i$. Such vectors are a function of the generalized coordinates of the mechanical system (and also of their derivatives as regards $\boldsymbol{\beta}_{\mathcal{T}i}$). In particular, it can be shown that $\boldsymbol{\beta}_{\mathcal{T}i} = \boldsymbol{\beta}_{\mathcal{T}i}(q_1, q_2, \ldots, q_i, \dot{q}_1, \dot{q}_2, \ldots, \dot{q}_i)$ and $\boldsymbol{\beta}_{\mathcal{U}i} = \boldsymbol{\beta}_{\mathcal{U}i}(q_1, q_2, \ldots, q_i)$, i.e., they do not depend on the variables of the joints subsequent to Link i.

At this point, it should be observed how the derivations required by the Lagrange equations in (7.2) do not alter the property of linearity in the parameters, and then the generalized force at Joint i can be written as

$$\xi_i = \sum_{j=1}^{n} \boldsymbol{y}_{ij}^T \boldsymbol{\pi}_j \tag{7.78}$$

where

$$\boldsymbol{y}_{ij} = \frac{d}{dt} \frac{\partial \boldsymbol{\beta}_{\mathcal{T}j}}{\partial \dot{q}_i} - \frac{\partial \boldsymbol{\beta}_{\mathcal{T}j}}{\partial q_i} + \frac{\partial \boldsymbol{\beta}_{\mathcal{U}j}}{\partial q_i}. \tag{7.79}$$

Since the partial derivatives of $\boldsymbol{\beta}_{\mathcal{T}j}$ and $\boldsymbol{\beta}_{\mathcal{U}j}$ appearing in (7.79) vanish for $j < i$, the following notable result is obtained:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_{11}^T & \boldsymbol{y}_{12}^T & \cdots & \boldsymbol{y}_{1n}^T \\ \boldsymbol{0}^T & \boldsymbol{y}_{22}^T & \cdots & \boldsymbol{y}_{2n}^T \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}^T & \boldsymbol{0}^T & \cdots & \boldsymbol{y}_{nn}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \vdots \\ \boldsymbol{\pi}_n \end{bmatrix} \tag{7.80}$$

which yields the property of *linearity of the model* of a manipulator with respect to a suitable set of *dynamic parameters*.

In the simple case of no contact forces ($\boldsymbol{h}_e = \boldsymbol{0}$), it may be worth including the viscous friction coefficient F_{vi} and Coulomb friction coefficient F_{si} in the parameters of the vector $\boldsymbol{\pi}_i$, thus leading to a total number of 13 parameters per joint. In summary, (7.80) can be compactly written as

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \boldsymbol{\pi} \tag{7.81}$$

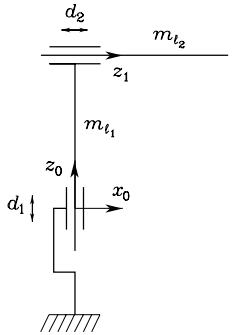

Fig. 7.3. Two-link Cartesian arm

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 matrix is usually called *regressor*. Regarding the dimension of the parameter vector, notice that $p \leq 13n$ since not all the thirteen parameters for each joint may explicitly appear in (7.81).

7.3 Dynamic Model of Simple Manipulator Structures

In the following, three examples of dynamic model computation are illustrated for simple two-DOF manipulator structures. Two DOFs, in fact, are enough to understand the physical meaning of all dynamic terms, especially the joint coupling terms. On the other hand, dynamic model computation for manipulators with more DOFs would be quite tedious and prone to errors, when carried out by paper and pencil. In those cases, it is advisable to perform it with the aid of a symbolic programming software package.

7.3.1 Two-link Cartesian Arm

Consider the two-link Cartesian arm in Fig. 7.3, for which the vector of generalized coordinates is $\boldsymbol{q} = [\, d_1 \quad d_2 \,]^T$. Let m_{ℓ_1}, m_{ℓ_2} be the masses of the two links, and m_{m_1}, m_{m_2} the masses of the rotors of the two joint motors. Also let I_{m_1}, I_{m_2} be the moments of inertia with respect to the axes of the two rotors. It is assumed that $\boldsymbol{p}_{m_i} = \boldsymbol{p}_{i-1}$ and $\boldsymbol{z}_{m_i} = \boldsymbol{z}_{i-1}$, for $i = 1, 2$, i.e., the motors are located on the joint axes with centres of mass located at the origins of the respective frames.

With the chosen coordinate frames, computation of the Jacobians in (7.16), (7.18) yields

$$\boldsymbol{J}_P^{(\ell_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad \boldsymbol{J}_P^{(\ell_2)} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Obviously, in this case there are no angular velocity contributions for both links.

Computation of the Jacobians in (7.25), (7.26) e (7.28), (7.29) yields

$$\boldsymbol{J}_P^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{J}_P^{(m_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\boldsymbol{J}_O^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{r1} & 0 \end{bmatrix} \qquad \boldsymbol{J}_O^{(m_2)} = \begin{bmatrix} 0 & k_{r2} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where k_{ri} is the gear reduction ratio of Motor i. It is obvious to see that $\boldsymbol{z}_1 = [\, 1 \quad 0 \quad 0 \,]^T$, which greatly simplifies computation of the second term in (4.34).

From (7.32), the inertia matrix is

$$\boldsymbol{B} = \begin{bmatrix} m_{\ell_1} + m_{m_2} + k_{r1}^2 I_{m_1} + m_{\ell_2} & 0 \\ 0 & m_{\ell_2} + k_{r2}^2 I_{m_2} \end{bmatrix}.$$

It is worth observing that $\boldsymbol{B}$ is *constant*, i.e., it does not depend on the arm configuration. This implies also that $\boldsymbol{C} = \boldsymbol{O}$, i.e., there are no contributions of centrifugal and Coriolis forces. As for the gravitational terms, since $\boldsymbol{g}_0 = [\, 0 \quad 0 \quad -g \,]^T$ (g is gravity acceleration), (7.39) with the above Jacobians gives

$$g_1 = (m_{\ell_1} + m_{m_2} + m_{\ell_2})g \qquad g_2 = 0.$$

In the absence of friction and tip contact forces, the resulting equations of motion are

$$\begin{aligned} (m_{\ell_1} + m_{m_2} + k_{r1}^2 I_{m_1} + m_{\ell_2})\ddot{d}_1 + (m_{\ell_1} + m_{m_2} + m_{\ell_2})g &= \tau_1 \\ (m_{\ell_2} + k_{r2}^2 I_{m_2})\ddot{d}_2 &= \tau_2 \end{aligned}$$

where τ_1 and τ_2 denote the forces applied to the two joints. Notice that a completely decoupled dynamics has been obtained. This is a consequence not only of the Cartesian structures but also of the particular geometry; in other words, if the second joint axis were not at a right angle with the first joint axis, the resulting inertia matrix would not be diagonal (see Problem 7.1).

7.3.2 Two-link Planar Arm

Consider the two-link planar arm in Fig. 7.4, for which the vector of generalized coordinates is $\boldsymbol{q} = [\, \vartheta_1 \quad \vartheta_2 \,]^T$. Let ℓ_1, ℓ_2 be the distances of the centres of mass of the two links from the respective joint axes. Also let m_{ℓ_1}, m_{ℓ_2} be the masses of the two links, and m_{m_1}, m_{m_2} the masses of the rotors of the two joint motors. Finally, let I_{m_1}, I_{m_2} be the moments of inertia with respect to the axes of the two rotors, and I_{ℓ_1}, I_{ℓ_2} the moments of inertia relative to the

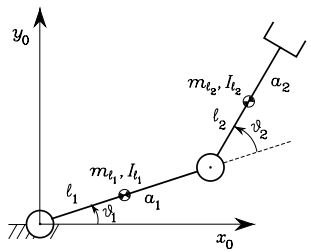

Fig. 7.4. Two-link planar arm

centres of mass of the two links, respectively. It is assumed that $\boldsymbol{p}_{m_i} = \boldsymbol{p}_{i-1}$ and $\boldsymbol{z}_{m_i} = \boldsymbol{z}_{i-1}$, for $i = 1, 2$, i.e., the motors are located on the joint axes with centres of mass located at the origins of the respective frames.

With the chosen coordinate frames, computation of the Jacobians in (7.16), (7.18) yields

$$\boldsymbol{J}_P^{(\ell_1)} = \begin{bmatrix} -\ell_1 s_1 & 0 \\ \ell_1 c_1 & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{J}_P^{(\ell_2)} = \begin{bmatrix} -a_1 s_1 - \ell_2 s_{12} & -\ell_2 s_{12} \\ a_1 c_1 + \ell_2 c_{12} & \ell_2 c_{12} \\ 0 & 0 \end{bmatrix},$$

whereas computation of the Jacobians in (7.17), (7.19) yields

$$\boldsymbol{J}_O^{(\ell_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad \boldsymbol{J}_O^{(\ell_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}.$$

Notice that $\boldsymbol{\omega}_i$, for $i = 1, 2$, is aligned with z_0, and thus $\boldsymbol{R}_i$ has *no* effect. It is then possible to refer to the scalar moments of inertia I_{ℓ_i}.

Computation of the Jacobians in (7.25), (7.26) yields

$$\boldsymbol{J}_P^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad \boldsymbol{J}_P^{(m_2)} = \begin{bmatrix} -a_1 s_1 & 0 \\ a_1 c_1 & 0 \\ 0 & 0 \end{bmatrix},$$

whereas computation of the Jacobians in (7.28), (7.29) yields

$$\boldsymbol{J}_O^{(m_1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ k_{r1} & 0 \end{bmatrix} \qquad \boldsymbol{J}_O^{(m_2)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & k_{r2} \end{bmatrix}$$

where k_{ri} is the gear reduction ratio of Motor i.

From (7.32), the inertia matrix is

$$\boldsymbol{B}(\boldsymbol{q}) = \begin{bmatrix} b_{11}(\vartheta_2) & b_{12}(\vartheta_2) \\ b_{21}(\vartheta_2) & b_{22} \end{bmatrix}$$

$$\begin{aligned} b_{11} &= I_{\ell_1} + m_{\ell_1}\ell_1^2 + k_{r1}^2 I_{m_1} + I_{\ell_2} + m_{\ell_2}(a_1^2 + \ell_2^2 + 2a_1\ell_2 c_2) \\ &\quad + I_{m_2} + m_{m_2} a_1^2 \\ b_{12} &= b_{21} = I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2} I_{m_2} \\ b_{22} &= I_{\ell_2} + m_{\ell_2}\ell_2^2 + k_{r2}^2 I_{m_2}. \end{aligned}$$

Compared to the previous example, the inertia matrix is now configuration-dependent. Notice that the term $k_{r2} I_{m_2}$ in the off-diagonal term of the inertia matrix derives from having considered the rotational part of the motor kinetic energy as due to the total angular velocity, i.e., its own angular velocity and that of the preceding link in the kinematic chain. At first approximation, especially in the case of high values of the gear reduction ratio, this contribution could be neglected; in the resulting reduced model, motor inertias would appear uniquely in the elements on the diagonal of the inertia matrix with terms of the type $k_{ri}^2 I_{m_i}$.

The computation of Christoffel symbols as in (7.45) gives

$$\begin{aligned} c_{111} &= \frac{1}{2}\frac{\partial b_{11}}{\partial q_1} = 0 \\ c_{112} &= c_{121} = \frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = -m_{\ell_2} a_1 \ell_2 s_2 = h \\ c_{122} &= \frac{\partial b_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial b_{22}}{\partial q_1} = h \\ c_{211} &= \frac{\partial b_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial b_{11}}{\partial q_2} = -h \\ c_{212} &= c_{221} = \frac{1}{2}\frac{\partial b_{22}}{\partial q_1} = 0 \\ c_{222} &= \frac{1}{2}\frac{\partial b_{22}}{\partial q_2} = 0, \end{aligned}$$

leading to the matrix

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} h\dot{\vartheta}_2 & h(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ -h\dot{\vartheta}_1 & 0 \end{bmatrix}.$$

Computing the matrix $\boldsymbol{N}$ in (7.47) gives

$$\begin{aligned} \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \dot{\boldsymbol{B}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ &= \begin{bmatrix} 2h\dot{\vartheta}_2 & h\dot{\vartheta}_2 \\ h\dot{\vartheta}_2 & 0 \end{bmatrix} - 2\begin{bmatrix} h\dot{\vartheta}_2 & h(\dot{\vartheta}_1 + \dot{\vartheta}_2) \\ -h\dot{\vartheta}_1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -2h\dot{\vartheta}_1 - h\dot{\vartheta}_2 \\ 2h\dot{\vartheta}_1 + h\dot{\vartheta}_2 & 0 \end{bmatrix} \end{aligned}$$

that allows the verification of the skew-symmetry property expressed by (7.48). See also Problem 7.2.

As for the gravitational terms, since $\boldsymbol{g}_0 = [\,0 \quad -g \quad 0\,]^T$, (7.39) with the above Jacobians gives

$$\begin{aligned} g_1 &= (m_{\ell_1}\ell_1 + m_{m_2}a_1 + m_{\ell_2}a_1)gc_1 + m_{\ell_2}\ell_2 g c_{12} \\ g_2 &= m_{\ell_2}\ell_2 g c_{12}. \end{aligned}$$

In the absence of friction and tip contact forces, the resulting equations of motion are

$$\begin{aligned} &\left(I_{\ell_1} + m_{\ell_1}\ell_1^2 + k_{r1}^2 I_{m_1} + I_{\ell_2} + m_{\ell_2}(a_1^2 + \ell_2^2 + 2a_1\ell_2 c_2) + I_{m_2} + m_{m_2}a_1^2\right)\ddot{\vartheta}_1 \\ &\quad + \left(I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2}I_{m_2}\right)\ddot{\vartheta}_2 \\ &\quad -2m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_1\dot{\vartheta}_2 - m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_2^2 \\ &\quad +(m_{\ell_1}\ell_1 + m_{m_2}a_1 + m_{\ell_2}a_1)gc_1 + m_{\ell_2}\ell_2 g c_{12} = \tau_1 \\ &\left(I_{\ell_2} + m_{\ell_2}(\ell_2^2 + a_1\ell_2 c_2) + k_{r2}I_{m_2}\right)\ddot{\vartheta}_1 + \left(I_{\ell_2} + m_{\ell_2}\ell_2^2 + k_{r2}^2 I_{m_2}\right)\ddot{\vartheta}_2 \\ &\quad +m_{\ell_2}a_1\ell_2 s_2\dot{\vartheta}_1^2 + m_{\ell_2}\ell_2 g c_{12} = \tau_2 \end{aligned} \tag{7.82}$$

where τ_1 and τ_2 denote the torques applied to the joints.

Finally, it is wished to derive a parameterization of the dynamic model (7.82) according to the relation (7.81). By direct inspection of the expressions of the joint torques, it is possible to find the following parameter vector:

$$\boldsymbol{\pi} = [\,\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4 \quad \pi_5 \quad \pi_6 \quad \pi_7 \quad \pi_8\,]^T \tag{7.83}$$

$$\begin{aligned} \pi_1 &= m_1 = m_{\ell_1} + m_{m_2} \\ \pi_2 &= m_1\ell_{C_1} = m_{\ell_1}(\ell_1 - a_1) \\ \pi_3 &= \widehat{I}_1 = I_{\ell_1} + m_{\ell_1}(\ell_1 - a_1)^2 + I_{m_2} \\ \pi_4 &= I_{m_1} \\ \pi_5 &= m_2 = m_{\ell_2} \\ \pi_6 &= m_2\ell_{C_2} = m_{\ell_2}(\ell_2 - a_2) \\ \pi_7 &= \widehat{I}_2 = I_{\ell_2} + m_{\ell_2}(\ell_2 - a_2)^2 \\ \pi_8 &= I_{m_2}, \end{aligned}$$

where the parameters for the augmented links have been found according to (7.77). It can be recognized that the number of non-null parameters is less than the maximum number of twenty-two parameters allowed in this case.[6] The regressor in (7.81) is

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & y_{17} & y_{18} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} & y_{26} & y_{27} & y_{28} \end{bmatrix} \tag{7.84}$$

[6] The number of parameters can be further reduced by resorting to a more accurate inspection, which leads to finding a minimum number of five parameters; those turn out to be a linear combination of the parameters in (7.83) (see Problem 7.4).

$$\begin{aligned} y_{11} &= a_1^2\ddot{\vartheta}_1 + a_1 g c_1 \\ y_{12} &= 2a_1\ddot{\vartheta}_1 + g c_1 \\ y_{13} &= \ddot{\vartheta}_1 \\ y_{14} &= k_{r1}^2\ddot{\vartheta}_1 \\ y_{15} &= (a_1^2 + 2a_1a_2c_2 + a_2^2)\ddot{\vartheta}_1 + (a_1a_2c_2 + a_2^2)\ddot{\vartheta}_2 - 2a_1a_2s_2\dot{\vartheta}_1\dot{\vartheta}_2 \\ &\quad -a_1a_2s_2\dot{\vartheta}_2^2 + a_1 g c_1 + a_2 g c_{12} \\ y_{16} &= (2a_1c_2 + 2a_2)\ddot{\vartheta}_1 + (a_1c_2 + 2a_2)\ddot{\vartheta}_2 - 2a_1s_2\dot{\vartheta}_1\dot{\vartheta}_2 - a_1s_2\dot{\vartheta}_2^2 \\ &\quad +g c_{12} \\ y_{17} &= \ddot{\vartheta}_1 + \ddot{\vartheta}_2 \\ y_{18} &= k_{r2}\ddot{\vartheta}_2 \end{aligned}$$

$$\begin{aligned} y_{21} &= 0 \\ y_{22} &= 0 \\ y_{23} &= 0 \\ y_{24} &= 0 \\ y_{25} &= (a_1a_2c_2 + a_2^2)\ddot{\vartheta}_1 + a_2^2\ddot{\vartheta}_2 + a_1a_2s_2\dot{\vartheta}_1^2 + a_2 g c_{12} \\ y_{26} &= (a_1c_2 + 2a_2)\ddot{\vartheta}_1 + 2a_2\ddot{\vartheta}_2 + a_1s_2\dot{\vartheta}_1^2 + g c_{12} \\ y_{27} &= \ddot{\vartheta}_1 + \ddot{\vartheta}_2 \\ y_{28} &= k_{r2}\ddot{\vartheta}_1 + k_{r2}^2\ddot{\vartheta}_2. \end{aligned}$$

Example 7.2

In order to understand the relative weight of the various torque contributions in the dynamic model (7.82), consider a two-link planar arm with the following data:

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

$$k_{r1} = k_{r2} = 100 \quad m_{m_1} = m_{m_2} = 5\,\text{kg} \quad I_{m_1} = I_{m_2} = 0.01\,\text{kg}\cdot\text{m}^2.$$

The two links have been chosen equal to illustrate better the dynamic interaction between the two joints.

Figure 7.5 shows the time history of positions, velocities, accelerations and torques resulting from joint trajectories with typical triangular velocity profile and equal time duration. The initial arm configuration is so that the tip is located at the point $(0.2, 0)\,\text{m}$ with a lower elbow posture. Both joints make a rotation of $\pi/2\,\text{rad}$ in a time of $0.5\,\text{s}$.

From the time history of the single torque contributions in Fig. 7.6 it can be recognized that:

- The inertia torque at Joint 1 due to Joint 1 acceleration follows the time history of the acceleration.
- The inertia torque at Joint 2 due to Joint 2 acceleration is piecewise constant, since the inertia moment at Joint 2 axis is constant.

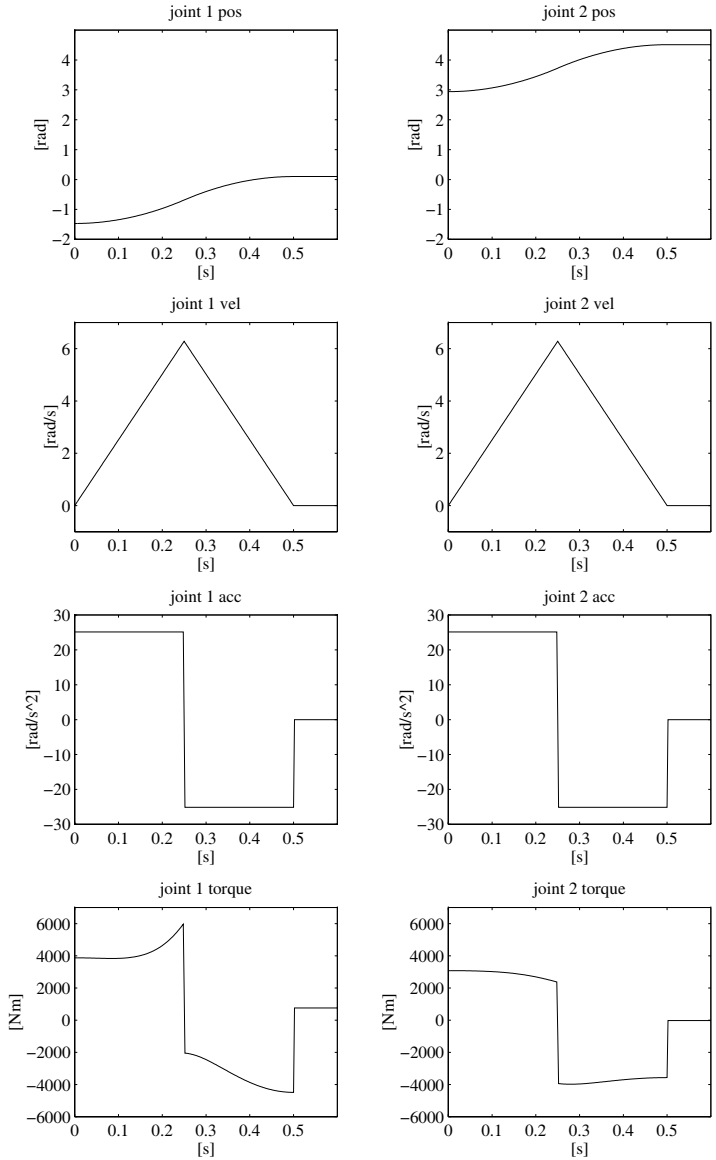

Fig. 7.5. Time history of positions, velocities, accelerations and torques with joint trajectories of equal duration

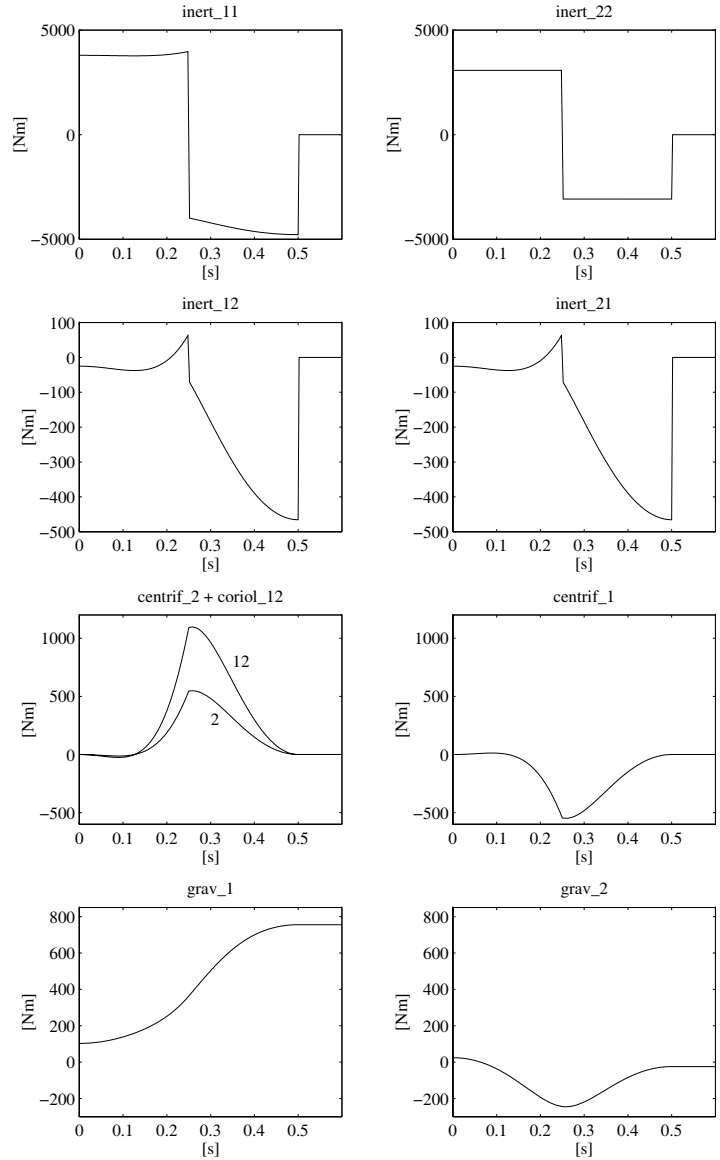

Fig. 7.6. Time history of torque contributions with joint trajectories of equal duration

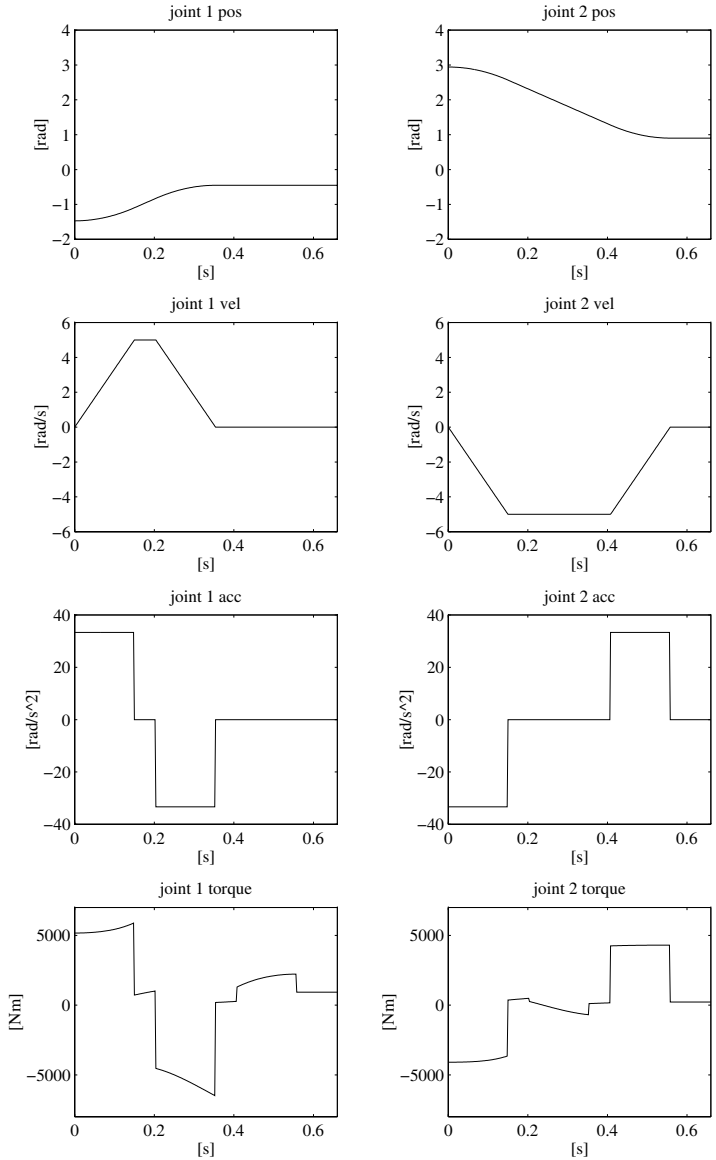

Fig. 7.7. Time history of positions, velocities, accelerations and torques with joint trajectories of different duration

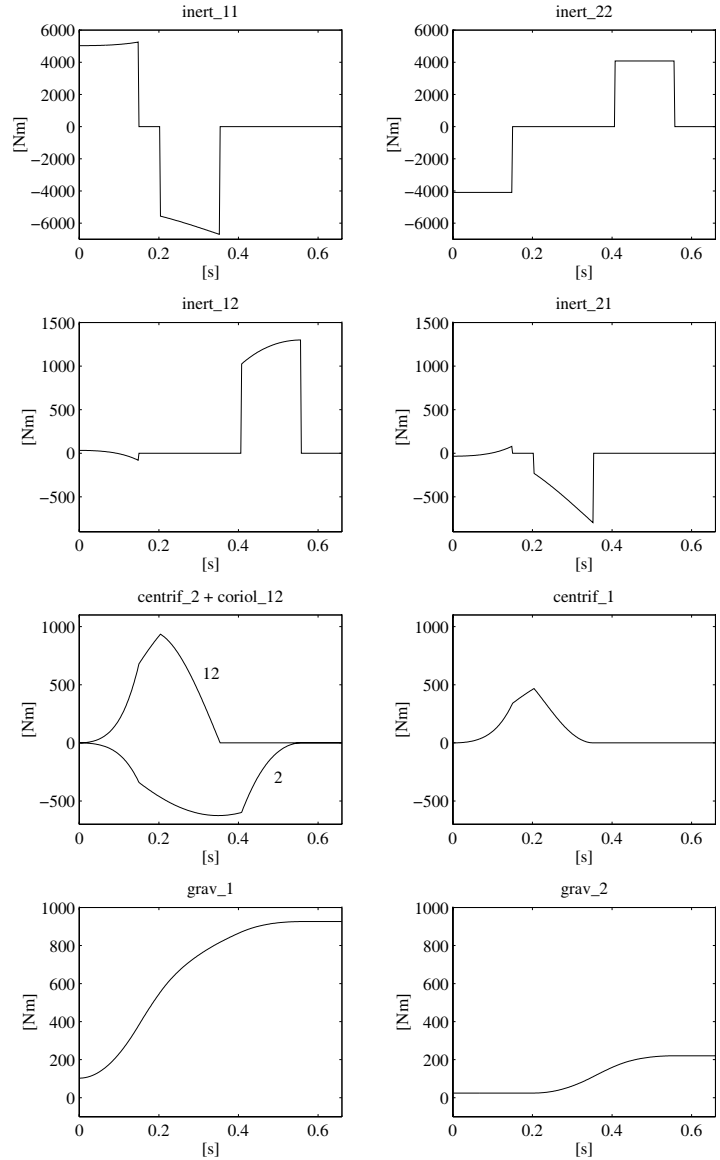

Fig. 7.8. Time history of torque contributions with joint trajectories of different duration

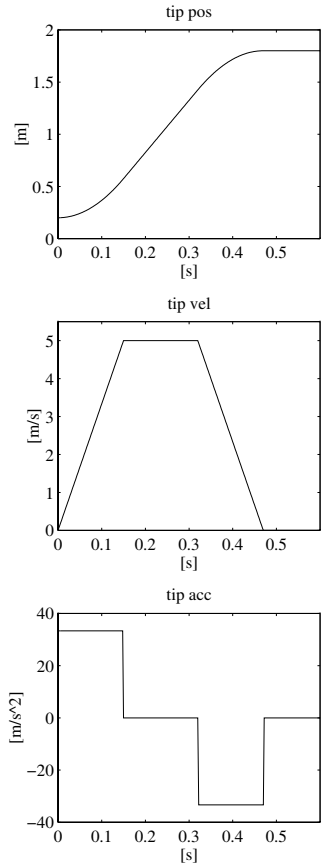

Fig. 7.9. Time history of tip position, velocity and acceleration with a straight line tip trajectory along the horizontal axis

- The inertia torques at each joint due to acceleration of the other joint confirm the symmetry of the inertia matrix, since the acceleration profiles are the same for both joints.
- The Coriolis effect is present only at Joint 1, since the arm tip moves with respect to the mobile frame attached to Link 1 but is fixed with respect to the frame attached to Link 2.
- The centrifugal and Coriolis torques reflect the above symmetry.

Figure 7.7 shows the time history of positions, velocities, accelerations and torques resulting from joint trajectories with typical trapezoidal velocity profile and different time duration. The initial configuration is the same as in the previous case. The two joints make a rotation so as to take the tip to the point $(1.8, 0)$ m. The acceleration time is 0.15 s and the maximum velocity is 5 rad/s for both joints.

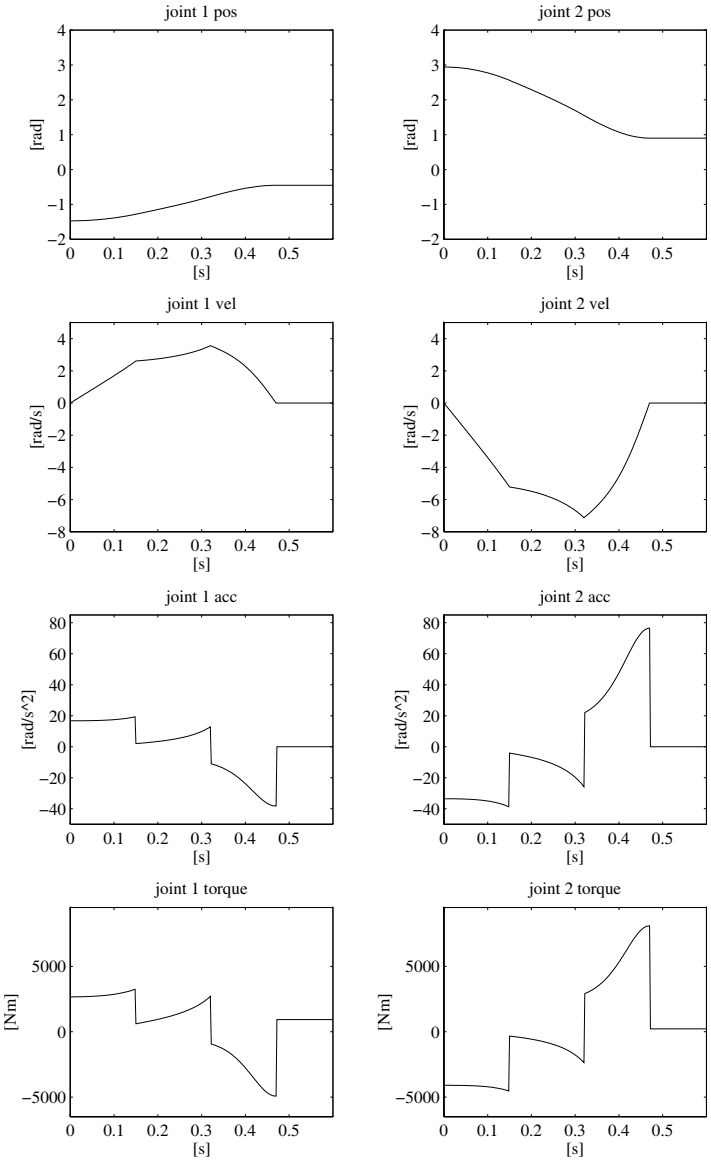

Fig. 7.10. Time history of joint positions, velocities, accelerations, and torques with a straight line tip trajectory along the horizontal axis

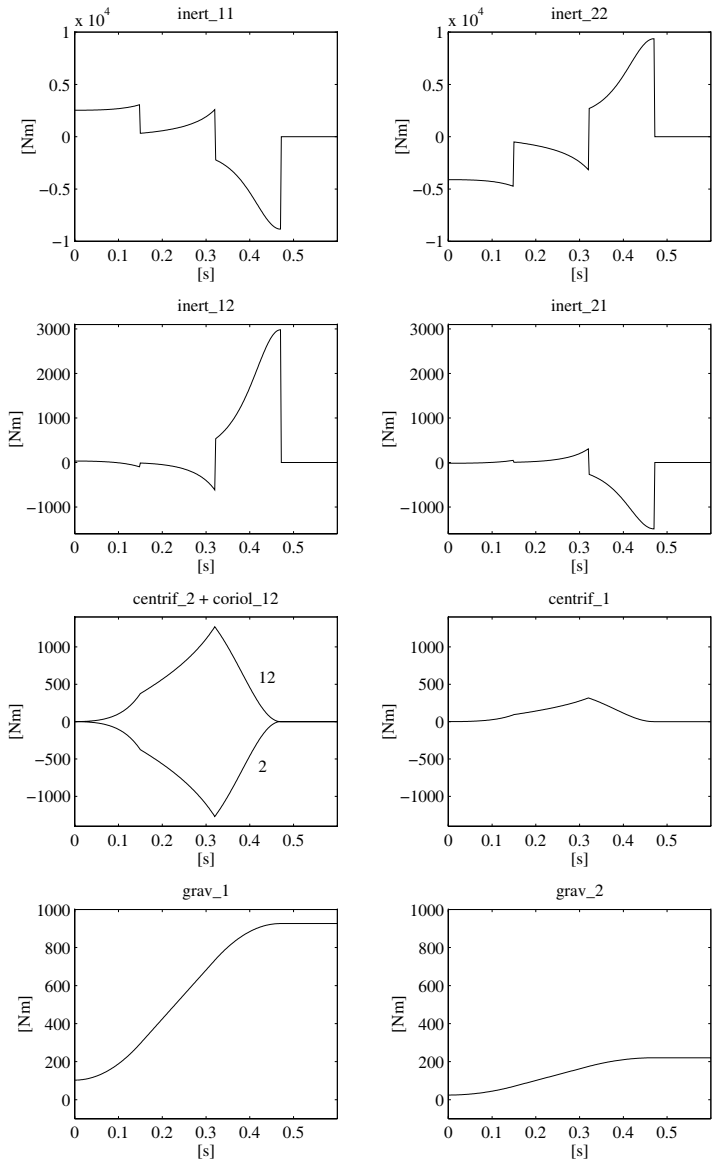

Fig. 7.11. Time history of joint torque contributions with a straight line tip trajectory along the horizontal axis

From the time history of the single torque contributions in Fig. 7.8 it can be recognized that:

- The inertia torque at Joint 1 due to Joint 2 acceleration is opposite to that at Joint 2 due to Joint 1 acceleration in that portion of trajectory when the two accelerations have the same magnitude but opposite sign.
- The different velocity profiles imply that the centrifugal effect induced at Joint 1 by Joint 2 velocity dies out later than the centrifugal effect induced at Joint 2 by Joint 1 velocity.
- The gravitational torque at Joint 2 is practically constant in the first portion of the trajectory, since Link 2 is almost kept in the same posture. As for the gravitational torque at Joint 1, instead, the centre of mass of the articulated system moves away from the origin of the axes.

Finally, Fig. 7.9 shows the time history of tip position, velocity and acceleration for a trajectory with a trapezoidal velocity profile. Starting from the same initial posture as above, the arm tip makes a translation of 1.6 m along the horizontal axis; the acceleration time is 0.15 s and the maximum velocity is 5 m/s.

As a result of an inverse kinematics procedure, the time history of joint positions, velocities and accelerations have been computed which are illustrated in Fig. 7.10, together with the joint torques that are needed to execute the assigned trajectory. It can be noticed that the time history of the represented quantities differs from the corresponding ones in the operational space, in view of the nonlinear effects introduced by kinematic relations.

For what concerns the time history of the individual torque contributions in Fig. 7.11, it is possible to make a number of remarks similar to those made above for trajectories assigned directly in the joint space.

7.3.3 Parallelogram Arm

Consider the parallelogram arm in Fig. 7.12. Because of the presence of the closed chain, the equivalent tree-structured open-chain arm is initially taken into account. Let $\ell_{1'}$, $\ell_{2'}$, $\ell_{3'}$ and $\ell_{1''}$ be the distances of the centres of mass of the three links along one branch of the tree, and of the single link along the other branch, from the respective joint axes. Also let $m_{\ell_{1'}}$, $m_{\ell_{2'}}$, $m_{\ell_{3'}}$ and $m_{\ell_{1''}}$ be the masses of the respective links, and $I_{\ell_{1'}}$, $I_{\ell_{2'}}$, $I_{\ell_{3'}}$ and $I_{\ell_{1''}}$ the moments of inertia relative to the centres of mass of the respective links. For the sake of simplicity, the contributions of the motors are neglected.

With the chosen coordinate frames, computation of the Jacobians in (7.16) (7.18) yields

$$\boldsymbol{J}_P^{(\ell_{1'})} = \begin{bmatrix} -\ell_{1'}s_{1'} & 0 & 0 \\ \ell_{1'}c_{1'} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{J}_P^{(\ell_{2'})} = \begin{bmatrix} -a_{1'}s_{1'} - \ell_2 s_{1'2'} & -\ell_{2'}s_{1'2'} & 0 \\ a_{1'}c_{1'} + \ell_{2'}c_{1'2'} & \ell_2 c_{1'2'} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{J}_P^{(\ell_{3'})} = \begin{bmatrix} -a_{1'}s_{1'} - a_{2'}s_{1'2'} - \ell_{3'}s_{1'2'3'} & -a_{2'}s_{1'2'} - \ell_{3'}s_{1'2'3'} & -\ell_{3'}s_{1'2'3'} \\ a_{1'}c_{1'} + a_{2'}c_{1'2'} + \ell_{3'}c_{1'2'3'} & a_{2'}c_{1'2'} + \ell_{3'}c_{1'2'3'} & \ell_{3'}c_{1'2'3'} \\ 0 & 0 & 0 \end{bmatrix}$$

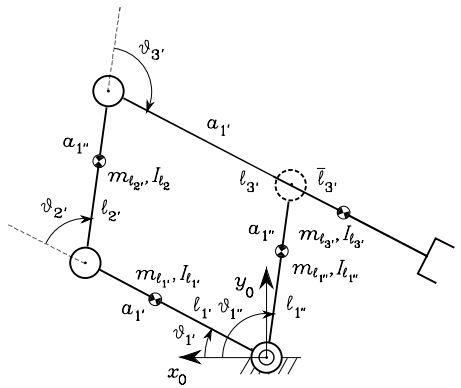

Fig. 7.12. Parallelogram arm

and

$$\boldsymbol{J}_P^{(\ell_{1''})} = \begin{bmatrix} -\ell_{1''} s_{1''} \\ \ell_{1''} c_{1''} \\ 0 \end{bmatrix},$$

whereas computation of the Jacobians in (7.17), (7.19) yields

$$\boldsymbol{J}_O^{(\ell_{1'})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{J}_O^{(\ell_{2'})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{J}_O^{(\ell_{3'})} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

and

$$\boldsymbol{J}_O^{(\ell_{1''})} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

From (7.32), the inertia matrix of the virtual arm composed of joints $\vartheta_{1'}$, $\vartheta_{2'}$, $\vartheta_{3'}$ is

$$\boldsymbol{B}'(\boldsymbol{q}') = \begin{bmatrix} b_{1'1'}(\vartheta_{2'}, \vartheta_{3'}) & b_{1'2'}(\vartheta_{2'}, \vartheta_{3'}) & b_{1'3'}(\vartheta_{2'}, \vartheta_{3'}) \\ b_{2'1'}(\vartheta_{2'}, \vartheta_{3'}) & b_{2'2'}(\vartheta_{3'}) & b_{2'3'}(\vartheta_{3'}) \\ b_{3'1'}(\vartheta_{2'}, \vartheta_{3'}) & b_{3'2'}(\vartheta_{3'}) & b_{3'3'} \end{bmatrix}$$

$$\begin{aligned}
b_{1'1'} &= I_{\ell_{1'}} + m_{\ell_{1'}} \ell_{1'}^2 + I_{\ell_{2'}} + m_{\ell_{2'}} (a_{1'}^2 + \ell_{2'}^2 + 2a_{1'} \ell_{2'} c_{2'}) + I_{\ell_{3'}} \\
&\quad + m_{\ell_{3'}} (a_{1'}^2 + a_{2'}^2 + \ell_{3'}^2 + 2a_{1'} a_{2'} c_{2'} + 2a_{1'} \ell_{3'} c_{2'3'} + 2a_{2'} \ell_{3'} c_{3'}) \\
b_{1'2'} &= b_{2'1'} = I_{\ell_{2'}} + m_{\ell_{2'}} (\ell_{2'}^2 + a_{1'} \ell_{2'} c_{2'}) + I_{\ell_{3'}} \\
&\quad + m_{\ell_{3'}} (a_{2'}^2 + \ell_{3'}^2 + a_{1'} a_{2'} c_{2'} + a_{1'} \ell_{3'} c_{2'3'} + 2a_{2'} \ell_{3'} c_{3'}) \\
b_{1'3'} &= b_{31} = I_{\ell_{3'}} + m_{\ell_{3'}} (\ell_{3'}^2 + a_{1'} \ell_{3'} c_{2'3'} + a_{2'} \ell_{3'} c_{3'}) \\
b_{2'2'} &= I_{\ell_{2'}} + m_{\ell_{2'}} \ell_{2'}^2 + I_{\ell_{3'}} + m_{\ell_{3'}} (a_{2'}^2 + \ell_{3'}^2 + 2a_{2'} \ell_{3'} c_{3'}) \\
b_{2'3'} &= I_{\ell_{3'}} + m_{\ell_{3'}} (\ell_{3'}^2 + a_{2'} \ell_{3'} c_{3'}) \\
b_{3'3'} &= I_{\ell_{3'}} + m_{\ell_{3'}} \ell_{3'}^2
\end{aligned}$$

while the moment of inertia of the virtual arm composed of just joint $\vartheta_{1''}$ is

$$b_{1''1''} = I_{\ell_{1''}} + m_{\ell_{1''}} \ell_{1''}^2.$$

Therefore, the inertial torque contributions of the two virtual arms are respectively:

$$\tau_{i'} = \sum_{j'=1'}^{3'} b_{i'j'} \ddot{\vartheta}_{j'} \qquad \tau_{1''} = b_{1''1''} \ddot{\vartheta}_{1''}.$$

At this point, in view of (2.64) and (3.121), the inertial torque contributions at the actuated joints for the closed-chain arm turn out to be

$$\boldsymbol{\tau}_a = \boldsymbol{B}_a \ddot{\boldsymbol{q}}_a$$

where $\boldsymbol{q}_a = [\, \vartheta_{1'} \quad \vartheta_{1''} \,]^T$, $\boldsymbol{\tau}_a = [\, \tau_{a1} \quad \tau_{a2} \,]^T$ and

$$\boldsymbol{B}_a = \begin{bmatrix} b_{a11} & b_{a12} \\ b_{a21} & b_{a22} \end{bmatrix}$$

$$\begin{aligned}
b_{a11} &= I_{\ell_{1'}} + m_{\ell_{1'}} \ell_{1'}^2 + m_{\ell_{2'}} a_{1'}^2 + I_{\ell_{3'}} + m_{\ell_{3'}} \ell_{3'}^2 + m_{\ell_{3'}} a_{1'}^2 \\
&\quad - 2a_{1'} m_{\ell_{3'}} \ell_{3'} \\
b_{a12} &= b_{a21} = \left(a_{1'} m_{\ell_{2'}} \ell_{2'} + a_{1''} m_{\ell_{3'}} (a_{1'} - \ell_{3'}) \right) \cos(\vartheta_{1''} - \vartheta_{1'}) \\
b_{a22} &= I_{\ell_{1'}} + m_{\ell_{1'}} \ell_{1'}^2 + I_{\ell_{2'}} + m_{\ell_{2'}} \ell_{2'}^2 + m_{\ell_{3'}} a_{1''}^2.
\end{aligned}$$

This expression reveals the possibility of obtaining a *configuration-independent* and *decoupled* inertia matrix; to this end it is sufficient to design the four links of the parallelogram so that

$$\frac{m_{\ell_{3'}} \bar{\ell}_{3'}}{m_{\ell_{2'}} \ell_{2'}} = \frac{a_{1'}}{a_{1''}}$$

where $\bar{\ell}_{3'} = \ell_{3'} - a_{1'}$ is the distance of the centre of mass of Link 3′ from the axis of Joint 4. If this condition is satisfied, then the inertia matrix is diagonal ($b_{a12} = b_{a21} = 0$) with

$$\begin{aligned}
b_{a11} &= I_{\ell_{1'}} + m_{\ell_{1'}} \ell_{1'}^2 + m_{\ell_{2'}} a_{1'}^2 \left(1 + \frac{\ell_{2'} \bar{\ell}_{3'}}{a_{1'} a_{1''}} \right) + I_{\ell_{3'}} \\
b_{a22} &= I_{\ell_{1'}} + m_{\ell_{1'}} \ell_{1'}^2 + I_{\ell_{2'}} + m_{\ell_{2'}} \ell_{2'}^2 \left(1 + \frac{a_{1'} a_{1''}}{\ell_{2'} \bar{\ell}_{3'}} \right).
\end{aligned}$$

As a consequence, no contributions of Coriolis and centrifugal torques are obtained. Such a result could not be achieved with the previous two-link planar arm, no matter how the design parameters were chosen.

As for the gravitational terms, since $\boldsymbol{g}_0 = [0 \quad -g \quad 0]^T$, (7.39) with the above Jacobians gives

$$\begin{aligned} g_{1'} \;& (m_{\ell_{1'}}\ell_{1'} + m_{\ell_{2'}}a_{1'} + m_{\ell_{3'}}a_{1'})gc_{1'} + (m_{\ell_{2'}}\ell_{2'} + m_{\ell_{3'}}a_{2'})gc_{1'2'} \\ & +m_{\ell_{3'}}\ell_{3'}gc_{1'2'3} \\ g_{2'} \;& (m_{\ell_{2'}}\ell_{2'} + m_{\ell_{3'}}a_{2'})gc_{1'2'} + m_{\ell_{3'}}\ell_{3'}gc_{1'2'3} \\ g_{3'} \;& m_{\ell_{3'}}\ell_{3'}gc_{1'2'3} \end{aligned}$$

and

$$g_{1''} = m_{\ell_{1''}}\ell_{1''}gc_{1''}.$$

Composing the various contributions as done above yields

$$\boldsymbol{g}_a = \begin{bmatrix} (m_{\ell_{1'}}\ell_{1'} + m_{\ell_{2'}}a_{1'} - m_{\ell_{3'}}\bar{\ell}_{3'})gc_{1'} \\ (m_{\ell_{1''}}\ell_{1''} + m_{\ell_{2'}}\ell_{2'} + m_{\ell_{3'}}a_{1''})gc_{1''} \end{bmatrix}$$

which, together with the inertial torques, completes the derivation of the sought dynamic model.

A final comment is in order. In spite of its kinematic equivalence with the two-link planar arm, the dynamic model of the parallelogram is remarkably lighter. This property is quite advantageous for trajectory planning and control purposes. For this reason, apart from obvious considerations related to manipulation of heavy payloads, the adoption of closed kinematic chains in the design of industrial robots has received a great deal of attention.

7.4 Dynamic Parameter Identification

The use of the dynamic model for solving simulation and control problems demands the knowledge of the values of dynamic parameters of the manipulator model.

Computing such parameters from the design data of the mechanical structure is not simple. CAD modelling techniques can be adopted which allow the computation of the values of the inertial parameters of the various components (links, actuators and transmissions) on the basis of their geometry and type of materials employed. Nevertheless, the estimates obtained by such techniques are inaccurate because of the simplification typically introduced by geometric modelling; moreover, complex dynamic effects, such as joint friction, cannot be taken into account.

A heuristic approach could be to dismantle the various components of the manipulator and perform a series of measurements to evaluate the inertial parameters. Such technique is not easy to implement and may be troublesome to measure the relevant quantities.

In order to find accurate estimates of dynamic parameters, it is worth resorting to *identification* techniques which conveniently exploit the *property of linearity* (7.81) of the manipulator model with respect to a suitable set of

dynamic parameters. Such techniques allow the computation of the parameter vector $\boldsymbol{\pi}$ from the measurements of joint torques $\boldsymbol{\tau}$ and of relevant quantities for the evaluation of the matrix $\boldsymbol{Y}$, when suitable motion trajectories are imposed to the manipulator.

On the assumption that the kinematic parameters in the matrix $\boldsymbol{Y}$ are known with good accuracy, e.g., as a result of a kinematic calibration, measurements of joint positions $\boldsymbol{q}$, velocities $\dot{\boldsymbol{q}}$ and accelerations $\ddot{\boldsymbol{q}}$ are required. Joint positions and velocities can be actually measured while numerical reconstruction of accelerations is needed; this can be performed on the basis of the position and velocity values recorded during the execution of the trajectories. The reconstructing filter does not work in real time and thus it can also be anti-causal, allowing an accurate reconstruction of the accelerations.

As regards joint torques, in the unusual case of torque sensors at the joint, these can be measured directly. Otherwise, they can be evaluated from either wrist force measurements or current measurements in the case of electric actuators.

If measurements of joint torques, positions, velocities and accelerations have been obtained at given time instants $t_1, \ldots, t_N$ along a given trajectory, one may write

$$\bar{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\tau}(t_1) \\ \vdots \\ \boldsymbol{\tau}(t_N) \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}(t_1) \\ \vdots \\ \boldsymbol{Y}(t_N) \end{bmatrix} \boldsymbol{\pi} = \bar{\boldsymbol{Y}}\boldsymbol{\pi}. \tag{7.85}$$

The number of time instants sets the number of measurements to perform and should be large enough (typically $Nn \gg p$) so as to avoid ill-conditioning of matrix $\bar{\boldsymbol{Y}}$. Solving (7.85) by a least-squares technique leads to the solution in the form

$$\boldsymbol{\pi} = (\bar{\boldsymbol{Y}}^T\bar{\boldsymbol{Y}})^{-1}\bar{\boldsymbol{Y}}^T\bar{\boldsymbol{\tau}} \tag{7.86}$$

where $(\bar{\boldsymbol{Y}}^T\bar{\boldsymbol{Y}})^{-1}\bar{\boldsymbol{Y}}^T$ is the *left pseudo-inverse* matrix of $\bar{\boldsymbol{Y}}$.

It should be noticed that, in view of the block triangular structure of matrix $\boldsymbol{Y}$ in (7.80), computation of parameter estimates could be simplified by resorting to a sequential procedure. Take the equation $\tau_n = \boldsymbol{y}_{nn}^T\boldsymbol{\pi}_n$ and solve it for $\boldsymbol{\pi}_n$ by specifying τ_n and $\boldsymbol{y}_{nn}^T$ for a given trajectory on Joint n. By iterating the procedure, the manipulator parameters can be identified on the basis of measurements performed joint by joint from the outer link to the base. Such procedure, however, may have the inconvenience to accumulate any error due to ill-conditioning of the matrices involved step by step. It may then be worth operating with a global procedure by imposing motions on all manipulator joints at the same time.

Regarding the rank of matrix $\bar{\boldsymbol{Y}}$, it is possible to identify only the dynamic parameters of the manipulator that contribute to the dynamic model. Example 7.2 has indeed shown that for the two-link planar arm considered, only 8 out of the 22 possible dynamic parameters appear in the dynamic model. Hence, there exist some dynamic parameters which, in view of the disposition

of manipulator links and joints, are *non-identifiable*, since for any trajectory assigned to the structure they do not contribute to the equations of motion. A direct consequence is that the columns of the matrix $\boldsymbol{Y}$ in (7.80) corresponding to such parameters are null and thus they have to be removed from the matrix itself; e.g., the resulting (2×8) matrix in (7.84).

Another issue to consider about determination of the effective number of parameters that can be identified by (7.86) is that some parameters can be *identified in linear combinations* whenever they do not appear isolated in the equations. In such a case, it is necessary, for each linear combination, to remove as many columns of the matrix $\boldsymbol{Y}$ as the number of parameters in the linear combination minus one.

For the determination of the minimum number of identifiable parameters that allow direct application of the least-squares technique based on (7.86), it is possible to inspect directly the equations of the dynamic model, as long as the manipulator has few joints. Otherwise, numerical techniques based on singular value decomposition of matrix $\bar{\boldsymbol{Y}}$ have to be used. If the matrix $\bar{\boldsymbol{Y}}$ resulting from a series of measurements is not full-rank, one has to resort to a *damped least-squares inverse* of $\bar{\boldsymbol{Y}}$ where solution accuracy depends on the weight of the damping factor.

In the above discussion, the type of trajectory imposed to the manipulator joints has not been explicitly addressed. It can be generally ascertained that the choice should be oriented in favor of polynomial type trajectories which are sufficiently *rich* to allow an accurate evaluation of the identifiable parameters. This corresponds to achieving a low condition number of the matrix $\bar{\boldsymbol{Y}}^T\bar{\boldsymbol{Y}}$ along the trajectory. On the other hand, such trajectories should not excite any unmodelled dynamic effects such as joint elasticity or link flexibility that would naturally lead to unreliable estimates of the dynamic parameters to identify.

Finally, it is worth observing that the technique presented above can also be extended to the identification of the dynamic parameters of an unknown payload at the manipulator's end-effector. In such a case, the payload can be regarded as a structural modification of the last link and one may proceed to identify the dynamic parameters of the modified link. To this end, if a force sensor is available at the manipulator's wrist, it is possible to characterize directly the dynamic parameters of the payload starting from force sensor measurements.

7.5 Newton–Euler Formulation

In the Lagrange formulation, the manipulator dynamic model is derived starting from the total Lagrangian of the system. On the other hand, the *Newton–Euler* formulation is based on a balance of all the forces acting on the generic link of the manipulator. This leads to a set of equations whose structure allows a recursive type of solution; a forward recursion is performed for propagating

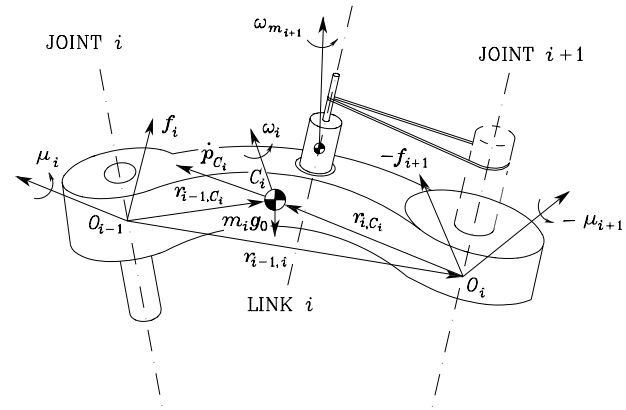

Fig. 7.13. Characterization of Link i for Newton–Euler formulation

link velocities and accelerations, followed by a backward recursion for propagating forces.

Consider the generic *augmented Link* i (Link i plus motor of Joint $i+1$) of the manipulator kinematic chain (Fig. 7.13). According to what was presented in Sect. 7.2.2, one can refer to the centre of mass C_i of the augmented link to characterize the following parameters:

- m_i mass of augmented link,
- $\bar{\boldsymbol{I}}_i$ inertia tensor of augmented link,
- I_{m_i} moment of inertia of rotor,
- $\boldsymbol{r}_{i-1,C_i}$ vector from origin of Frame $(i-1)$ to centre of mass C_i,
- $\boldsymbol{r}_{i,C_i}$ vector from origin of Frame i to centre of mass C_i,
- $\boldsymbol{r}_{i-1,i}$ vector from origin of Frame $(i-1)$ to origin of Frame i.

The velocities and accelerations to be considered are:

- $\dot{\boldsymbol{p}}_{C_i}$ linear velocity of centre of mass C_i,
- $\dot{\boldsymbol{p}}_i$ linear velocity of origin of Frame i,
- $\boldsymbol{\omega}_i$ angular velocity of link,
- $\boldsymbol{\omega}_{m_i}$ angular velocity of rotor,
- $\ddot{\boldsymbol{p}}_{C_i}$ linear acceleration of centre of mass C_i,
- $\ddot{\boldsymbol{p}}_i$ linear acceleration of origin of Frame i,
- $\dot{\boldsymbol{\omega}}_i$ angular acceleration of link,
- $\dot{\boldsymbol{\omega}}_{m_i}$ angular acceleration of rotor,
- $\boldsymbol{g}_0$ gravity acceleration.

The forces and moments to be considered are:

- $\boldsymbol{f}_i$ force exerted by Link $i-1$ on Link i,
- $-\boldsymbol{f}_{i+1}$ force exerted by Link $i+1$ on Link i,

- $\boldsymbol{\mu}_i$ moment exerted by Link $i-1$ on Link i with respect to origin of Frame $i-1$,
- $-\boldsymbol{\mu}_{i+1}$ moment exerted by Link $i+1$ on Link i with respect to origin of Frame i.

Initially, all the vectors and matrices are assumed to be expressed with reference to the *base frame*.

As already anticipated, the Newton–Euler formulation describes the motion of the link in terms of a balance of forces and moments acting on it.

The *Newton* equation for the *translational* motion of the centre of mass can be written as

$$\boldsymbol{f}_i - \boldsymbol{f}_{i+1} + m_i \boldsymbol{g}_0 = m_i \ddot{\boldsymbol{p}}_{C_i}. \tag{7.87}$$

The *Euler* equation for the *rotational* motion of the link (referring moments to the centre of mass) can be written as

$$\boldsymbol{\mu}_i + \boldsymbol{f}_i \times \boldsymbol{r}_{i-1,C_i} - \boldsymbol{\mu}_{i+1} - \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,C_i} = \frac{d}{dt}(\bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}), \tag{7.88}$$

where (7.67) has been used for the angular momentum of the rotor. Notice that the gravitational force $m_i \boldsymbol{g}_0$ does not generate any moment, since it is concentrated at the centre of mass.

As pointed out in the above Lagrange formulation, it is convenient to express the inertia tensor in the current frame (constant tensor). Hence, according to (7.12), one has $\bar{\boldsymbol{I}}_i = \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T$, where $\boldsymbol{R}_i$ is the rotation matrix from Frame i to the base frame. Substituting this relation in the first term on the right-hand side of (7.88) yields

$$\begin{aligned} \frac{d}{dt}(\bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i) &= \dot{\boldsymbol{R}}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \dot{\boldsymbol{R}}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T \dot{\boldsymbol{\omega}}_i \\ &= \boldsymbol{S}(\boldsymbol{\omega}_i) \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T \boldsymbol{S}^T(\boldsymbol{\omega}_i) \boldsymbol{\omega}_i + \boldsymbol{R}_i \bar{\boldsymbol{I}}_i^i \boldsymbol{R}_i^T \dot{\boldsymbol{\omega}}_i \\ &= \bar{\boldsymbol{I}}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i) \end{aligned} \tag{7.89}$$

where the second term represents the *gyroscopic* torque induced by the dependence of $\bar{\boldsymbol{I}}_i$ on link orientation.[7] Moreover, by observing that the unit vector $\boldsymbol{z}_{m_{i+1}}$ rotates accordingly to Link i, the derivative needed in the second term on the right-hand side of (7.88) is

$$\frac{d}{dt}(\dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}}) = \ddot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}} + \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{\omega}_i \times \boldsymbol{z}_{m_{i+1}} \tag{7.90}$$

By substituting (7.89), (7.90) in (7.88), the resulting Euler equation is

$$\begin{aligned} \boldsymbol{\mu}_i + \boldsymbol{f}_i \times \boldsymbol{r}_{i-1,C_i} \; -\boldsymbol{\mu}_{i+1} - \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,C_i} &= \bar{\boldsymbol{I}}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i) \\ + k_{r,i+1} \ddot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}} &+ k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{\omega}_i \times \boldsymbol{z}_{m_{i+1}}. \end{aligned} \tag{7.91}$$

[7] In deriving (7.89), the operator $\boldsymbol{S}$ has been introduced to compute the derivative of $\boldsymbol{R}_i$, as in (3.8); also, the property $\boldsymbol{S}^T(\boldsymbol{\omega}_i)\boldsymbol{\omega}_i = \boldsymbol{0}$ has been utilized.

The generalized force at Joint i can be computed by projecting the force $\boldsymbol{f}_i$ for a prismatic joint, or the moment $\boldsymbol{\mu}_i$ for a revolute joint, along the joint axis. In addition, there is the contribution of the rotor inertia torque $k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}_{m_i}^T \boldsymbol{z}_{m_i}$. Hence, the generalized force at Joint i is expressed by

$$\tau_i = \begin{cases} \boldsymbol{f}_i^T \boldsymbol{z}_{i-1} + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}_{m_i}^T \boldsymbol{z}_{m_i} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{\mu}_i^T \boldsymbol{z}_{i-1} + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}_{m_i}^T \boldsymbol{z}_{m_i} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{7.92}$$

7.5.1 Link Accelerations

The Newton–Euler equations in (7.87), (7.91) and the equation in (7.92) require the computation of linear and angular acceleration of Link i and Rotor i. This computation can be carried out on the basis of the relations expressing the linear and angular velocities previously derived. The equations in (3.21), (3.22), (3.25), (3.26) can be briefly rewritten as

$$\boldsymbol{\omega}_i = \begin{cases} \boldsymbol{\omega}_{i-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{\omega}_{i-1} + \dot{\vartheta}_i \boldsymbol{z}_{i-1} & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.93}$$

and

$$\dot{\boldsymbol{p}}_i = \begin{cases} \dot{\boldsymbol{p}}_{i-1} + \dot{d}_i \boldsymbol{z}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i} & \text{for a } \textit{prismatic} \text{ joint} \\ \dot{\boldsymbol{p}}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i} & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{7.94}$$

As for the angular acceleration of the link, it can be seen that, for a prismatic joint, differentiating (3.21) with respect to time gives

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1}, \tag{7.95}$$

whereas, for a revolute joint, differentiating (3.25) with respect to time gives

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + \ddot{\vartheta}_i \boldsymbol{z}_{i-1} + \dot{\vartheta}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{i-1}. \tag{7.96}$$

As for the linear acceleration of the link, for a prismatic joint, differentiating (3.22) with respect to time gives

$$\begin{aligned} \ddot{\boldsymbol{p}}_i = \ddot{\boldsymbol{p}}_{i-1} + \ddot{d}_i \boldsymbol{z}_{i-1} + \dot{d}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} \\ + \boldsymbol{\omega}_i \times \dot{d}_i \boldsymbol{z}_{i-1} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_{i-1} \times \boldsymbol{r}_{i-1,i}) \end{aligned} \tag{7.97}$$

where the relation $\dot{\boldsymbol{r}}_{i-1,i} = \dot{d}_i \boldsymbol{z}_{i-1} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{r}_{i-1,i}$ has been used. Hence, in view of (3.21), the equation in (7.97) can be rewritten as

$$\ddot{\boldsymbol{p}}_i = \ddot{\boldsymbol{p}}_{i-1} + \ddot{d}_i \boldsymbol{z}_{i-1} + 2\dot{d}_i \boldsymbol{\omega}_i \times \boldsymbol{z}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}). \tag{7.98}$$

Also, for a revolute joint, differentiating (3.26) with respect to time gives

$$\ddot{\boldsymbol{p}}_i = \ddot{\boldsymbol{p}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}). \tag{7.99}$$

In summary, the equations in (7.95), (7.96), (7.98), (7.99) can be compactly rewritten as

$$\dot{\boldsymbol{\omega}}_i = \begin{cases} \dot{\boldsymbol{\omega}}_{i-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \dot{\boldsymbol{\omega}}_{i-1} + \ddot{\vartheta}_i \boldsymbol{z}_{i-1} + \dot{\vartheta}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{i-1} & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.100}$$

and

$$\ddot{\boldsymbol{p}}_i = \begin{cases} \ddot{\boldsymbol{p}}_{i-1} + \ddot{d}_i \boldsymbol{z}_{i-1} + 2\dot{d}_i \boldsymbol{\omega}_i \times \boldsymbol{z}_{i-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \quad + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}) & \\ \ddot{\boldsymbol{p}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} & \text{for a } \textit{revolute} \text{ joint.} \\ \quad + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}) & \end{cases} \tag{7.101}$$

The acceleration of the centre of mass of Link i required by the Newton equation in (7.87) can be derived from (3.15), since $\dot{\boldsymbol{r}}^i_{i,C_i} = \boldsymbol{0}$; by differentiating (3.15) with respect to time, the acceleration of the centre of mass C_i can be expressed as a function of the velocity and acceleration of the origin of Frame i, i.e.,

$$\ddot{\boldsymbol{p}}_{C_i} = \ddot{\boldsymbol{p}}_i + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i,C_i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i,C_i}). \tag{7.102}$$

Finally, the angular acceleration of the rotor can be obtained by time differentiation of (7.23), i.e.,

$$\dot{\boldsymbol{\omega}}_{m_i} = \dot{\boldsymbol{\omega}}_{i-1} + k_{ri} \ddot{q}_i \boldsymbol{z}_{m_i} + k_{ri} \dot{q}_i \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{m_i}. \tag{7.103}$$

7.5.2 Recursive Algorithm

It is worth remarking that the resulting Newton–Euler equations of motion are *not* in *closed form*, since the motion of a single link is coupled to the motion of the other links through the kinematic relationship for velocities and accelerations.

Once the joint positions, velocities and accelerations are known, one can compute the link velocities and accelerations, and the Newton–Euler equations can be utilized to find the forces and moments acting on each link in a recursive fashion, starting from the force and moment applied to the end-effector. On the other hand, also link and rotor velocities and accelerations can be computed recursively starting from the velocity and acceleration of the base link. In summary, a computationally *recursive algorithm* can be constructed that features a *forward recursion* relative to the propagation of *velocities and accelerations* and a *backward recursion* for the propagation of *forces and moments* along the structure.

For the forward recursion, once $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$, and the velocity and acceleration of the base link $\boldsymbol{\omega}_0$, $\ddot{\boldsymbol{p}}_0 - \boldsymbol{g}_0$, $\dot{\boldsymbol{\omega}}_0$ are specified, $\boldsymbol{\omega}_i$, $\dot{\boldsymbol{\omega}}_i$, $\ddot{\boldsymbol{p}}_i$, $\ddot{\boldsymbol{p}}_{C_i}$, $\dot{\boldsymbol{\omega}}_{m_i}$ can be computed using (7.93), (7.100), (7.101), (7.102), (7.103), respectively. Notice that the linear acceleration has been taken as $\ddot{\boldsymbol{p}}_0 - \boldsymbol{g}_0$ so as to incorporate the term $-\boldsymbol{g}_0$ in the computation of the acceleration of the centre of mass $\ddot{\boldsymbol{p}}_{C_i}$ via (7.101), (7.102).

Having computed the velocities and accelerations with the forward recursion from the base link to the end-effector, a backward recursion can be carried out for the forces. In detail, once $\boldsymbol{h}_e = [\, \boldsymbol{f}^T_{n+1} \quad \boldsymbol{\mu}^T_{n+1} \,]^T$ is given (eventually $\boldsymbol{h}_e = \boldsymbol{0}$), the Newton equation in (7.87) to be used for the recursion can be rewritten as

$$\boldsymbol{f}_i = \boldsymbol{f}_{i+1} + m_i \ddot{\boldsymbol{p}}_{C_i} \tag{7.104}$$

since the contribution of gravity acceleration has already been included in $\ddot{\boldsymbol{p}}_{C_i}$. Further, the Euler equation gives

$$\begin{aligned} \boldsymbol{\mu}_i = {} & -\boldsymbol{f}_i \times (\boldsymbol{r}_{i-1,i} + \boldsymbol{r}_{i,C_i}) + \boldsymbol{\mu}_{i+1} + \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,C_i} + \bar{\boldsymbol{I}}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\bar{\boldsymbol{I}}_i \boldsymbol{\omega}_i) \\ & + k_{r,i+1} \ddot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}_{m_{i+1}} + k_{r,i+1} \dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{\omega}_i \times \boldsymbol{z}_{m_{i+1}} \end{aligned} \tag{7.105}$$

which derives from (7.91), where $\boldsymbol{r}_{i-1,C_i}$ has been expressed as the sum of the two vectors appearing already in the forward recursion. Finally, the generalized forces resulting at the joints can be computed from (7.92) as

$$\tau_i = \begin{cases} \boldsymbol{f}^T_i \boldsymbol{z}_{i-1} + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}^T_{m_i} \boldsymbol{z}_{m_i} & \\ \quad + F_{vi} \dot{d}_i + F_{si} \operatorname{sgn}(\dot{d}_i) & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{\mu}^T_i \boldsymbol{z}_{i-1} + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}^T_{m_i} \boldsymbol{z}_{m_i} & \\ \quad + F_{vi} \dot{\vartheta}_i + F_{si} \operatorname{sgn}(\dot{\vartheta}_i) & \text{for a } \textit{revolute} \text{ joint,} \end{cases} \tag{7.106}$$

where joint viscous and Coulomb friction torques have been included.

In the above derivation, it has been assumed that all vectors were referred to the base frame. To simplify greatly computation, however, the recursion is computationally more efficient if all vectors are referred to the current frame on Link i. This implies that all vectors that need to be transformed from Frame $i+1$ into Frame i have to be multiplied by the rotation matrix $\boldsymbol{R}^i_{i+1}$, whereas all vectors that need to be transformed from Frame $i-1$ into Frame i have to be multiplied by the rotation matrix $\boldsymbol{R}^{i-1T}_i$. Therefore, the equations in (7.93), (7.100), (7.101), (7.102), (7.103), (7.104), (7.105), (7.106) can be rewritten as:

$$\boldsymbol{\omega}^i_i = \begin{cases} \boldsymbol{R}^{i-1T}_i \boldsymbol{\omega}^{i-1}_{i-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{R}^{i-1T}_i (\boldsymbol{\omega}^{i-1}_{i-1} + \dot{\vartheta}_i \boldsymbol{z}_0) & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.107}$$

$$\dot{\boldsymbol{\omega}}^i_i = \begin{cases} \boldsymbol{R}^{i-1T}_i \dot{\boldsymbol{\omega}}^{i-1}_{i-1} & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{R}^{i-1T}_i (\dot{\boldsymbol{\omega}}^{i-1}_{i-1} + \ddot{\vartheta}_i \boldsymbol{z}_0 + \dot{\vartheta}_i \boldsymbol{\omega}^{i-1}_{i-1} \times \boldsymbol{z}_0) & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.108}$$

$$\ddot{\boldsymbol{p}}^i_i = \begin{cases} \boldsymbol{R}^{i-1T}_i (\ddot{\boldsymbol{p}}^{i-1}_{i-1} + \ddot{d}_i \boldsymbol{z}_0) + 2\dot{d}_i \boldsymbol{\omega}^i_i \times \boldsymbol{R}^{i-1T}_i \boldsymbol{z}_0 & \\ \quad + \dot{\boldsymbol{\omega}}^i_i \times \boldsymbol{r}^i_{i-1,i} + \boldsymbol{\omega}^i_i \times (\boldsymbol{\omega}^i_i \times \boldsymbol{r}^i_{i-1,i}) & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{R}^{i-1T}_i \ddot{\boldsymbol{p}}^{i-1}_{i-1} + \dot{\boldsymbol{\omega}}^i_i \times \boldsymbol{r}^i_{i-1,i} & \\ \quad + \boldsymbol{\omega}^i_i \times (\boldsymbol{\omega}^i_i \times \boldsymbol{r}^i_{i-1,i}) & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{7.109}$$

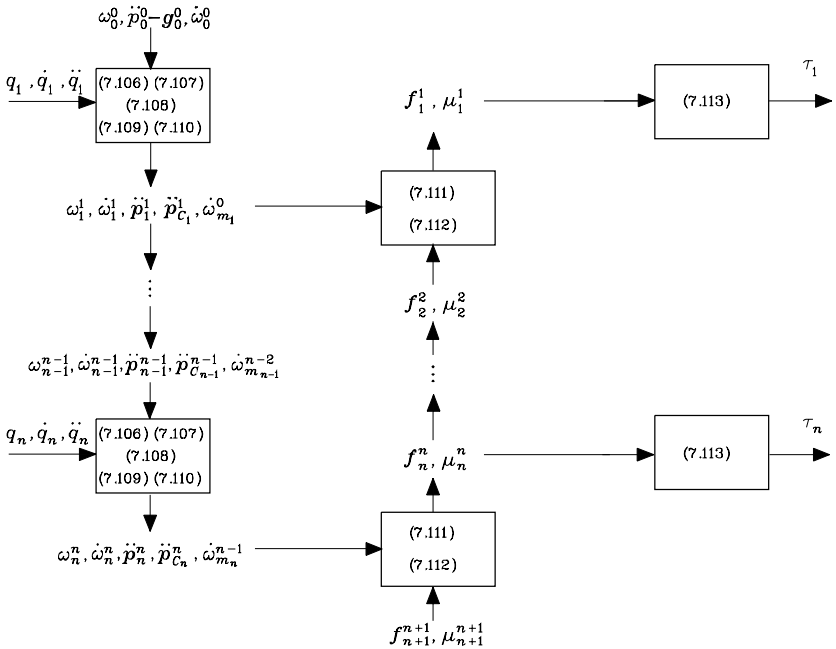

Fig. 7.14. Computational structure of the Newton–Euler recursive algorithm

$$\ddot{\boldsymbol{p}}^i_{C_i} = \ddot{\boldsymbol{p}}^i_i + \dot{\boldsymbol{\omega}}^i_i \times \boldsymbol{r}^i_{i,C_i} + \boldsymbol{\omega}^i_i \times (\boldsymbol{\omega}^i_i \times \boldsymbol{r}^i_{i,C_i}) \tag{7.110}$$

$$\dot{\boldsymbol{\omega}}^{i-1}_{m_i} = \dot{\boldsymbol{\omega}}^{i-1}_{i-1} + k_{ri}\ddot{q}_i \boldsymbol{z}^{i-1}_{m_i} + k_{ri}\dot{q}_i \boldsymbol{\omega}^{i-1}_{i-1} \times \boldsymbol{z}^{i-1}_{m_i} \tag{7.111}$$

$$\boldsymbol{f}^i_i = \boldsymbol{R}^i_{i+1}\boldsymbol{f}^{i+1}_{i+1} + m_i \ddot{\boldsymbol{p}}^i_{C_i} \tag{7.112}$$

$$\begin{aligned}\boldsymbol{\mu}^i_i = {} & -\boldsymbol{f}^i_i \times (\boldsymbol{r}^i_{i-1,i} + \boldsymbol{r}^i_{i,C_i}) + \boldsymbol{R}^i_{i+1}\boldsymbol{\mu}^{i+1}_{i+1} + \boldsymbol{R}^i_{i+1}\boldsymbol{f}^{i+1}_{i+1} \times \boldsymbol{r}^i_{i,C_i} \\ & + \bar{\boldsymbol{I}}^i_i \dot{\boldsymbol{\omega}}^i_i + \boldsymbol{\omega}^i_i \times (\bar{\boldsymbol{I}}^i_i \boldsymbol{\omega}^i_i) \\ & + \boldsymbol{\omega}^i_i \times (\bar{\boldsymbol{I}}^i_i \boldsymbol{\omega}^i_i) + k_{r,i+1}\ddot{q}_{i+1} I_{m_{i+1}} \boldsymbol{z}^i_{m_{i+1}} + k_{r,i+1}\dot{q}_{i+1} I_{m_{i+1}} \boldsymbol{\omega}^i_i \times \boldsymbol{z}^i_{m_{i+1}}\end{aligned} \tag{7.113}$$

$$\tau_i = \begin{cases} \boldsymbol{f}^{iT}_i \boldsymbol{R}^{i-1T}_i \boldsymbol{z}_0 + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}^{i-1T}_{m_i} \boldsymbol{z}^{i-1}_{m_i} \\ \quad + F_{vi}\dot{d}_i + F_{si}\,\mathrm{sgn}\,(\dot{d}_i) & \text{for a } \textit{prismatic} \text{ joint} \\ \boldsymbol{\mu}^{iT}_i \boldsymbol{R}^{i-1T}_i \boldsymbol{z}_0 + k_{ri} I_{m_i} \dot{\boldsymbol{\omega}}^{i-1T}_{m_i} \boldsymbol{z}^{i-1}_{m_i} \\ \quad + F_{vi}\dot{\vartheta}_i + F_{si}\,\mathrm{sgn}\,(\dot{\vartheta}_i) & \text{for a } \textit{revolute} \text{ joint.} \end{cases} \tag{7.114}$$

The above equations have the advantage that the quantities $\bar{\boldsymbol{I}}^i_i$, $\boldsymbol{r}^i_{i,C_i}$, $\boldsymbol{z}^{i-1}_{m_i}$ are *constant*; further, it is $\boldsymbol{z}_0 = [\,0 \quad 0 \quad 1\,]^T$.

To summarize, for given joint positions, velocities and accelerations, the recursive algorithm is carried out in the following two phases:

- With known initial conditions $\boldsymbol{\omega}^0_0$, $\ddot{\boldsymbol{p}}^0_0 - \boldsymbol{g}^0_0$, and $\dot{\boldsymbol{\omega}}^0_0$, use (7.107), (7.108), (7.109), (7.110), (7.111), for $i = 1, \ldots, n$, to compute $\boldsymbol{\omega}^i_i$, $\dot{\boldsymbol{\omega}}^i_i$, $\ddot{\boldsymbol{p}}^i_i$, $\ddot{\boldsymbol{p}}^i_{C_i}$, $\dot{\boldsymbol{\omega}}^{i-1}_{m_i}$.
- With known terminal conditions $\boldsymbol{f}^{n+1}_{n+1}$ and $\boldsymbol{\mu}^{n+1}_{n+1}$, use (7.112), (7.113), for $i = n, \ldots, 1$, to compute $\boldsymbol{f}^i_i$, $\boldsymbol{\mu}^i_i$, and then (7.114) to compute τ_i.

The computational structure of the algorithm is schematically illustrated in Fig. 7.14.

7.5.3 Example

In the following, an example to illustrate the single steps of the Newton–Euler algorithm is developed. Consider the two-link planar arm whose dynamic model has already been derived in Example 7.2.

Start by imposing the initial conditions for the velocities and accelerations:

$$\ddot{\boldsymbol{p}}^0_0 - \boldsymbol{g}^0_0 = [\,0 \quad g \quad 0\,]^T \qquad \boldsymbol{\omega}^0_0 = \dot{\boldsymbol{\omega}}^0_0 = \boldsymbol{0},$$

and the terminal conditions for the forces:

$$\boldsymbol{f}^3_3 = \boldsymbol{0} \qquad \boldsymbol{\mu}^3_3 = \boldsymbol{0}.$$

All quantities are referred to the current link frame. As a consequence, the following constant vectors are obtained:

$$\boldsymbol{r}^1_{1,C_1} = \begin{bmatrix} \ell_{C_1} \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}^1_{0,1} = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}^2_{2,C_2} = \begin{bmatrix} \ell_{C_2} \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{r}^2_{1,2} = \begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix}$$

where ℓ_{C_1} and ℓ_{C_2} are both negative quantities. The rotation matrices needed for vector transformation from one frame to another are

$$\boldsymbol{R}^{i-1}_i = \begin{bmatrix} c_i & -s_i & 0 \\ s_i & c_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad i = 1, 2 \qquad \boldsymbol{R}^2_3 = \boldsymbol{I}.$$

Further, it is assumed that the axes of rotation of the two rotors coincide with the respective joint axes, i.e., $\boldsymbol{z}^{i-1}_{m_i} = \boldsymbol{z}_0 = [\,0 \quad 0 \quad 1\,]^T$ for $i = 1, 2$.

According to (7.107)–(7.114), the Newton–Euler algorithm requires the execution of the following steps:

- Forward recursion: Link 1

$$\boldsymbol{\omega}^1_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\vartheta}_1 \end{bmatrix}$$

$$\dot{\boldsymbol{\omega}}_1^1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\vartheta}_1 \end{bmatrix}$$

$$\ddot{\boldsymbol{p}}_1^1 = \begin{bmatrix} -a_1\dot{\vartheta}_1^2 + gs_1 \\ a_1\ddot{\vartheta}_1 + gc_1 \\ 0 \end{bmatrix}$$

$$\ddot{\boldsymbol{p}}_{C_1}^1 = \begin{bmatrix} -(\ell_{C_1} + a_1)\dot{\vartheta}_1^2 + gs_1 \\ (\ell_{C_1} + a_1)\ddot{\vartheta}_1 + gc_1 \\ 0 \end{bmatrix}$$

$$\dot{\boldsymbol{\omega}}_{m_1}^0 = \begin{bmatrix} 0 \\ 0 \\ k_{r1}\ddot{\vartheta}_1 \end{bmatrix}.$$

- Forward recursion: Link 2

$$\boldsymbol{\omega}_2^2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\vartheta}_1 + \dot{\vartheta}_2 \end{bmatrix}$$

$$\dot{\boldsymbol{\omega}}_2^2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\vartheta}_1 + \ddot{\vartheta}_2 \end{bmatrix}$$

$$\ddot{\boldsymbol{p}}_2^2 = \begin{bmatrix} a_1 s_2\ddot{\vartheta}_1 - a_1 c_2\dot{\vartheta}_1^2 - a_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 + gs_{12} \\ a_1 c_2\ddot{\vartheta}_1 + a_2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + a_1 s_2\dot{\vartheta}_1^2 + gc_{12} \\ 0 \end{bmatrix}$$

$$\ddot{\boldsymbol{p}}_{C_2}^2 = \begin{bmatrix} a_1 s_2\ddot{\vartheta}_1 - a_1 c_2\dot{\vartheta}_1^2 - (\ell_{C_2} + a_2)(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 + gs_{12} \\ a_1 c_2\ddot{\vartheta}_1 + (\ell_{C_2} + a_2)(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + a_1 s_2\dot{\vartheta}_1^2 + gc_{12} \\ 0 \end{bmatrix}$$

$$\dot{\boldsymbol{\omega}}_{m_2}^1 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\vartheta}_1 + k_{r2}\ddot{\vartheta}_2 \end{bmatrix}.$$

- Backward recursion: Link 2

$$\boldsymbol{f}_2^2 = \begin{bmatrix} m_2\big(a_1 s_2\ddot{\vartheta}_1 - a_1 c_2\dot{\vartheta}_1^2 - (\ell_{C_2} + a_2)(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 + gs_{12}\big) \\ m_2\big(a_1 c_2\ddot{\vartheta}_1 + (\ell_{C_2} + a_2)(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + a_1 s_2\dot{\vartheta}_1^2 + gc_{12}\big) \\ 0 \end{bmatrix}$$

$$\boldsymbol{\mu}_2^2 = \begin{bmatrix} * \\ * \\ \bar{I}_{2zz}(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + m_2(\ell_{C_2} + a_2)^2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + m_2 a_1(\ell_{C_2} + a_2)c_2\ddot{\vartheta}_1 \\ \qquad + m_2 a_1(\ell_{C_2} + a_2)s_2\dot{\vartheta}_1^2 + m_2(\ell_{C_2} + a_2)gc_{12} \end{bmatrix}$$

$$\begin{aligned} \tau_2 = {} & \big(\bar{I}_{2zz} + m_2\left((\ell_{C_2} + a_2)^2 + a_1(\ell_{C_2} + a_2)c_2\right) + k_{r2}I_{m_2}\big)\ddot{\vartheta}_1 \\ & + \left(\bar{I}_{2zz} + m_2(\ell_{C_2} + a_2)^2 + k_{r2}^2 I_{m_2}\right)\ddot{\vartheta}_2 \\ & + m_2 a_1(\ell_{C_2} + a_2)s_2\dot{\vartheta}_1^2 + m_2(\ell_{C_2} + a_2)gc_{12}. \end{aligned}$$

- Backward recursion: Link 1

$$\boldsymbol{f}_1^1 = \begin{bmatrix} -m_2(\ell_{C_2} + a_2)s_2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) - m_1(\ell_{C_1} + a_1)\dot{\vartheta}_1^2 - m_2 a_1\dot{\vartheta}_1^2 \\ -m_2(\ell_{C_2} + a_2)c_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 + (m_1 + m_2)gs_1 \\ m_1(\ell_{C_1} + a_1)\ddot{\vartheta}_1 + m_2 a_1\ddot{\vartheta}_1 + m_2(\ell_{C_2} + a_2)c_2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) \\ -m_2(\ell_{C_2} + a_2)s_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 + (m_1 + m_2)gc_1 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\mu}_1^1 = \begin{bmatrix} * \\ * \\ \bar{I}_{1zz}\ddot{\vartheta}_1 + m_2 a_1^2\ddot{\vartheta}_1 + m_1(\ell_{C_1} + a_1)^2\ddot{\vartheta}_1 + m_2 a_1(\ell_{C_2} + a_2)c_2\ddot{\vartheta}_1 \\ \qquad + \bar{I}_{2zz}(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + m_2 a_1(\ell_{C_2} + a_2)c_2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) \\ \qquad + m_2(\ell_{C_2} + a_2)^2(\ddot{\vartheta}_1 + \ddot{\vartheta}_2) + k_{r2}I_{m_2}\ddot{\vartheta}_2 \\ \qquad + m_2 a_1(\ell_{C_2} + a_2)s_2\dot{\vartheta}_1^2 - m_2 a_1(\ell_{C_2} + a_2)s_2(\dot{\vartheta}_1 + \dot{\vartheta}_2)^2 \\ \qquad + m_1(\ell_{C_1} + a_1)gc_1 + m_2 a_1 gc_1 + m_2(\ell_{C_2} + a_2)gc_{12} \end{bmatrix}$$

$$\begin{aligned} \tau_1 = {} & \Big(\bar{I}_{1zz} + m_1(\ell_{C_1} + a_1)^2 + k_{r1}^2 I_{m_1} + \bar{I}_{2zz} \\ & \qquad + m_2\left(a_1^2 + (\ell_{C_2} + a_2)^2 + 2a_1(\ell_{C_2} + a_2)c_2\right)\Big)\ddot{\vartheta}_1 \\ & + \Big(\bar{I}_{2zz} + m_2\left((\ell_{C_2} + a_2)^2 + a_1(\ell_{C_2} + a_2)c_2\right) + k_{r2}I_{m_2}\Big)\ddot{\vartheta}_2 \\ & - 2m_2 a_1(\ell_{C_2} + a_2)s_2\dot{\vartheta}_1\dot{\vartheta}_2 - m_2 a_1(\ell_{C_2} + a_2)s_2\dot{\vartheta}_2^2 \\ & + \big(m_1(\ell_{C_1} + a_1) + m_2 a_1\big)gc_1 + m_2(\ell_{C_2} + a_2)gc_{12}. \end{aligned}$$

As for the moment components, those marked by the symbol '$*$' have not been computed, since they are not related to the joint torques τ_2 and τ_1.

Expressing the dynamic parameters in the above torques as a function of the link and rotor parameters as in (7.83) yields

$$\begin{aligned} m_1 &= m_{\ell_1} + m_{m_2} \\ m_1 \ell_{C_1} &= m_{\ell_1}(\ell_1 - a_1) \\ \bar{I}_{1zz} + m_1 \ell_{C_1}^2 = \widehat{I}_1 &= I_{\ell_1} + m_{\ell_1}(\ell_1 - a_1)^2 + I_{m_2} \\ m_2 &= m_{\ell_2} \\ m_2 \ell_{C_2} &= m_{\ell_2}(\ell_2 - a_2) \\ \bar{I}_{2zz} + m_2 \ell_{C_2}^2 = \widehat{I}_2 &= I_{\ell_2} + m_{\ell_2}(\ell_2 - a_2)^2. \end{aligned}$$

On the basis of these relations, it can be verified that the resulting dynamic model coincides with the model derived in (7.82) with Lagrange formulation.

7.6 Direct Dynamics and Inverse Dynamics

Both Lagrange formulation and Newton–Euler formulation allow the computation of the relationship between the joint torques — and, if present, the end-effector forces — and the motion of the structure. A comparison between the two approaches reveals what follows. The *Lagrange* formulation has the following advantages:

- It is *systematic* and of immediate comprehension.
- It provides the equations of motion in a compact *analytical form* containing the inertia matrix, the matrix in the centrifugal and Coriolis forces, and the vector of gravitational forces. Such a form is advantageous for *control design*.
- It is effective if it is wished to include more complex mechanical effects such as flexible link deformation.

The *Newton–Euler* formulation has the following fundamental advantage:

- It is an inherently *recursive* method that is computationally efficient.

In the study of dynamics, it is relevant to find a solution to two kinds of problems concerning computation of direct dynamics and inverse dynamics.

The *direct dynamics* problem consists of determining, for $t > t_0$, the joint accelerations $\ddot{\boldsymbol{q}}(t)$ (and thus $\dot{\boldsymbol{q}}(t)$, $\boldsymbol{q}(t)$) resulting from the given joint torques $\boldsymbol{\tau}(t)$ — and the possible end-effector forces $\boldsymbol{h}_e(t)$ — once the initial positions $\boldsymbol{q}(t_0)$ and velocities $\dot{\boldsymbol{q}}(t_0)$ are known (initial state of the system).

The *inverse dynamics* problem consists of determining the joint torques $\boldsymbol{\tau}(t)$ which are needed to generate the motion specified by the joint accelerations $\ddot{\boldsymbol{q}}(t)$, velocities $\dot{\boldsymbol{q}}(t)$, and positions $\boldsymbol{q}(t)$ — once the possible end-effector forces $\boldsymbol{h}_e(t)$ are known.

Solving the direct dynamics problem is useful for manipulator *simulation*. Direct dynamics allows the motion of the real physical system to be described in terms of the joint accelerations, when a set of assigned joint torques is applied to the manipulator; joint velocities and positions can be obtained by integrating the system of nonlinear differential equations.

Since the equations of motion obtained with Lagrange formulation give the analytical relationship between the joint torques (and the end-effector forces) and the joint positions, velocities and accelerations, these can be computed from (7.42) as

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{-1}(\boldsymbol{q})(\boldsymbol{\tau} - \boldsymbol{\tau}') \tag{7.115}$$

where

$$\boldsymbol{\tau}'(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}_v \dot{\boldsymbol{q}} + \boldsymbol{F}_s \,\mathbf{sgn}\,(\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}_e \tag{7.116}$$

denotes the torque contributions depending on joint positions and velocities. Therefore, for simulation of manipulator motion, once the state at the time instant t_k is known in terms of the position $\boldsymbol{q}(t_k)$ and velocity $\dot{\boldsymbol{q}}(t_k)$, the acceleration $\ddot{\boldsymbol{q}}(t_k)$ can be computed by (7.115). Then using a numerical integration method, e.g., Runge–Kutta, with integration step Δt, the velocity $\dot{\boldsymbol{q}}(t_{k+1})$ and position $\boldsymbol{q}(t_{k+1})$ at the instant $t_{k+1} = t_k + \Delta t$ can be computed.

If the equations of motion are obtained with Newton–Euler formulation, it is possible to compute direct dynamics by using a computationally more efficient method. In fact, for given $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, the torques $\boldsymbol{\tau}'(\boldsymbol{q}, \dot{\boldsymbol{q}})$ in (7.116) can be computed as the torques given by the algorithm of Fig. 7.14 with $\ddot{\boldsymbol{q}} = \boldsymbol{0}$. Further, column $\boldsymbol{b}_i$ of matrix $\boldsymbol{B}(\boldsymbol{q})$ can be computed as the torque vector given by the algorithm of Fig. 7.14 with $\boldsymbol{g}_0 = \boldsymbol{0}$, $\dot{\boldsymbol{q}} = \boldsymbol{0}$, $\ddot{q}_i = 1$ and $\ddot{q}_j = 0$ for $j \neq i$; iterating this procedure for $i = 1, \ldots, n$ leads to constructing the matrix $\boldsymbol{B}(\boldsymbol{q})$. Hence, from the current values of $\boldsymbol{B}(\boldsymbol{q})$ and $\boldsymbol{\tau}'(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and the given $\boldsymbol{\tau}$, the equations in (7.115) can be integrated as illustrated above.

Solving the inverse dynamics problem is useful for manipulator trajectory planning and control algorithm implementation. Once a joint trajectory is specified in terms of positions, velocities and accelerations (typically as a result of an inverse kinematics procedure), and if the end-effector forces are known, inverse dynamics allows computation of the torques to be applied to the joints to obtain the desired motion. This computation turns out to be useful both for verifying feasibility of the imposed trajectory and for compensating nonlinear terms in the dynamic model of a manipulator. To this end, Newton–Euler formulation provides a computationally efficient recursive method for on-line computation of inverse dynamics. Nevertheless, it can be shown that also Lagrange formulation is liable to a computationally efficient recursive implementation, though with a nonnegligible reformulation effort.

For an n-joint manipulator the *number of operations* required is:[8]

[8] See Sect. E.1 for the definition of computational complexity of an algorithm.

- $O(n^2)$ for computing *direct dynamics*,
- $O(n)$ for computing *inverse dynamics*.

7.7 Dynamic Scaling of Trajectories

The existence of *dynamic constraints* to be taken into account for trajectory generation has been mentioned in Sect. 4.1. In practice, with reference to the given trajectory time or path shape (segments with high curvature), the trajectories that can be obtained with any of the previously illustrated methods may impose too severe dynamic performance for the manipulator. A typical case is that when the required torques to generate the motion are larger than the maximum torques the actuators can supply. In this case, an infeasible trajectory has to be suitably time-scaled.

Suppose a trajectory has been generated for all the manipulator joints as $\boldsymbol{q}(t)$, for $t \in [0, t_f]$. Computing inverse dynamics allows the evaluation of the time history of the torques $\boldsymbol{\tau}(t)$ required for the execution of the given motion. By comparing the obtained torques with the *torque limits* available at the actuators, it is easy to check whether or not the trajectory is actually executable. The problem is then to seek an automatic trajectory *dynamic scaling* technique — avoiding inverse dynamics recomputation — so that the manipulator can execute the motion on the specified path with a proper timing law without exceeding the torque limits.

Consider the manipulator dynamic model as given in (7.42) with $\boldsymbol{F}_v = \boldsymbol{O}$, $\boldsymbol{F}_s = \boldsymbol{O}$ and $\boldsymbol{h}_e = \boldsymbol{0}$, for simplicity. The term $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ accounting for centrifugal and Coriolis forces has a quadratic dependence on joint velocities, and thus it can be formally rewritten as

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\Gamma}(\boldsymbol{q})[\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}], \tag{7.117}$$

where $[\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}]$ indicates the symbolic notation of the $(n(n+1)/2 \times 1)$ vector

$$[\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}] = [\, \dot{q}_1^2 \quad \dot{q}_1\dot{q}_2 \quad \dots \quad \dot{q}_{n-1}\dot{q}_n \quad \dot{q}_n^2 \,]^T;$$

$\boldsymbol{\Gamma}(\boldsymbol{q})$ is a proper $(n \times n(n+1)/2)$ matrix that satisfies (7.117). In view of such position, the manipulator dynamic model can be expressed as

$$\boldsymbol{B}(\boldsymbol{q}(t))\ddot{\boldsymbol{q}}(t) + \boldsymbol{\Gamma}(\boldsymbol{q}(t))[\dot{\boldsymbol{q}}(t)\dot{\boldsymbol{q}}(t)] + \boldsymbol{g}(\boldsymbol{q}(t)) = \boldsymbol{\tau}(t), \tag{7.118}$$

where the explicit dependence on time t has been shown.

Consider the new variable $\bar{\boldsymbol{q}}(r(t))$ satisfying the equation

$$\boldsymbol{q}(t) = \bar{\boldsymbol{q}}(r(t)), \tag{7.119}$$

where $r(t)$ is a strictly increasing scalar function of time with $r(0) = 0$ and $r(t_f) = \bar{t}_f$.

Differentiating (7.119) twice with respect to time provides the following relations:

$$\dot{\boldsymbol{q}} = \dot{r}\bar{\boldsymbol{q}}'(r) \tag{7.120}$$

$$\ddot{\boldsymbol{q}} = \dot{r}^2\bar{\boldsymbol{q}}''(r) + \ddot{r}\bar{\boldsymbol{q}}'(r) \tag{7.121}$$

where the prime denotes the derivative with respect to r. Substituting (7.120), (7.121) into (7.118) yields

$$\dot{r}^2\Big(\boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}''(r) + \boldsymbol{\Gamma}(\bar{\boldsymbol{q}}(r))[\bar{\boldsymbol{q}}'(r)\bar{\boldsymbol{q}}'(r)]\Big) + \ddot{r}\boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}'(r) + \boldsymbol{g}(\bar{\boldsymbol{q}}(r)) = \boldsymbol{\tau}. \tag{7.122}$$

In (7.118) it is possible to identify the term

$$\boldsymbol{\tau}_s(t) = \boldsymbol{B}(\boldsymbol{q}(t))\ddot{\boldsymbol{q}}(t) + \boldsymbol{\Gamma}(\boldsymbol{q}(t))[\dot{\boldsymbol{q}}(t)\dot{\boldsymbol{q}}(t)], \tag{7.123}$$

representing the torque contribution that depends on velocities and accelerations. Correspondingly, in (7.122) one can set

$$\boldsymbol{\tau}_s(t) = \dot{r}^2\Big(\boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}''(r) + \boldsymbol{\Gamma}(\bar{\boldsymbol{q}}(r))[\bar{\boldsymbol{q}}'(r)\bar{\boldsymbol{q}}'(r)]\Big) + \ddot{r}\boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}'(r). \tag{7.124}$$

By analogy with (7.123), it can be written

$$\bar{\boldsymbol{\tau}}_s(r) = \boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}''(r) + \boldsymbol{\Gamma}(\bar{\boldsymbol{q}}(r))[\bar{\boldsymbol{q}}'(r)\bar{\boldsymbol{q}}'(r)] \tag{7.125}$$

and then (7.124) becomes

$$\boldsymbol{\tau}_s(t) = \dot{r}^2\bar{\boldsymbol{\tau}}_s(r) + \ddot{r}\boldsymbol{B}(\bar{\boldsymbol{q}}(r))\bar{\boldsymbol{q}}'(r). \tag{7.126}$$

The expression in (7.126) gives the relationship between the torque contributions depending on velocities and accelerations required by the manipulator when this is subject to motions having the same path but different timing laws, obtained through a time scaling of joint variables as in (7.119).

Gravitational torques have not been considered, since they are a function of the joint positions only, and thus their contribution is not influenced by time scaling.

The simplest choice for the scaling function $r(t)$ is certainly the *linear* function

$$r(t) = ct$$

with c a positive constant. In this case, (7.126) becomes

$$\boldsymbol{\tau}_s(t) = c^2\bar{\boldsymbol{\tau}}_s(ct),$$

which reveals that a linear time scaling by c causes a scaling of the magnitude of the torques by the coefficient c^2. Let $c > 1$: (7.119) shows that the trajectory described by $\bar{\boldsymbol{q}}(r(t))$, assuming $r = ct$ as the independent variable, has a duration $\bar{t}_f > t_f$ to cover the entire path specified by $\boldsymbol{q}$. Correspondingly, the

torque contributions $\bar{\boldsymbol{\tau}}_s(ct)$ computed as in (7.125) are scaled by the factor c^2 with respect to the torque contributions $\boldsymbol{\tau}_s(t)$ required to execute the original trajectory $\boldsymbol{q}(t)$.

With the use of a recursive algorithm for inverse dynamics computation, it is possible to check whether the torques exceed the allowed limits during trajectory execution; obviously, limit violation should not be caused by the sole gravity torques. It is necessary to find the joint for which the torque has exceeded the limit more than the others, and to compute the torque contribution subject to scaling, which in turn determines the factor c^2. It is then possible to compute the time-scaled trajectory as a function of the new time variable $r = ct$ which no longer exceeds torque limits. It should be pointed out, however, that with this kind of linear scaling the entire trajectory may be penalized, even when a torque limit on a single joint is exceeded only for a short interval of time.

7.8 Operational Space Dynamic Model

As an alternative to the joint space dynamic model, the equations of motion of the system can be expressed directly in the operational space; to this end it is necessary to find a *dynamic model* which describes the relationship between the generalized forces acting on the manipulator and the number of minimal variables chosen to describe the end-effector position and orientation in the *operational space*.

Similar to kinematic description of a manipulator in the operational space, the presence of redundant DOFs and/or kinematic and representation singularities deserves careful attention in the derivation of an operational space dynamic model.

The determination of the dynamic model with Lagrange formulation using operational space variables allows a complete description of the system motion only in the case of a *nonredundant* manipulator, when the above variables constitute a set of *generalized coordinates* in terms of which the kinetic energy, the potential energy, and the nonconservative forces doing work on them can be expressed.

This way of proceeding does not provide a complete description of dynamics for a *redundant* manipulator; in this case, in fact, it is reasonable to expect the occurrence of *internal motions* of the structure caused by those joint generalized forces which do not affect the end-effector motion.

To develop an operational space model which can be adopted for both redundant and nonredundant manipulators, it is then convenient to start from the joint space model which is in all general. In fact, solving (7.42) for the joint accelerations, and neglecting the joint friction torques for simplicity, yields

$$\ddot{\boldsymbol{q}} = -\boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})(\boldsymbol{\gamma}_e - \boldsymbol{h}_e), \quad (7.127)$$

where the joint torques $\boldsymbol{\tau}$ have been expressed in terms of the equivalent end-effector forces $\boldsymbol{\gamma}$ according to (3.111). It is worth noting that $\boldsymbol{h}$ represents the contribution of the end-effector forces due to contact with the environment, whereas $\boldsymbol{\gamma}$ expresses the contribution of the end-effector forces due to joint actuation.

On the other hand, the second-order differential kinematics equation in (3.98) describes the relationship between joint space and operational space accelerations, i.e.,

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

The solution in (7.127) features the geometric Jacobian $\boldsymbol{J}$, whereas the analytical Jacobian $\boldsymbol{J}_A$ appears in (3.98). For notation uniformity, in view of (3.66), one can set

$$\boldsymbol{T}_A^T(\boldsymbol{x}_e)\boldsymbol{\gamma}_e = \boldsymbol{\gamma}_A \qquad \boldsymbol{T}_A^T(\boldsymbol{x}_e)\boldsymbol{h}_e = \boldsymbol{h}_A \quad (7.128)$$

where $\boldsymbol{T}_A$ is the transformation matrix between the two Jacobians. Substituting (7.127) into (3.98) and accounting for (7.128) gives

$$\ddot{\boldsymbol{x}}_e = -\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{g} + \dot{\boldsymbol{J}}_A\dot{\boldsymbol{q}} + \boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{J}_A^T(\boldsymbol{\gamma}_A - \boldsymbol{h}_A). \quad (7.129)$$

where the dependence on $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ has been omitted. With the positions

$$\boldsymbol{B}_A = (\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{J}_A^T)^{-1} \quad (7.130)$$

$$\boldsymbol{C}_A\dot{\boldsymbol{x}}_e = \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{B}_A\dot{\boldsymbol{J}}_A\dot{\boldsymbol{q}} \quad (7.131)$$

$$\boldsymbol{g}_A = \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{g}, \quad (7.132)$$

the expression in (7.129) can be rewritten as

$$\boldsymbol{B}_A(\boldsymbol{x}_e)\ddot{\boldsymbol{x}}_e + \boldsymbol{C}_A(\boldsymbol{x}_e,\dot{\boldsymbol{x}}_e)\dot{\boldsymbol{x}}_e + \boldsymbol{g}_A(\boldsymbol{x}_e) = \boldsymbol{\gamma}_A - \boldsymbol{h}_A, \quad (7.133)$$

which is formally analogous to the joint space dynamic model (7.42). Notice that the matrix $\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{J}_A^T$ is invertible if and only if $\boldsymbol{J}_A$ is full-rank, that is, in the absence of both kinematic and representation singularities.

For a nonredundant manipulator in a nonsingular configuration, the expressions in (7.130)–(7.132) become:

$$\boldsymbol{B}_A = \boldsymbol{J}_A^{-T}\boldsymbol{B}\boldsymbol{J}_A^{-1} \quad (7.134)$$

$$\boldsymbol{C}_A\dot{\boldsymbol{x}}_e = \boldsymbol{J}_A^{-T}\boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{B}_A\dot{\boldsymbol{J}}_A\dot{\boldsymbol{q}} \quad (7.135)$$

$$\boldsymbol{g}_A = \boldsymbol{J}_A^{-T}\boldsymbol{g}. \quad (7.136)$$

As anticipated above, the main feature of the obtained model is its formal validity also for a redundant manipulator, even though the variables $\boldsymbol{x}_e$ do not constitute a set of generalized coordinates for the system; in this case, the matrix $\boldsymbol{B}_A$ is representative of a *kinetic pseudo-energy*.

In the following, the utility of the operational space dynamic model in (7.133) for solving direct and inverse dynamics problems is investigated. The

following derivation is meaningful for redundant manipulators; for a nonredundant manipulator, in fact, using (7.133) does not pose specific problems as long as $\boldsymbol{J}_A$ is nonsingular ((7.134)–(7.136)).

With reference to operational space, the *direct dynamics* problem consists of determining the resulting end-effector accelerations $\ddot{\boldsymbol{x}}_e(t)$ (and thus $\dot{\boldsymbol{x}}_e(t)$, $\boldsymbol{x}_e(t)$) from the given joint torques $\boldsymbol{\tau}(t)$ and end-effector forces $\boldsymbol{h}_e(t)$. For a redundant manipulator, (7.133) cannot be directly used, since (3.111) has a solution in $\boldsymbol{\gamma}_e$ only if $\boldsymbol{\tau} \in \mathcal{R}(\boldsymbol{J}^T)$. It follows that for simulation purposes, the solution to the problem is naturally obtained in the joint space; in fact, the expression in (7.42) allows the computation of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ which, substituted into the direct kinematics equations in ((2.82), (3.62), (3.98), give $\boldsymbol{x}_e$, $\dot{\boldsymbol{x}}_e$, $\ddot{\boldsymbol{x}}_e$, respectively.

Formulation of an *inverse dynamics* problem in the operational space requires the determination of the joint torques $\boldsymbol{\tau}(t)$ that are needed to generate a specific motion assigned in terms of $\ddot{\boldsymbol{x}}_e(t)$, $\dot{\boldsymbol{x}}_e(t)$, $\boldsymbol{x}_e(t)$, for given end-effector forces $\boldsymbol{h}_e(t)$. A possible way of solution is to solve a complete inverse kinematics problem for (2.82), (3.62), (3.98), and then compute the required torques with the joint space inverse dynamics as in (7.42). Hence, for redundant manipulators, redundancy resolution is performed at kinematic level.

An alternative solution to the inverse dynamics problem consists of computing $\boldsymbol{\gamma}_A$ as in (7.133) and the joint torques $\boldsymbol{\tau}$ as in (3.111). In this way, however, the presence of redundant DOFs is not exploited at all, since the computed torques do not generate internal motions of the structure.

If it is desired to find a formal solution that allows redundancy resolution at dynamic level, it is necessary to determine those torques corresponding to the equivalent end-effector forces computed as in (7.133). By analogy with the differential kinematics solution (3.54), the expression of the torques to be determined will feature the presence of a minimum-norm term and a homogeneous term. Since the joint torques have to be computed, it is convenient to express the model (7.133) in terms of $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$. By recalling the positions (7.131), (7.132), the expression in (7.133) becomes

$$\boldsymbol{B}_A(\ddot{\boldsymbol{x}}_e - \dot{\boldsymbol{J}}_A\dot{\boldsymbol{q}}) + \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{g} = \boldsymbol{\gamma}_A - \boldsymbol{h}_A$$

and, in view of (3.98),

$$\boldsymbol{B}_A\boldsymbol{J}_A\ddot{\boldsymbol{q}} + \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{B}_A\boldsymbol{J}_A\boldsymbol{B}^{-1}\boldsymbol{g} = \boldsymbol{\gamma}_A - \boldsymbol{h}_A. \tag{7.137}$$

By setting

$$\bar{\boldsymbol{J}}_A(\boldsymbol{q}) = \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{B}_A(\boldsymbol{q}), \tag{7.138}$$

the expression in (7.137) becomes

$$\bar{\boldsymbol{J}}_A^T(\boldsymbol{B}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g}) = \boldsymbol{\gamma}_A - \boldsymbol{h}_A. \tag{7.139}$$

At this point, from the joint space dynamic model in (7.42), it is easy to recognize that (7.139) can be written as

$$\bar{\boldsymbol{J}}_A^T(\boldsymbol{\tau} - \boldsymbol{J}_A^T\boldsymbol{h}_A) = \boldsymbol{\gamma}_A - \boldsymbol{h}_A$$

from which

$$\bar{\boldsymbol{J}}_A^T\boldsymbol{\tau} = \boldsymbol{\gamma}_A. \tag{7.140}$$

The general solution to (7.140) is of the form (see Problem 7.10)

$$\boldsymbol{\tau} = \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{\gamma}_A + \big(\boldsymbol{I}_n - \boldsymbol{J}_A^T(\boldsymbol{q})\bar{\boldsymbol{J}}_A^T(\boldsymbol{q})\big)\boldsymbol{\tau}_0, \tag{7.141}$$

that can be derived by observing that $\boldsymbol{J}_A^T$ in (7.138) is a *right pseudo-inverse* of $\bar{\boldsymbol{J}}_A^T$ weighted by the inverse of the inertia matrix $\boldsymbol{B}^{-1}$. The $(n\times 1)$ vector of arbitrary torques $\boldsymbol{\tau}_0$ in (7.141) does not contribute to the end-effector forces, since it is projected in the null space of $\bar{\boldsymbol{J}}_A^T$.

To summarize, for given $\boldsymbol{x}_e$, $\dot{\boldsymbol{x}}_e$, $\ddot{\boldsymbol{x}}_e$ and $\boldsymbol{h}_A$, the expression in (7.133) allows the computation of $\boldsymbol{\gamma}_A$. Then, (7.141) gives the torques $\boldsymbol{\tau}$ which, besides executing the assigned end-effector motion, generate internal motions of the structure to be employed for handling redundancy at dynamic level through a suitable choice of $\boldsymbol{\tau}_0$.

7.9 Dynamic Manipulability Ellipsoid

The availability of the dynamic model allows formulation of the *dynamic manipulability ellipsoid* which provides a useful tool for manipulator dynamic performance analysis. This can be used for mechanical structure design as well as for seeking optimal manipulator configurations.

Consider the set of joint torques of constant (unit) norm

$$\boldsymbol{\tau}^T\boldsymbol{\tau} = 1 \tag{7.142}$$

describing the points on the surface of a sphere. It is desired to describe the operational space accelerations that can be generated by the given set of joint torques.

For studying dynamic manipulability, suppose to consider the case of a manipulator standing still ($\dot{\boldsymbol{q}} = \boldsymbol{0}$), not in contact with the environment ($\boldsymbol{h}_e = \boldsymbol{0}$). The simplified model is

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}. \tag{7.143}$$

The joint accelerations $\ddot{\boldsymbol{q}}$ can be computed from the second-order differential kinematics that can be obtained by differentiating (3.39), and imposing successively $\dot{\boldsymbol{q}} = \boldsymbol{0}$, leading to

$$\dot{\boldsymbol{v}}_e = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}. \tag{7.144}$$

Solving for minimum-norm accelerations only, for a *nonsingular Jacobian*, and substituting in (7.143) yields the expression of the torques

$$\boldsymbol{\tau} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{v}}_e + \boldsymbol{g}(\boldsymbol{q}) \tag{7.145}$$

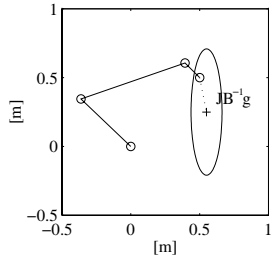

Fig. 7.15. Effect of gravity on the dynamic manipulability ellipsoid for a three-link planar arm

needed to derive the ellipsoid. In fact, substituting (7.145) into (7.142) gives

$$\big(\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{v}}_e+\boldsymbol{g}(\boldsymbol{q})\big)^T\big(\boldsymbol{B}(\boldsymbol{q})\boldsymbol{J}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{v}}_e+\boldsymbol{g}(\boldsymbol{q})\big)=1.$$

The vector on the right-hand side of (7.145) can be rewritten as

$$\begin{aligned}\boldsymbol{B}\boldsymbol{J}^{\dagger}\dot{\boldsymbol{v}}_e+\boldsymbol{g}&=\boldsymbol{B}(\boldsymbol{J}^{\dagger}\dot{\boldsymbol{v}}_e+\boldsymbol{B}^{-1}\boldsymbol{g})\\&=\boldsymbol{B}(\boldsymbol{J}^{\dagger}\dot{\boldsymbol{v}}_e+\boldsymbol{B}^{-1}\boldsymbol{g}+\boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g}-\boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})\\&=\boldsymbol{B}\big(\boldsymbol{J}^{\dagger}\dot{\boldsymbol{v}}_e+\boldsymbol{J}^{\dagger}\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g}+(\boldsymbol{I}_n-\boldsymbol{J}^{\dagger}\boldsymbol{J})\boldsymbol{B}^{-1}\boldsymbol{g}\big),\end{aligned}\tag{7.146}$$

where the dependence on $\boldsymbol{q}$ has been omitted. According to what was done for solving (7.144), one can neglect the contribution of the accelerations given by $\boldsymbol{B}^{-1}\boldsymbol{g}$ which are in the null space of $\boldsymbol{J}$ and then produce no end-effector acceleration. Hence, (7.146) becomes

$$\boldsymbol{B}\boldsymbol{J}^{\dagger}\dot{\boldsymbol{v}}_e+\boldsymbol{g}=\boldsymbol{B}\boldsymbol{J}^{\dagger}(\dot{\boldsymbol{v}}_e+\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})\tag{7.147}$$

and the dynamic manipulability ellipsoid can be expressed in the form

$$(\dot{\boldsymbol{v}}_e+\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})^T\boldsymbol{J}^{\dagger T}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{J}^{\dagger}(\dot{\boldsymbol{v}}_e+\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})=1.\tag{7.148}$$

The core of the quadratic form $\boldsymbol{J}^{\dagger T}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{J}^{\dagger}$ depends on the geometrical and inertial characteristics of the manipulator and determines the volume and principal axes of the ellipsoid. The vector $-\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g}$, describing the contribution of gravity, produces a constant translation of the centre of the ellipsoid (for each manipulator configuration) with respect to the origin of the reference frame; see the example in Fig. 7.15 for a three-link planar arm.

The meaning of the dynamic manipulability ellipsoid is conceptually similar to that of the ellipsoids considered with reference to kineto-statics duality. In fact, the distance of a point on the surface of the ellipsoid from the end-effector gives a measure of the accelerations which can be imposed to the end-effector along the given direction, with respect to the constraint (7.142). With reference to Fig. 7.15, it is worth noticing how the presence of gravity acceleration allows the execution of larger accelerations downward, as natural to predict.

In the case of a nonredundant manipulator, the ellipsoid reduces to

$$(\dot{\boldsymbol{v}}_e+\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})^T\boldsymbol{J}^{-T}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{J}^{-1}(\dot{\boldsymbol{v}}_e+\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{g})=1.\tag{7.149}$$

Bibliography

The derivation of the dynamic model for rigid manipulators can be found in several classical robotics texts, such as [180, 10, 248, 53, 217, 111].

The first works on the computation of the dynamic model of open-chain manipulators based on the Lagrange formulation are [234, 19, 221, 236]. A computationally efficient formulation is presented in [96].

Dynamic model computation for robotic systems having a closed-chain or a tree kinematic structure can be found in [11, 144] and [112], respectively. Joint friction models are analyzed in [9].

The notable properties of the dynamic model deriving from the principle of energy conservation are underlined in [213], on the basis of the work in [119]. Algorithms to find the parameterization of the dynamic model in terms of a minimum number of parameters are considered in [115], which utilizes the results in [166]. Methods for symbolic computation of those parameters are presented in [85] for open kinematic chains and [110] for closed kinematic chains. Parameter identification methods based on least-squares techniques are given in [13].

The Newton–Euler formulation is proposed in [172], and a computationally efficient version for inverse dynamics can be found in [142]; an analogous formulation is employed for direct dynamics computation in [237]. The Lagrange and Newton–Euler formulations are compared by a computational viewpoint in [211], while they are utilized in [201] for dynamic model computation with inclusion of inertial and gyroscopic effects of actuators. Efficient algorithms for direct dynamics computation are given in [76, 77].

The trajectory dynamic scaling technique is presented in [97]. The operational space dynamic model is illustrated in [114] and the concept of weighted pseudo-inverse of the inertia matrix is introduced in [78]. The manipulability ellipsoids are analyzed in [246, 38].

Problems

7.1. Find the dynamic model of a two-link Cartesian arm in the case when the second joint axis forms an angle of $\pi/4$ with the first joint axis; compare the result with the model of the manipulator in Fig. 7.3.

7.2. For the two-link planar arm of Sect. 7.3.2, prove that with a different choice of the matrix $\boldsymbol{C}$, (7.49) holds true while (7.48) does not.

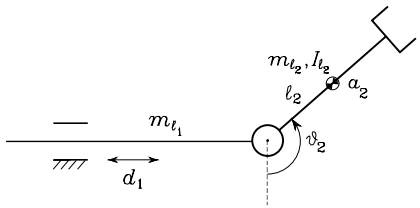

Fig. 7.16. Two-link planar arm with a prismatic joint and a revolute joint

7.3. Find the dynamic model of the SCARA manipulator in Fig. 2.36.

7.4. For the planar arm of Sect. 7.3.2, find a minimal parameterization of the dynamic model in (7.82).

7.5. Find the dynamic model of the two-link planar arm with a prismatic joint and a revolute joint in Fig. 7.16 with the Lagrange formulation. Then, consider the addition of a concentrated tip payload of mass m_L, and express the resulting model in a linear form with respect to a suitable set of dynamic parameters as in (7.81).

7.6. For the two-link planar arm of Fig. 7.4, find the dynamic model with the Lagrange formulation when the absolute angles with respect to the base frame are chosen as generalized coordinates. Discuss the result in view of a comparison with the model derived in (7.82).

7.7. Compute the joint torques for the two-link planar arm of Fig. 7.4 with the data and along the trajectories of Example 7.2, in the case of tip forces $\boldsymbol{f} = [\,500 \quad 500\,]^T$ N.

7.8. Find the dynamic model of the two-link planar arm with a prismatic joint and a revolute joint in Fig. 7.16 by using the recursive Newton–Euler algorithm.

7.9. Show that for the operational space dynamic model (7.133) a skew-symmetry property holds which is analogous to (7.48).

7.10. Show how to obtain the general solution to (7.140) in the form (7.141).

7.11. For a nonredundant manipulator, compute the relationship between the dynamic manipulability measure that can be defined for the dynamic manipulability ellipsoid and the manipulability measure defined in (3.56).

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

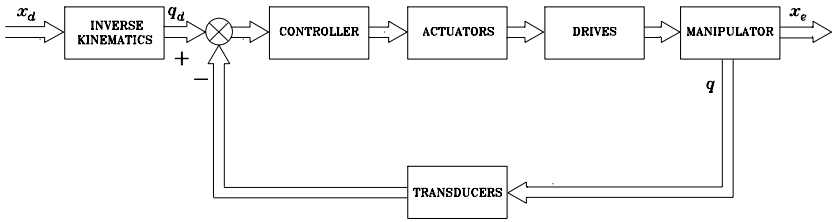

Fig. 8.1. General scheme of joint space control

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

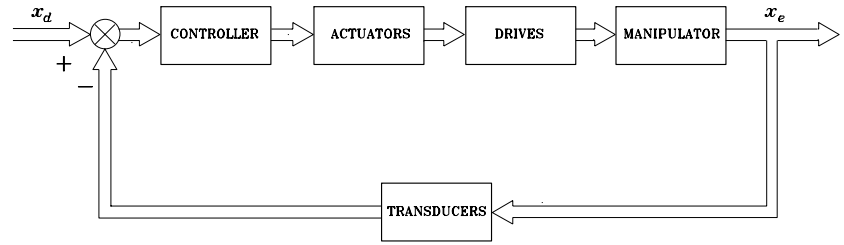

Fig. 8.2. General scheme of operational space control

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:

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

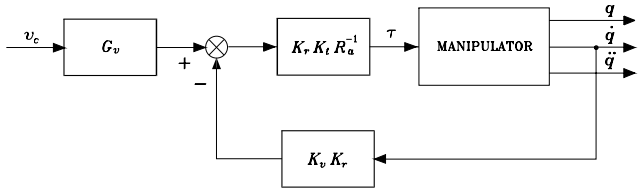

Fig. 8.3. Block scheme of the manipulator and drives system as a voltage-controlled system

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.

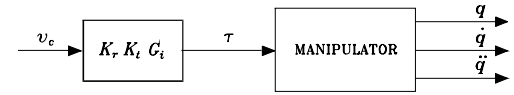

Fig. 8.4. Block scheme of the manipulator and drives system as a torque-controlled system

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

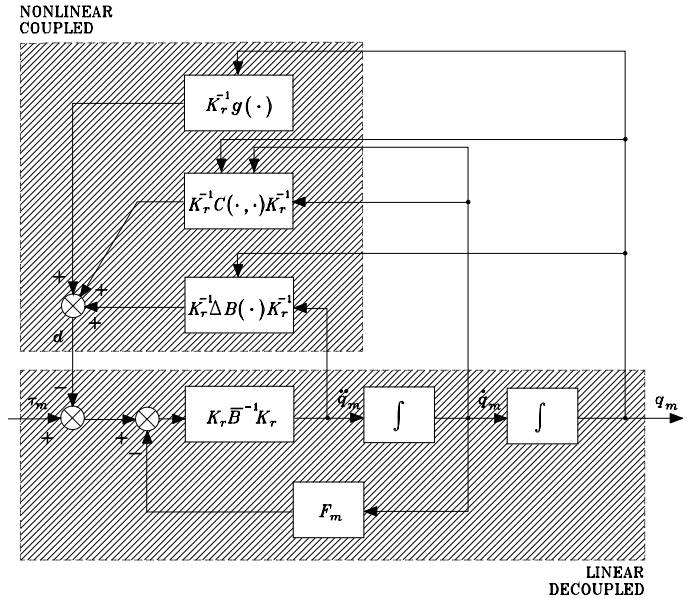

Fig. 8.5. Block scheme of the system of manipulator with drives

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).

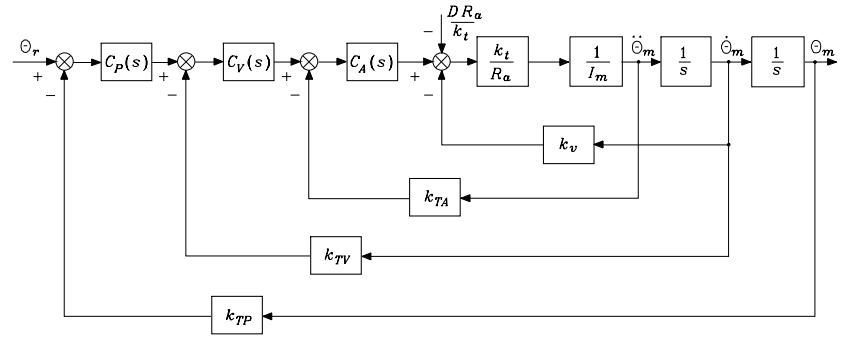

Fig. 8.6. Block scheme of general independent joint control

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.

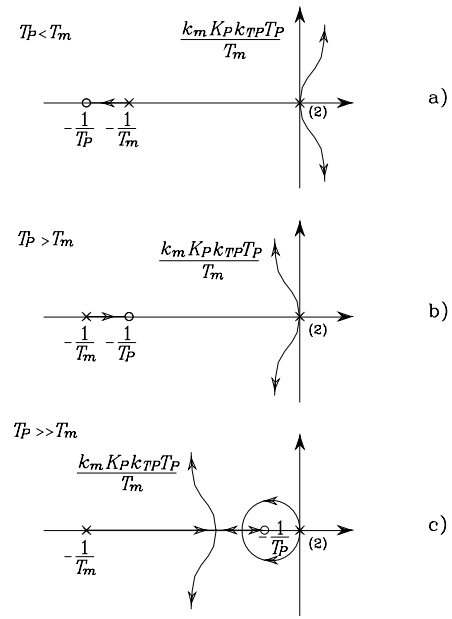

Fig. 8.7. Root loci for the position feedback control scheme

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

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

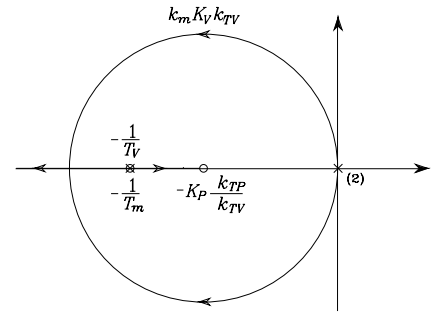

Fig. 8.8. Root locus for the position and velocity feedback control scheme

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:

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

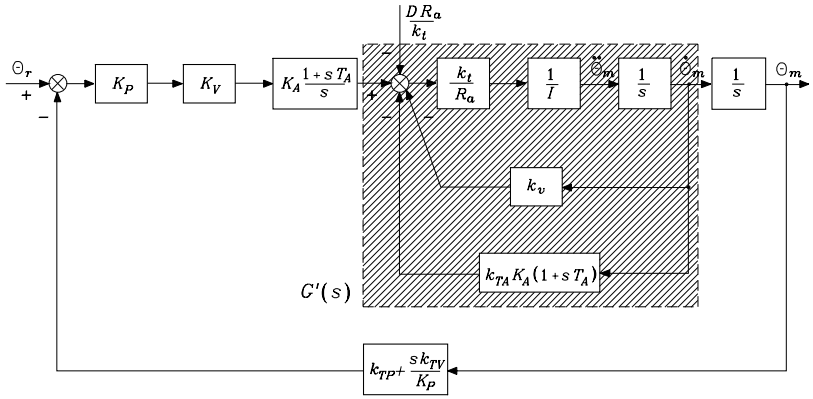

Fig. 8.9. Block scheme of position, velocity and acceleration feedback control

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

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 $\zeta\omega_n < 1/2T_m$.

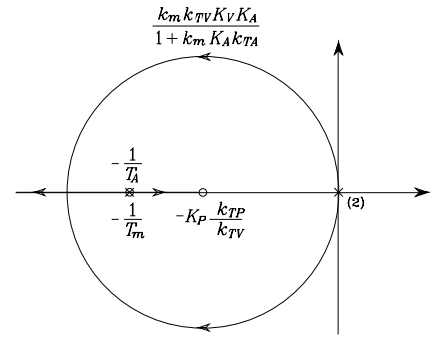

Fig. 8.10. Root locus for the position, velocity and acceleration feedback control scheme

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

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

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

$$T_A = T_m,$$

or by making

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

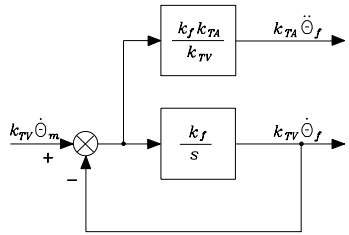

Fig. 8.11. Block scheme of a first-order filter

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),

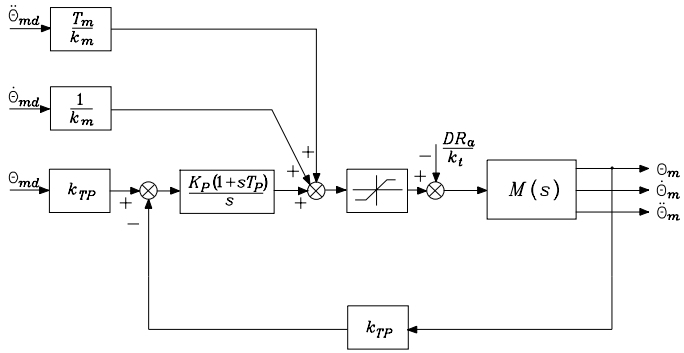

Fig. 8.12. Block scheme of position feedback control with decentralized feedforward compensation

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.

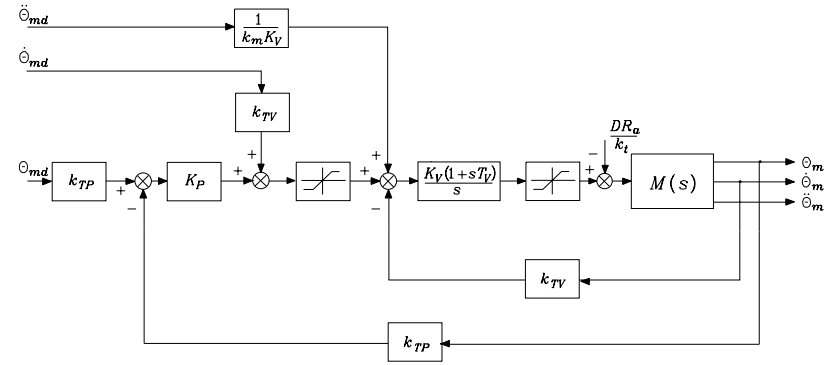

Fig. 8.13. Block scheme of position and velocity feedback control with decentralized feedforward compensation

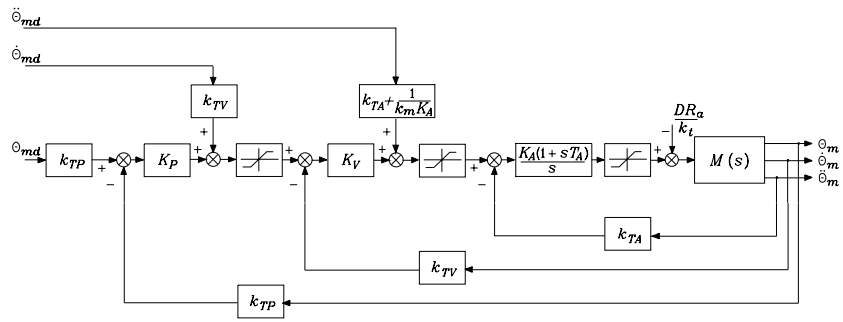

Fig. 8.14. Block scheme of position, velocity and acceleration feedback control with decentralized feedforward compensation

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.

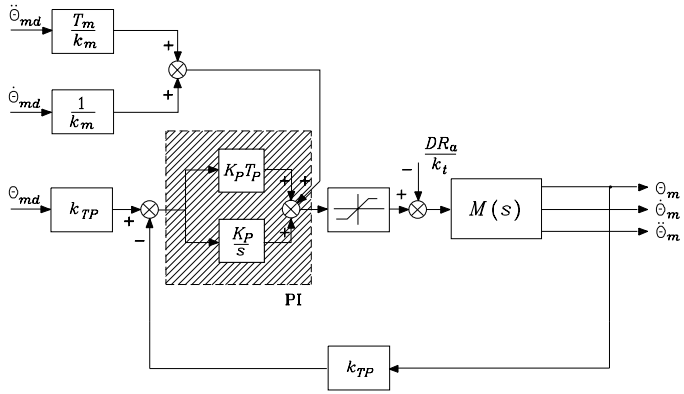

Fig. 8.15. Equivalent control scheme of PI type

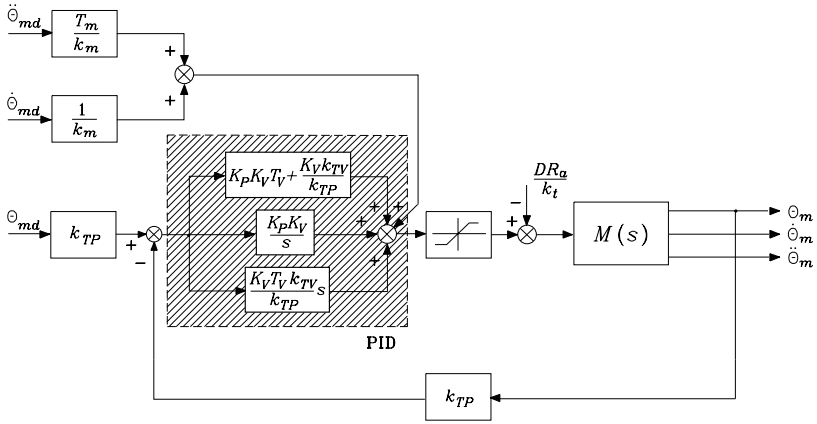

Fig. 8.16. Equivalent control scheme of PID type

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 $I_m\ddot{\vartheta}_d +$

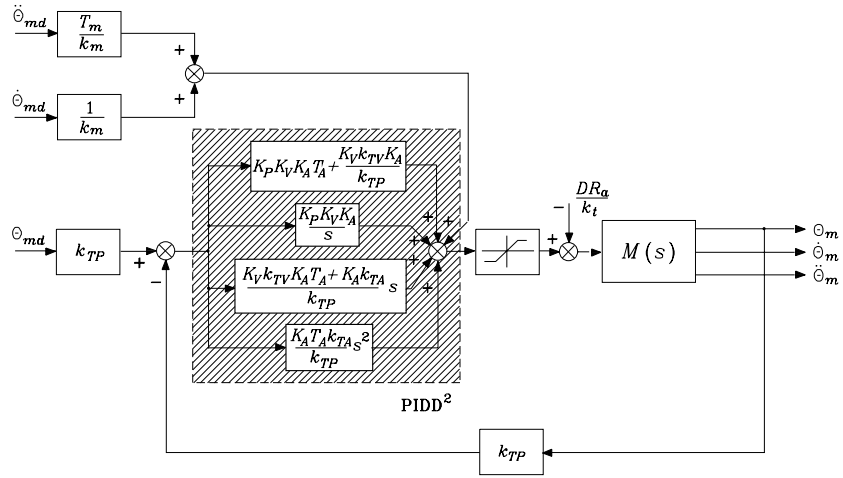

Fig. 8.17. Equivalent control scheme of PIDD^2 type

$F_m\dot{\vartheta}_d$ 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

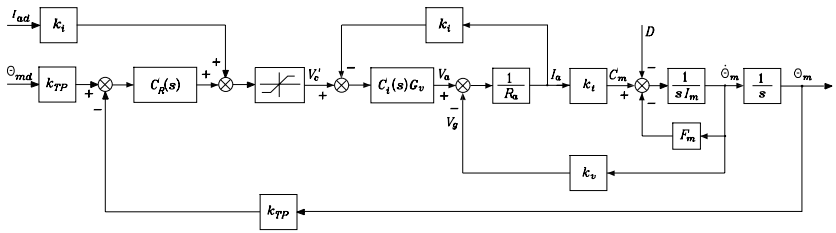

Fig. 8.18. Control scheme with current-controlled drive and current feedforward action

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

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

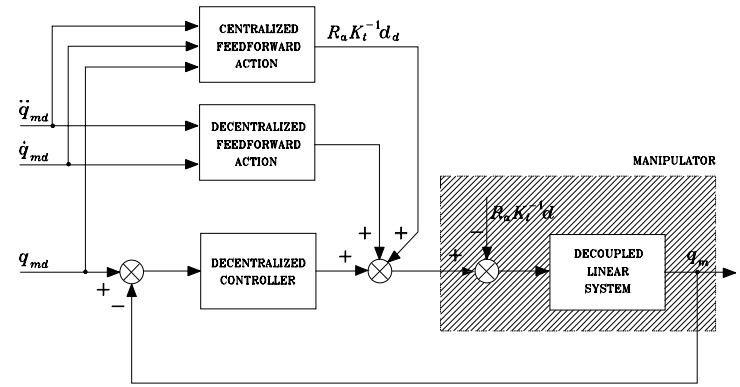

Fig. 8.19. Block scheme of computed torque feedforward control

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

$$\frac{E(s)}{D(s)} = \frac{\frac{R_a}{k_t}s}{a_2' s^3 + a_1' s^2 + a_0' s + a_{-1}'},$$

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

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

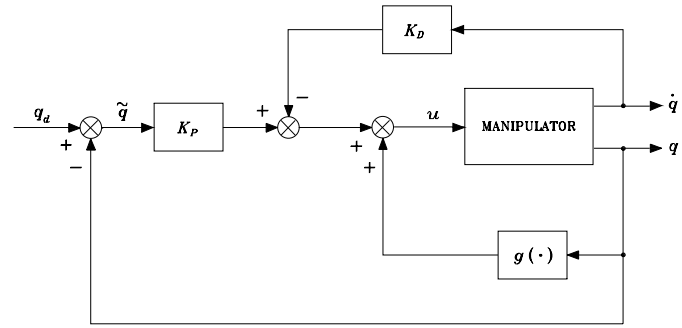

Fig. 8.20. Block scheme of joint space PD control with gravity compensation

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

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

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

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

and then

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

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.

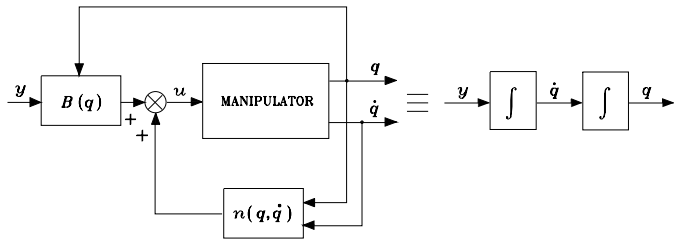

Fig. 8.21. Exact linearization performed by inverse dynamics control

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.

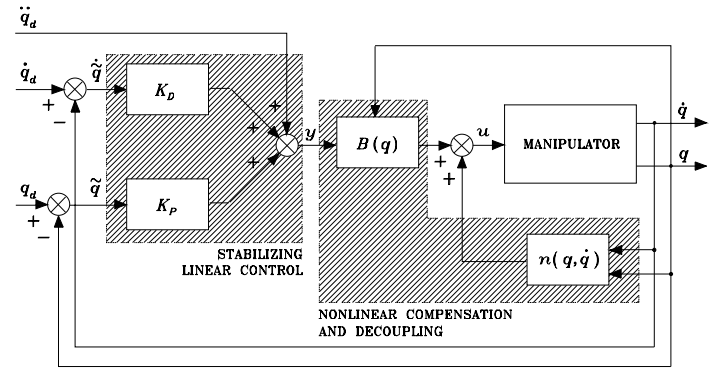

Fig. 8.22. Block scheme of joint space inverse dynamics control

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

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

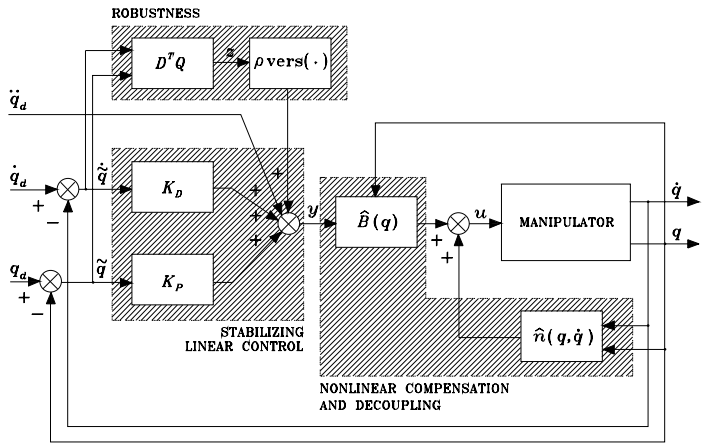

Fig. 8.23. Block scheme of joint space robust control

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 $\ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{K}_P\tilde{\boldsymbol{q}}$ 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

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

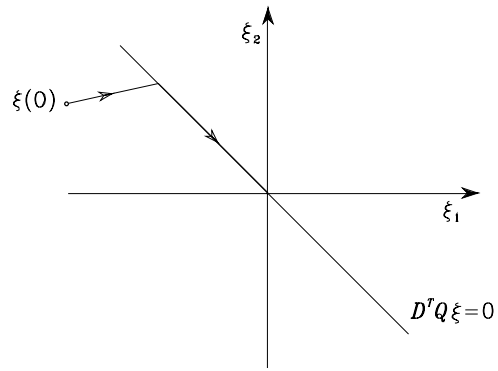

Fig. 8.24. Error trajectory with robust control

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

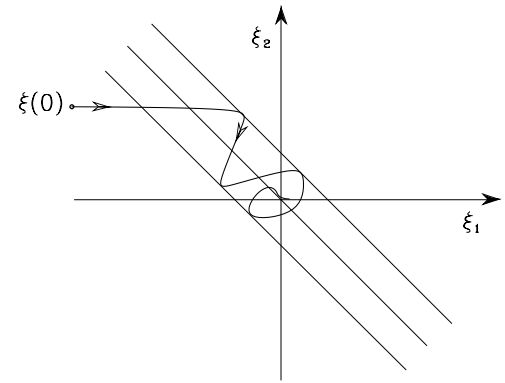

Fig. 8.25. Error trajectory with robust control and chattering elimination

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

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

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

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

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

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

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

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

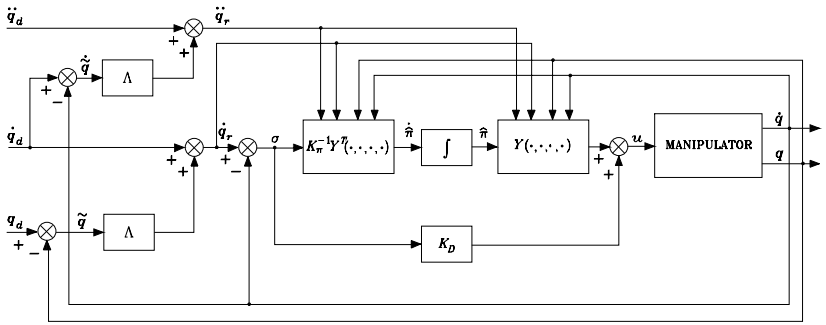

Fig. 8.26. Block scheme of joint space adaptive control

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 $\boldsymbol{Y}\widehat{\boldsymbol{\pi}}\approx\boldsymbol{Y}\boldsymbol{\pi}$.

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.

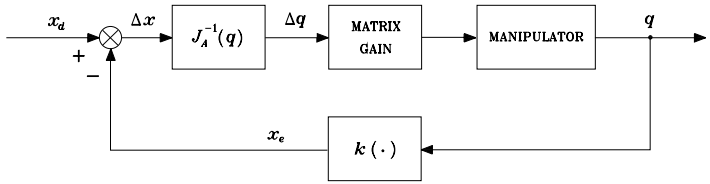

Fig. 8.27. Block scheme of Jacobian inverse control

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

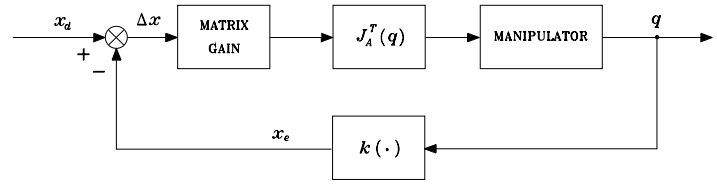

Fig. 8.28. Block scheme of Jacobian transpose control

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 $\boldsymbol{\Delta q}$ 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 $\boldsymbol{\Delta x}$. In other words, the resulting force drives the end-effector along a direction so as to reduce $\boldsymbol{\Delta x}$. 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

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

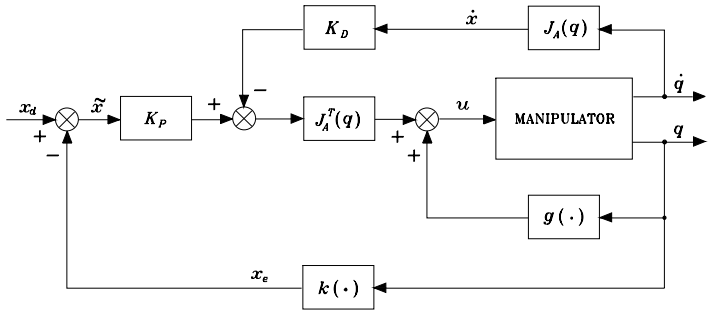

Fig. 8.29. Block scheme of operational space PD control with gravity compensation

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

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

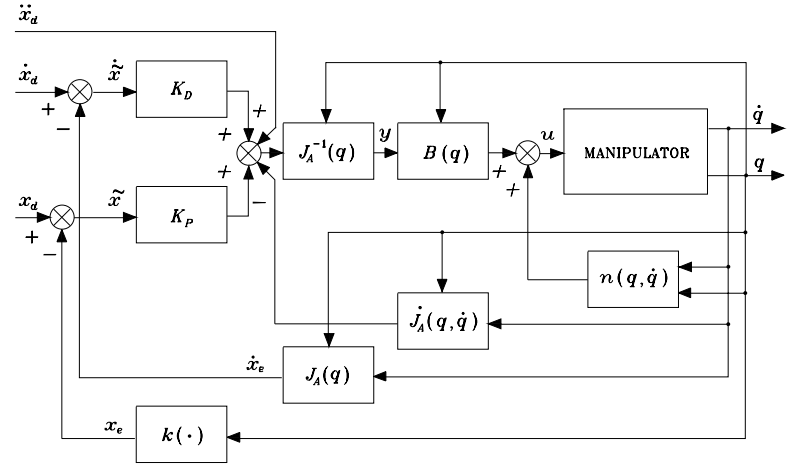

Fig. 8.30. Block scheme of operational space inverse dynamics control

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 (analytic) 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:

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

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:

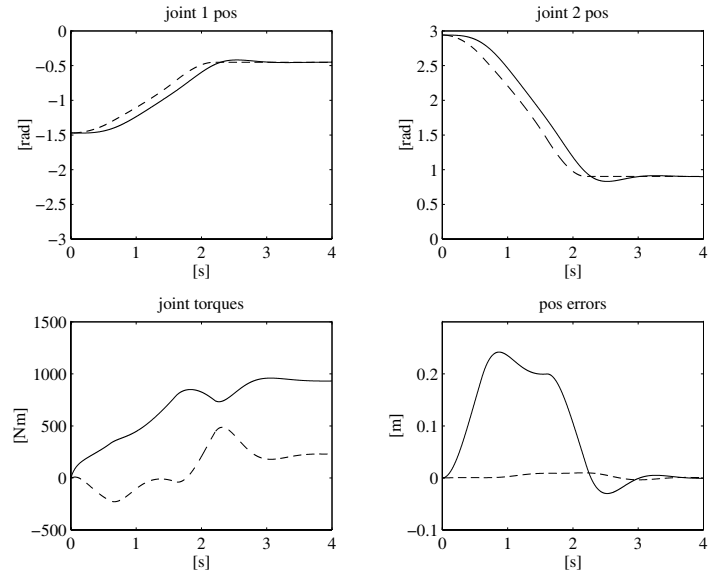

Fig. 8.31. Time history of the joint positions and torques and of the tip position errors for the *fast* trajectory with control scheme **A**

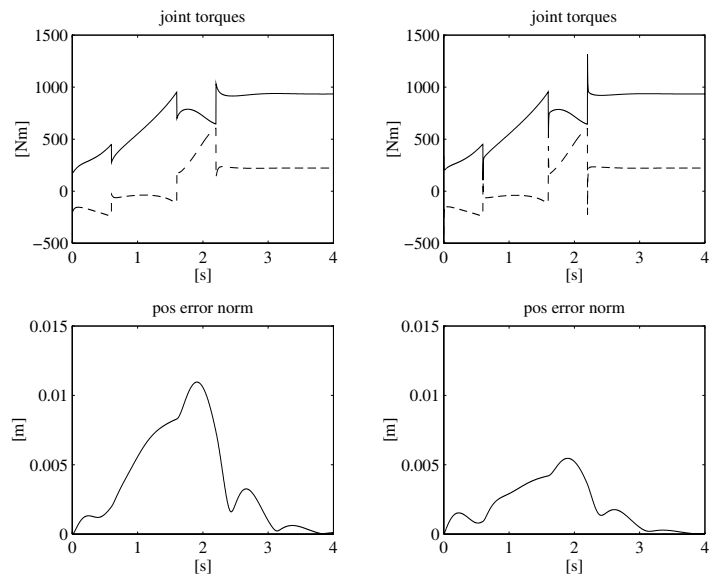

Fig. 8.32. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory; *left*: with control scheme **C**, *right*: with control scheme **D**

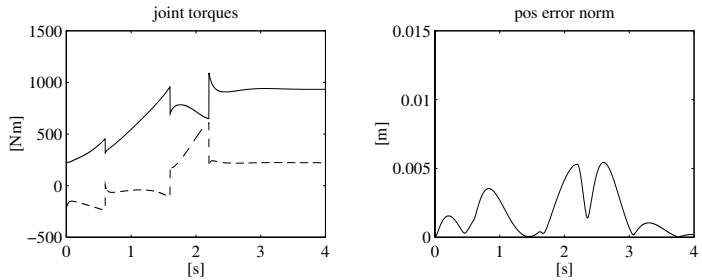

Fig. 8.33. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **E**

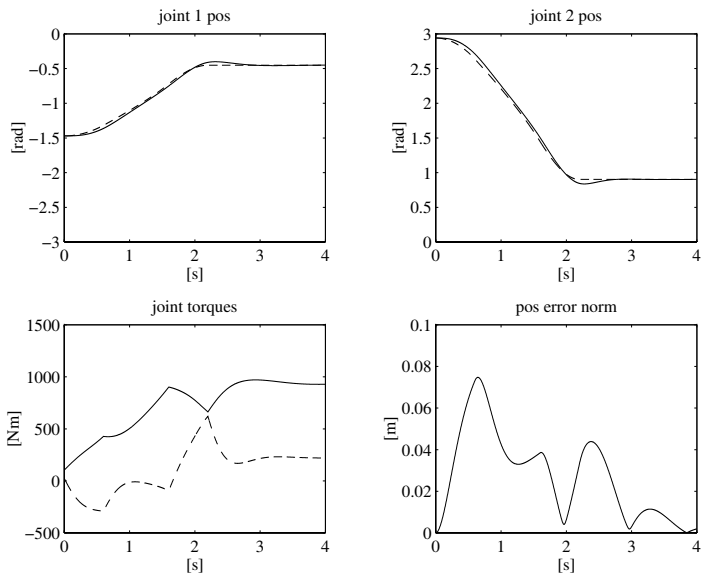

Fig. 8.34. Time history of the joint positions and torques and of the norm of tip position error for the *fast* trajectory with control scheme **F**

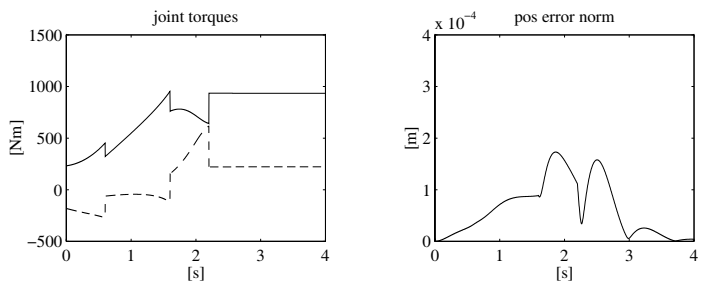

Fig. 8.35. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **G**

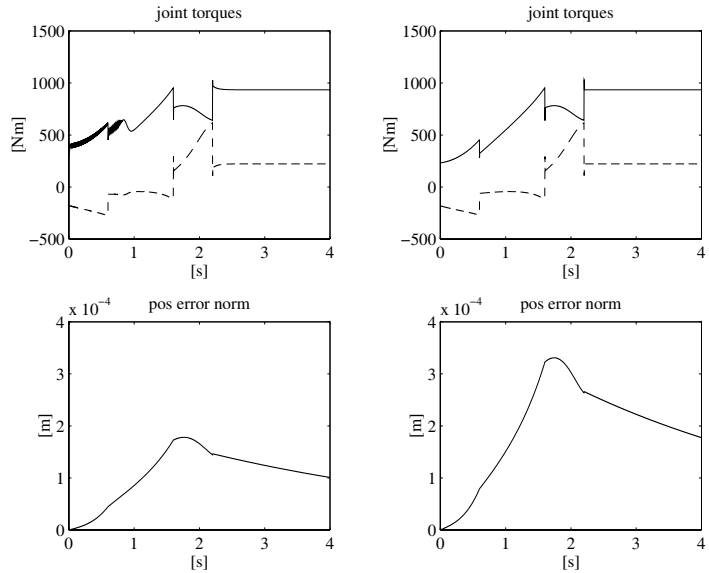

Fig. 8.36. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory; *left*: with control scheme **H**, *right*: with control scheme **I**

- 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.

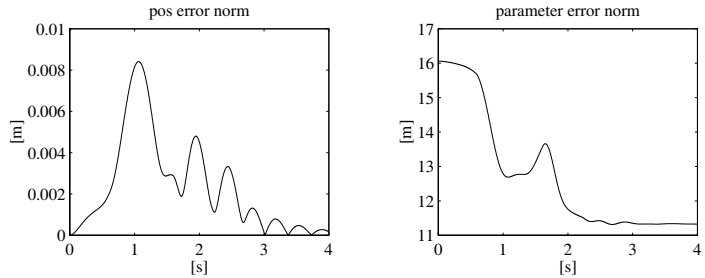

Fig. 8.37. Time history of the norm of tip position error and of the norm of parameter error vector for the *fast* trajectory with control scheme **J**

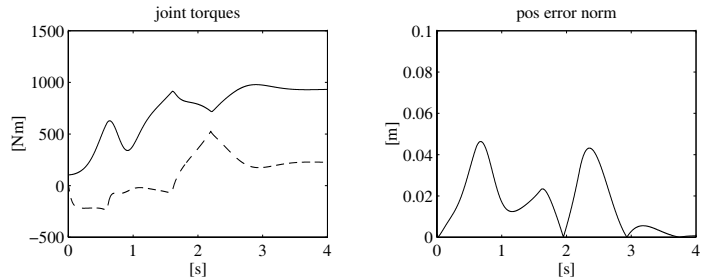

Fig. 8.38. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **K**

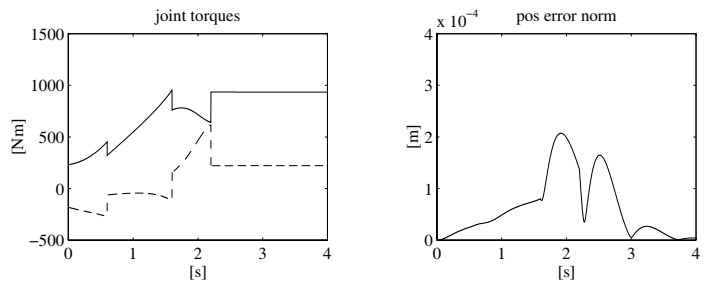

Fig. 8.39. Time history of the joint torques and of the norm of tip position error for the *fast* trajectory with control scheme **L**

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

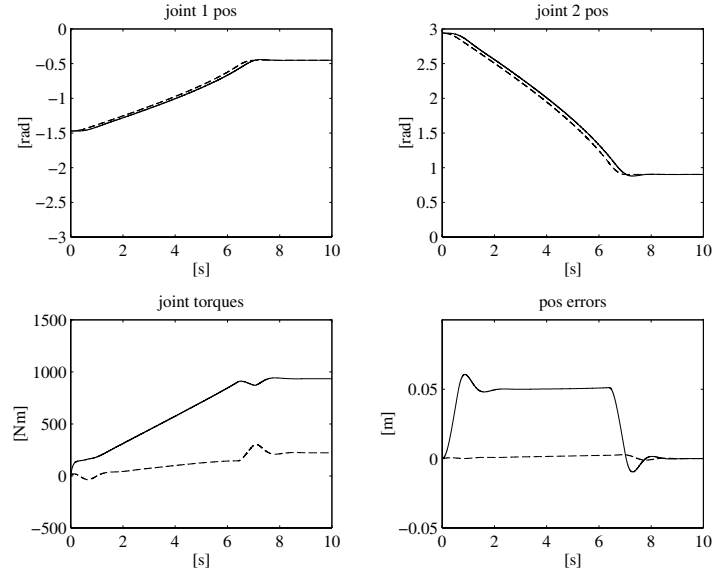

Fig. 8.40. Time history of the joint positions and torques and of the tip position errors for the *slow* trajectory with control scheme **A**

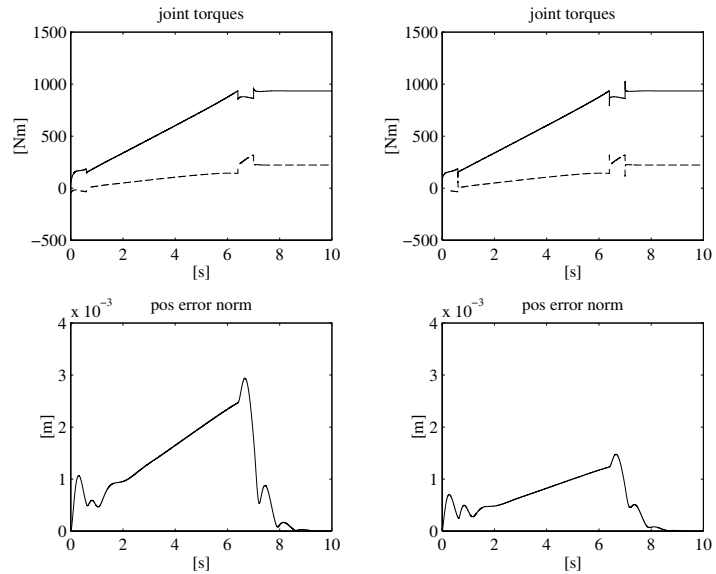

Fig. 8.41. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory; *left*: with control scheme **C**, *right*: with control scheme **D**

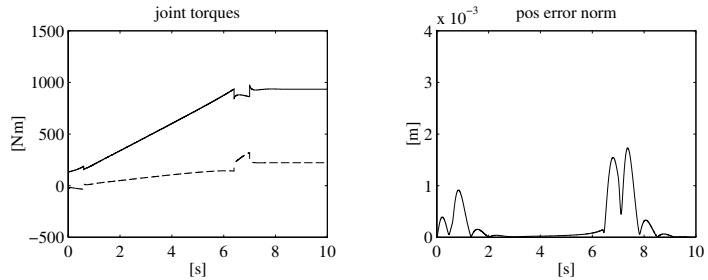

Fig. 8.42. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **E**

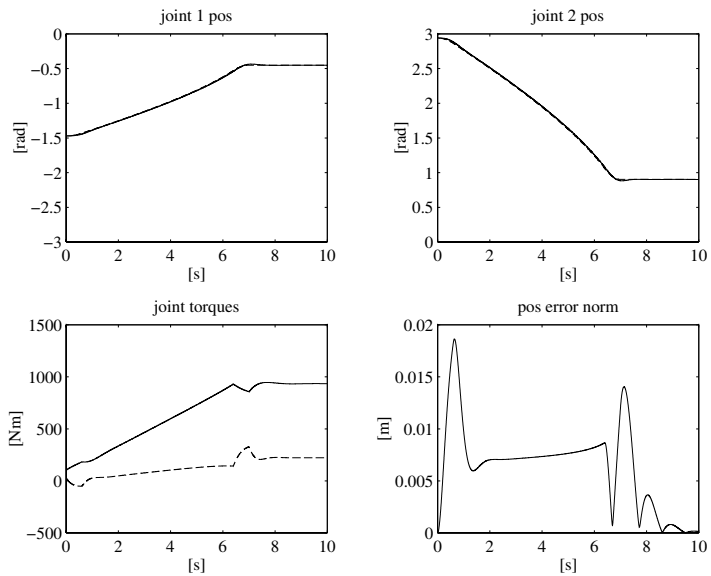

Fig. 8.43. Time history of the joint positions and torques and of the norm of tip position error for the *slow* trajectory with control scheme **F**

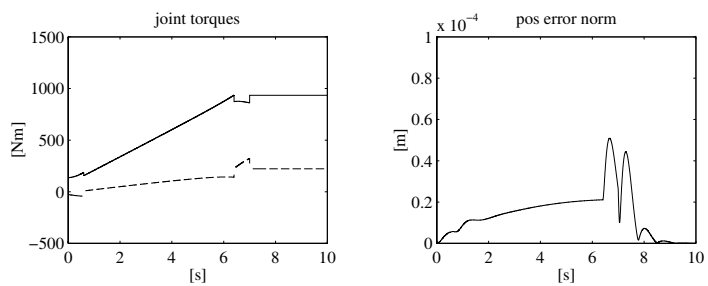

Fig. 8.44. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **G**

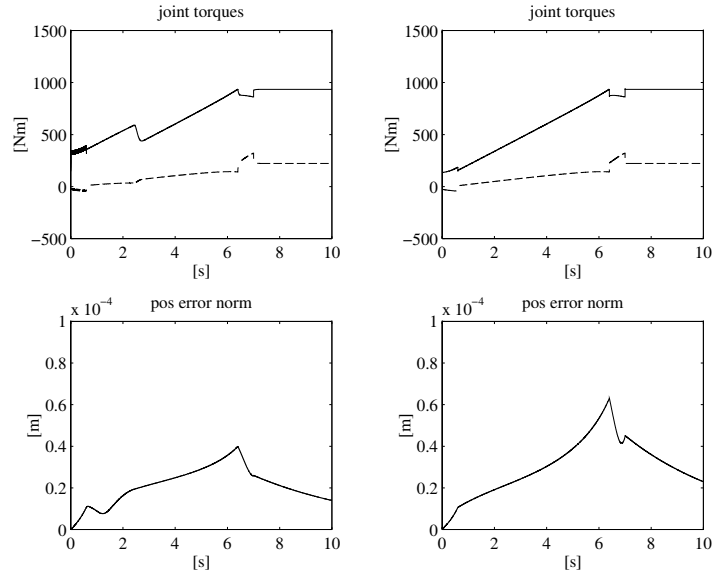

Fig. 8.45. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory; *left*: with control scheme **H**, *right*: with control scheme **I**

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-

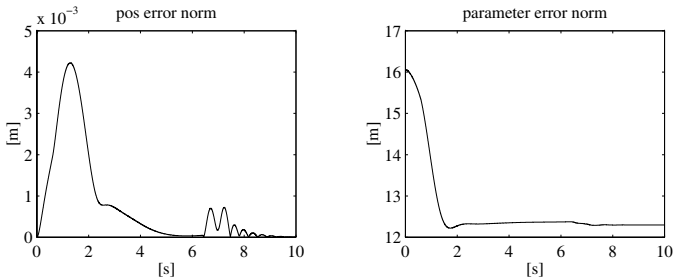

Fig. 8.46. Time history of the norm of tip position error and of the norm of parameter error vector for the *slow* trajectory with control scheme **J**

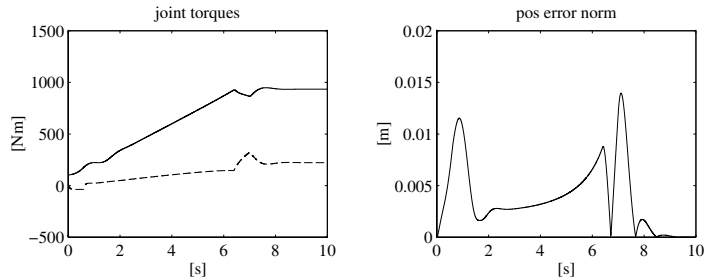

Fig. 8.47. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **K**

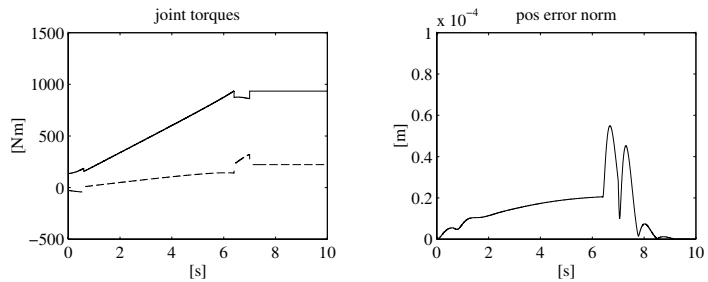

Fig. 8.48. Time history of the joint torques and of the norm of tip position error for the *slow* trajectory with control scheme **L**

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.

Appendices

A

Linear Algebra

Since modelling and control of robot manipulators requires an extensive use of *matrices* and *vectors* as well as of matrix and vector *operations*, the goal of this appendix is to provide a brush-up of *linear algebra*.

A.1 Definitions

A *matrix* of dimensions $(m \times n)$, with m and n positive integers, is an array of elements a_{ij} arranged into m *rows* and n *columns*:

$$\boldsymbol{A} = [a_{ij}]_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}. \tag{A.1}$$

If $m = n$, the matrix is said to be *square*; if $m < n$, the matrix has more columns than rows; if $m > n$ the matrix has more rows than columns. Further, if $n = 1$, the notation (A.1) is used to represent a (column) vector $\boldsymbol{a}$ of dimensions $(m \times 1)$;[1] the elements a_i are said to be vector components.

A square matrix $\boldsymbol{A}$ of dimensions $(n \times n)$ is said to be *upper triangular* if $a_{ij} = 0$ for $i > j$:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ 0 & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & a_{nn} \end{bmatrix};$$

the matrix is said to be *lower triangular* if $a_{ij} = 0$ for $i < j$.

[1] According to standard mathematical notation, small boldface is used to denote vectors while capital boldface is used to denote matrices. Scalars are denoted by roman characters.

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *diagonal* if $a_{ij} = 0$ for $i \neq j$, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix} = \text{diag}\{a_{11}, a_{22}, \dots, a_{nn}\}.$$

If an $(n \times n)$ diagonal matrix has all unit elements on the diagonal ($a_{ii} = 1$), the matrix is said to be *identity* and is denoted by $\boldsymbol{I}_n$.[2] A matrix is said to be *null* if all its elements are null and is denoted by $\boldsymbol{O}$. The null column vector is denoted by $\boldsymbol{0}$.

The *transpose* $\boldsymbol{A}^T$ of a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ is the matrix of dimensions $(n \times m)$ which is obtained from the original matrix by interchanging its rows and columns:

$$\boldsymbol{A}^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}. \tag{A.2}$$

The transpose of a column vector $\boldsymbol{a}$ is the row vector $\boldsymbol{a}^T$.

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *symmetric* if $\boldsymbol{A}^T = \boldsymbol{A}$, and thus $a_{ij} = a_{ji}$:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{12} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{nn} \end{bmatrix}.$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *skew-symmetric* if $\boldsymbol{A}^T = -\boldsymbol{A}$, and thus $a_{ij} = -a_{ji}$ for $i \neq j$ and $a_{ii} = 0$, leading to

$$\boldsymbol{A} = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ -a_{12} & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{1n} & -a_{2n} & \dots & 0 \end{bmatrix}.$$

A *partitioned* matrix is a matrix whose elements are matrices (*blocks*) of proper dimensions:

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \dots & \boldsymbol{A}_{1n} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \dots & \boldsymbol{A}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{m1} & \boldsymbol{A}_{m2} & \dots & \boldsymbol{A}_{mn} \end{bmatrix}.$$

[2] Subscript n is usually omitted if the dimensions are clear from the context.

A partitioned matrix may be block-triangular or block-diagonal. Special partitions of a matrix are that by columns

$$\boldsymbol{A} = [\, \boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \dots \quad \boldsymbol{a}_n \,]$$

and that by rows

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_m^T \end{bmatrix}.$$

Given a square matrix $\boldsymbol{A}$ of dimensions $(n \times n)$, the *algebraic complement* $\boldsymbol{A}_{(ij)}$ of element a_{ij} is the matrix of dimensions $((n-1) \times (n-1))$ which is obtained by eliminating row i and column j of matrix $\boldsymbol{A}$.

A.2 Matrix Operations

The *trace* of an $(n \times n)$ square matrix $\boldsymbol{A}$ is the sum of the elements on the diagonal:

$$\text{Tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}. \tag{A.3}$$

Two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the same dimensions $(m \times n)$ are equal if $a_{ij} = b_{ij}$. If $\boldsymbol{A}$ and $\boldsymbol{B}$ are two matrices of the same dimensions, their *sum* is the matrix

$$\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B} \tag{A.4}$$

whose elements are given by $c_{ij} = a_{ij} + b_{ij}$. The following properties hold:

$$\begin{aligned} &\boldsymbol{A} + \boldsymbol{O} = \boldsymbol{A} \\ &\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A} \\ &(\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C}). \end{aligned}$$

Notice that two matrices of the same dimensions and partitioned in the same way can be summed formally by operating on the blocks in the same position and treating them like elements.

The *product of a scalar α by an $(m \times n)$ matrix* $\boldsymbol{A}$ is the matrix $\alpha\boldsymbol{A}$ whose elements are given by αa_{ij}. If $\boldsymbol{A}$ is an $(n \times n)$ diagonal matrix with all equal elements on the diagonal ($a_{ii} = a$), it follows that $\boldsymbol{A} = a\boldsymbol{I}_n$.

If $\boldsymbol{A}$ is a square matrix, one may write

$$\boldsymbol{A} = \boldsymbol{A}_s + \boldsymbol{A}_a \tag{A.5}$$

where

$$\boldsymbol{A}_s = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^T) \tag{A.6}$$

is a symmetric matrix representing the *symmetric* part of $\boldsymbol{A}$, and

$$\boldsymbol{A}_a = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^T) \tag{A.7}$$

is a skew-symmetric matrix representing the *skew-symmetric* part of $\boldsymbol{A}$.

The row-by-column *product* of a matrix $\boldsymbol{A}$ of dimensions $(m \times p)$ by a matrix $\boldsymbol{B}$ of dimensions $(p \times n)$ is the matrix of dimensions $(m \times n)$

$$\boldsymbol{C} = \boldsymbol{AB} \tag{A.8}$$

whose elements are given by $c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}$. The following properties hold:

$$\begin{aligned}
&\boldsymbol{A} = \boldsymbol{A}\boldsymbol{I}_p = \boldsymbol{I}_m\boldsymbol{A} \\
&\boldsymbol{A}(\boldsymbol{BC}) = (\boldsymbol{AB})\boldsymbol{C} \\
&\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C}) = \boldsymbol{AB} + \boldsymbol{AC} \\
&(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{C} = \boldsymbol{AC} + \boldsymbol{BC} \\
&(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T.
\end{aligned}$$

Notice that, in general, $\boldsymbol{AB} \neq \boldsymbol{BA}$, and $\boldsymbol{AB} = \boldsymbol{O}$ does not imply that $\boldsymbol{A} = \boldsymbol{O}$ or $\boldsymbol{B} = \boldsymbol{O}$; further, notice that $\boldsymbol{AC} = \boldsymbol{BC}$ does not imply that $\boldsymbol{A} = \boldsymbol{B}$.

If an $(m \times p)$ matrix $\boldsymbol{A}$ and a $(p \times n)$ matrix $\boldsymbol{B}$ are partitioned in such a way that the number of blocks for each row of $\boldsymbol{A}$ is equal to the number of blocks for each column of $\boldsymbol{B}$, and the blocks $\boldsymbol{A}_{ik}$ and $\boldsymbol{B}_{kj}$ have dimensions compatible with product, the matrix product $\boldsymbol{AB}$ can be formally obtained by operating by rows and columns on the blocks of proper position and treating them like elements.

For an $(n \times n)$ *square* matrix $\boldsymbol{A}$, the *determinant* of $\boldsymbol{A}$ is the scalar given by the following expression, which holds $\forall i = 1, \ldots, n$:

$$\det(\boldsymbol{A}) = \sum_{j=1}^{n} a_{ij}(-1)^{i+j} \det\big(\boldsymbol{A}_{(ij)}\big). \tag{A.9}$$

The determinant can be computed according to any row i as in (A.9); the same result is obtained by computing it according to any column j. If $n = 1$, then $\det(a_{11}) = a_{11}$. The following property holds:

$$\det(\boldsymbol{A}) = \det(\boldsymbol{A}^T).$$

Moreover, interchanging two generic columns p and q of a matrix $\boldsymbol{A}$ yields

$$\det\big([\,\boldsymbol{a}_1 \ldots \boldsymbol{a}_p \ldots \boldsymbol{a}_q \ldots \boldsymbol{a}_n\,]\big) = -\det\big([\,\boldsymbol{a}_1 \ldots \boldsymbol{a}_q \ldots \boldsymbol{a}_p \ldots \boldsymbol{a}_n\,]\big).$$

As a consequence, if a matrix has two equal columns (rows), then its determinant is null. Also, it is $\det(\alpha\boldsymbol{A}) = \alpha^n \det(\boldsymbol{A})$.

Given an $(m \times n)$ matrix $\boldsymbol{A}$, the determinant of the square block obtained by selecting an equal number k of rows and columns is said to be k-order *minor* of matrix $\boldsymbol{A}$. The minors obtained by taking the *first* k rows and columns of $\boldsymbol{A}$ are said to be *principal* minors.

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices, then

$$\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B}). \tag{A.10}$$

If $\boldsymbol{A}$ is an $(n \times n)$ triangular matrix (in particular diagonal), then

$$\det(\boldsymbol{A}) = \prod_{i=1}^{n} a_{ii}.$$

More generally, if $\boldsymbol{A}$ is block-triangular with m blocks $\boldsymbol{A}_{ii}$ on the diagonal, then

$$\det(\boldsymbol{A}) = \prod_{i=1}^{m} \det(\boldsymbol{A}_{ii}).$$

A square matrix $\boldsymbol{A}$ is said to be *singular* when $\det(\boldsymbol{A}) = 0$.

The *rank* $\varrho(\boldsymbol{A})$ of a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ is the maximum integer r so that at least a non-null minor of order r exists. The following properties hold:

$$\begin{aligned}
&\varrho(\boldsymbol{A}) \leq \min\{m, n\} \\
&\varrho(\boldsymbol{A}) = \varrho(\boldsymbol{A}^T) \\
&\varrho(\boldsymbol{A}^T\boldsymbol{A}) = \varrho(\boldsymbol{A}) \\
&\varrho(\boldsymbol{AB}) \leq \min\{\varrho(\boldsymbol{A}), \varrho(\boldsymbol{B})\}.
\end{aligned}$$

A matrix so that $\varrho(\boldsymbol{A}) = \min\{m, n\}$ is said to be *full-rank*.

The *adjoint* of a square matrix $\boldsymbol{A}$ is the matrix

$$\operatorname{Adj} \boldsymbol{A} = [(-1)^{i+j} \det(\boldsymbol{A}_{(ij)})]^T_{\substack{i=1,\ldots,n \\ j=1,\ldots,n}}. \tag{A.11}$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *invertible* if a matrix $\boldsymbol{A}^{-1}$ exists, termed *inverse* of $\boldsymbol{A}$, so that

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}_n.$$

Since $\varrho(\boldsymbol{I}_n) = n$, an $(n \times n)$ square matrix $\boldsymbol{A}$ is invertible if and only if $\varrho(\boldsymbol{A}) = n$, i.e., $\det(\boldsymbol{A}) \neq 0$ (nonsingular matrix). The inverse of $\boldsymbol{A}$ can be computed as

$$\boldsymbol{A}^{-1} = \frac{1}{\det(\boldsymbol{A})} \operatorname{Adj} \boldsymbol{A}. \tag{A.12}$$

The following properties hold:

$$\begin{aligned}
&(\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A} \\
&(\boldsymbol{A}^T)^{-1} = (\boldsymbol{A}^{-1})^T.
\end{aligned}$$

If the inverse of a square matrix is equal to its transpose

$$\boldsymbol{A}^T = \boldsymbol{A}^{-1} \tag{A.13}$$

then the matrix is said to be *orthogonal*; in this case it is

$$\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}. \tag{A.14}$$

A square matrix $\boldsymbol{A}$ is said *idempotent* if

$$\boldsymbol{A}\boldsymbol{A} = \boldsymbol{A}. \tag{A.15}$$

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are invertible square matrices of the same dimensions, then

$$(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}. \tag{A.16}$$

Given n square matrices $\boldsymbol{A}_{ii}$ all invertible, the following expression holds:

$$\big(\mathrm{diag}\{\boldsymbol{A}_{11},\ldots,\boldsymbol{A}_{nn}\}\big)^{-1} = \mathrm{diag}\{\boldsymbol{A}_{11}^{-1},\ldots,\boldsymbol{A}_{nn}^{-1}\}.$$

where $\mathrm{diag}\{\boldsymbol{A}_{11},\ldots,\boldsymbol{A}_{nn}\}$ denotes the block-diagonal matrix.

If $\boldsymbol{A}$ and $\boldsymbol{C}$ are invertible square matrices of proper dimensions, the following expression holds:

$$(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{C}\boldsymbol{D})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B}+\boldsymbol{C}^{-1})^{-1}\boldsymbol{D}\boldsymbol{A}^{-1},$$

where the matrix $\boldsymbol{D}\boldsymbol{A}^{-1}\boldsymbol{B}+\boldsymbol{C}^{-1}$ must be invertible.

If a block-partitioned matrix is invertible, then its inverse is given by the general expression

$$\begin{bmatrix}\boldsymbol{A} & \boldsymbol{D}\\ \boldsymbol{C} & \boldsymbol{B}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{A}^{-1}+\boldsymbol{E}\boldsymbol{\Delta}^{-1}\boldsymbol{F} & -\boldsymbol{E}\boldsymbol{\Delta}^{-1}\\ -\boldsymbol{\Delta}^{-1}\boldsymbol{F} & \boldsymbol{\Delta}^{-1}\end{bmatrix} \tag{A.17}$$

where $\boldsymbol{\Delta} = \boldsymbol{B}-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{D}$, $\boldsymbol{E} = \boldsymbol{A}^{-1}\boldsymbol{D}$ and $\boldsymbol{F} = \boldsymbol{C}\boldsymbol{A}^{-1}$, under the assumption that the inverses of matrices $\boldsymbol{A}$ and $\boldsymbol{\Delta}$ exist. In the case of a block-triangular matrix, invertibility of the matrix requires invertibility of the blocks on the diagonal. The following expressions hold:

$$\begin{bmatrix}\boldsymbol{A} & \boldsymbol{O}\\ \boldsymbol{C} & \boldsymbol{B}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{A}^{-1} & \boldsymbol{O}\\ -\boldsymbol{B}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{B}^{-1}\end{bmatrix}$$

$$\begin{bmatrix}\boldsymbol{A} & \boldsymbol{D}\\ \boldsymbol{O} & \boldsymbol{B}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{D}\boldsymbol{B}^{-1}\\ \boldsymbol{O} & \boldsymbol{B}^{-1}\end{bmatrix}.$$

The *derivative* of an $(m\times n)$ matrix $\boldsymbol{A}(t)$, whose elements $a_{ij}(t)$ are differentiable functions, is the matrix

$$\dot{\boldsymbol{A}}(t) = \frac{d}{dt}\boldsymbol{A}(t) = \left[\frac{d}{dt}a_{ij}(t)\right]_{\substack{i=1,\ldots,m\\ j=1,\ldots,n}}. \tag{A.18}$$

If an $(n\times n)$ square matrix $\boldsymbol{A}(t)$ is so that $\varrho(\boldsymbol{A}(t)) = n\ \forall t$ and its elements $a_{ij}(t)$ are differentiable functions, then the derivative of the *inverse* of $\boldsymbol{A}(t)$ is given by

$$\frac{d}{dt}\boldsymbol{A}^{-1}(t) = -\boldsymbol{A}^{-1}(t)\dot{\boldsymbol{A}}(t)\boldsymbol{A}^{-1}(t). \tag{A.19}$$

Given a scalar function $f(\boldsymbol{x})$, endowed with partial derivatives with respect to the elements x_i of the $(n\times 1)$ vector $\boldsymbol{x}$, the *gradient* of function f with respect to vector $\boldsymbol{x}$ is the $(n\times 1)$ column vector

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T = \left[\begin{array}{cccc}\dfrac{\partial f(\boldsymbol{x})}{\partial x_1} & \dfrac{\partial f(\boldsymbol{x})}{\partial x_2} & \ldots & \dfrac{\partial f(\boldsymbol{x})}{\partial x_n}\end{array}\right]^T. \tag{A.20}$$

Further, if $\boldsymbol{x}(t)$ is a differentiable function with respect to t, then

$$\dot{f}(\boldsymbol{x}) = \frac{d}{dt}f(\boldsymbol{x}(t)) = \frac{\partial f}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \nabla_{\boldsymbol{x}}^T f(\boldsymbol{x})\dot{\boldsymbol{x}}. \tag{A.21}$$

Given a vector function $\boldsymbol{g}(\boldsymbol{x})$ of dimensions $(m\times 1)$, whose elements g_i are differentiable with respect to the vector $\boldsymbol{x}$ of dimensions $(n\times 1)$, the Jacobian matrix (or simply *Jacobian*) of the function is defined as the $(m\times n)$ matrix

$$\boldsymbol{J}_g(\boldsymbol{x}) = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix}\dfrac{\partial g_1(\boldsymbol{x})}{\partial \boldsymbol{x}}\\ \dfrac{\partial g_2(\boldsymbol{x})}{\partial \boldsymbol{x}}\\ \vdots\\ \dfrac{\partial g_m(\boldsymbol{x})}{\partial \boldsymbol{x}}\end{bmatrix}. \tag{A.22}$$

If $\boldsymbol{x}(t)$ is a differentiable function with respect to t, then

$$\dot{\boldsymbol{g}}(\boldsymbol{x}) = \frac{d}{dt}\boldsymbol{g}(\boldsymbol{x}(t)) = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}} = \boldsymbol{J}_g(\boldsymbol{x})\dot{\boldsymbol{x}}. \tag{A.23}$$

A.3 Vector Operations

Given n vectors $\boldsymbol{x}_i$ of dimensions $(m\times 1)$, they are said to be *linearly independent* if the expression

$$k_1\boldsymbol{x}_1 + k_2\boldsymbol{x}_2 + \ldots + k_n\boldsymbol{x}_n = \boldsymbol{0}$$

holds true only when all the constants k_i vanish. A necessary and sufficient condition for the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_n$ to be linearly independent is that the matrix

$$\boldsymbol{A} = [\,\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \ldots \quad \boldsymbol{x}_n\,]$$

has rank n; this implies that a necessary condition for linear independence is that $n \leq m$. If instead $\varrho(\boldsymbol{A}) = r < n$, then only r vectors are linearly independent and the remaining $n - r$ vectors can be expressed as a linear combination of the previous ones.

A system of vectors $\mathcal{X}$ is a *vector space* on the field of real numbers $\mathbb{R}$ if the operations of *sum of two vectors* of $\mathcal{X}$ and *product of a scalar by a vector* of $\mathcal{X}$ have values in $\mathcal{X}$ and the following properties hold:

$$\begin{aligned}
&\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x} \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \\
&(\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} = \boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) \qquad \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{X} \\
&\exists \boldsymbol{0} \in \mathcal{X} : \boldsymbol{x} + \boldsymbol{0} = \boldsymbol{x} \qquad \forall \boldsymbol{x} \in \mathcal{X} \\
&\forall \boldsymbol{x} \in \mathcal{X}, \quad \exists (-\boldsymbol{x}) \in \mathcal{X} : \boldsymbol{x} + (-\boldsymbol{x}) = \boldsymbol{0} \\
&1\boldsymbol{x} = \boldsymbol{x} \qquad \forall \boldsymbol{x} \in \mathcal{X} \\
&\alpha(\beta \boldsymbol{x}) = (\alpha\beta)\boldsymbol{x} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x} \in \mathcal{X} \\
&(\alpha + \beta)\boldsymbol{x} = \alpha\boldsymbol{x} + \beta\boldsymbol{x} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x} \in \mathcal{X} \\
&\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha\boldsymbol{x} + \alpha\boldsymbol{y} \qquad \forall \alpha \in \mathbb{R} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}.
\end{aligned}$$

The *dimension* of the space $\dim(\mathcal{X})$ is the maximum number of linearly independent vectors $\boldsymbol{x}$ in the space. A set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ of linearly independent vectors is a *basis* of vector space $\mathcal{X}$, and each vector $\boldsymbol{y}$ in the space can be uniquely expressed as a linear combination of vectors from the basis

$$\boldsymbol{y} = c_1\boldsymbol{x}_1 + c_2\boldsymbol{x}_2 + \ldots + c_n\boldsymbol{x}_n, \tag{A.24}$$

where the constants $c_1, c_2, \ldots, c_n$ are said to be the *components* of the vector $\boldsymbol{y}$ in the basis $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$.

A subset $\mathcal{Y}$ of a vector space $\mathcal{X}$ is a *subspace* $\mathcal{Y} \subseteq \mathcal{X}$ if it is a vector space with the operations of vector sum and product of a scalar by a vector, i.e.,

$$\alpha\boldsymbol{x} + \beta\boldsymbol{y} \in \mathcal{Y} \qquad \forall \alpha, \beta \in \mathbb{R} \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Y}.$$

According to a geometric interpretation, a subspace is a *hyperplane* passing by the origin (null element) of $\mathcal{X}$.

The *scalar product* $< \boldsymbol{x}, \boldsymbol{y} >$ of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of dimensions $(m \times 1)$ is the scalar that is obtained by summing the products of the respective components in a given basis

$$< \boldsymbol{x}, \boldsymbol{y} >= x_1y_1 + x_2y_2 + \ldots + x_my_m = \boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{x}. \tag{A.25}$$

Two vectors are said to be *orthogonal* when their scalar product is null:

$$\boldsymbol{x}^T\boldsymbol{y} = 0. \tag{A.26}$$

The *norm* of a vector can be defined as

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}. \tag{A.27}$$

It is possible to show that both the *triangle inequality*

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\| \tag{A.28}$$

and the *Schwarz inequality*

$$|\boldsymbol{x}^T\boldsymbol{y}| \leq \|\boldsymbol{x}\| \, \|\boldsymbol{y}\|. \tag{A.29}$$

hold. A *unit vector* $\hat{\boldsymbol{x}}$ is a vector whose *norm* is unity, i.e., $\hat{\boldsymbol{x}}^T\hat{\boldsymbol{x}} = 1$. Given a vector $\boldsymbol{x}$, its unit vector is obtained by dividing each component by its norm:

$$\widehat{\boldsymbol{x}} = \frac{1}{\|\boldsymbol{x}\|}\boldsymbol{x}. \tag{A.30}$$

A typical example of vector space is the *Euclidean space* whose dimension is 3; in this case a basis is constituted by the unit vectors of a coordinate frame.

The *vector product* of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in the Euclidean space is the vector

$$\boldsymbol{x} \times \boldsymbol{y} = \begin{bmatrix} x_2y_3 - x_3y_2 \\ x_3y_1 - x_1y_3 \\ x_1y_2 - x_2y_1 \end{bmatrix}. \tag{A.31}$$

The following properties hold:

$$\begin{aligned}
&\boldsymbol{x} \times \boldsymbol{x} = \boldsymbol{0} \\
&\boldsymbol{x} \times \boldsymbol{y} = -\boldsymbol{y} \times \boldsymbol{x} \\
&\boldsymbol{x} \times (\boldsymbol{y} + \boldsymbol{z}) = \boldsymbol{x} \times \boldsymbol{y} + \boldsymbol{x} \times \boldsymbol{z}.
\end{aligned}$$

The vector product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ can be expressed also as the product of a matrix operator $\boldsymbol{S}(\boldsymbol{x})$ by the vector $\boldsymbol{y}$. In fact, by introducing the *skew-symmetric* matrix

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{A.32}$$

obtained with the components of vector $\boldsymbol{x}$, the vector product $\boldsymbol{x} \times \boldsymbol{y}$ is given by

$$\boldsymbol{x} \times \boldsymbol{y} = \boldsymbol{S}(\boldsymbol{x})\boldsymbol{y} = -\boldsymbol{S}(\boldsymbol{y})\boldsymbol{x} \tag{A.33}$$

as can be easily verified. Moreover, the following properties hold:

$$\begin{aligned}
&\boldsymbol{S}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{S}^T(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{0} \\
&\boldsymbol{S}(\alpha\boldsymbol{x} + \beta\boldsymbol{y}) = \alpha\boldsymbol{S}(\boldsymbol{x}) + \beta\boldsymbol{S}(\boldsymbol{y}).
\end{aligned}$$

Given three vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ in the Euclidean space, the following expressions hold for the *scalar triple products*:

$$\boldsymbol{x}^T(\boldsymbol{y} \times \boldsymbol{z}) = \boldsymbol{y}^T(\boldsymbol{z} \times \boldsymbol{x}) = \boldsymbol{z}^T(\boldsymbol{x} \times \boldsymbol{y}). \tag{A.34}$$

If any two vectors of three are equal, then the scalar triple product is null; e.g.,

$$\boldsymbol{x}^T(\boldsymbol{x} \times \boldsymbol{y}) = \boldsymbol{0}.$$

A.4 Linear Transformation

Consider a vector space $\mathcal{X}$ of dimension n and a vector space $\mathcal{Y}$ of dimension m with $m \leq n$. The *linear transformation* (or linear map) between the vectors $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ can be defined as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{A.35}$$

in terms of the matrix $\boldsymbol{A}$ of dimensions $(m \times n)$. The *range space* (or simply range) of the transformation is the subspace

$$\mathcal{R}(\boldsymbol{A}) = \{\boldsymbol{y} : \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},\ \boldsymbol{x} \in \mathcal{X}\} \subseteq \mathcal{Y}, \tag{A.36}$$

which is the subspace generated by the linearly independent columns of matrix $\boldsymbol{A}$ taken as a basis of $\mathcal{Y}$. It is easy to recognize that

$$\varrho(\boldsymbol{A}) = \dim(\mathcal{R}(\boldsymbol{A})). \tag{A.37}$$

On the other hand, the *null space* (or simply null) of the transformation is the subspace

$$\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0},\ \boldsymbol{x} \in \mathcal{X}\} \subseteq \mathcal{X}. \tag{A.38}$$

Given a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$, the notable result holds:

$$\varrho(\boldsymbol{A}) + \dim(\mathcal{N}(\boldsymbol{A})) = n. \tag{A.39}$$

Therefore, if $\varrho(\boldsymbol{A}) = r \leq \min\{m, n\}$, then $\dim(\mathcal{R}(\boldsymbol{A})) = r$ and $\dim(\mathcal{N}(\boldsymbol{A})) = n - r$. It follows that if $m < n$, then $\mathcal{N}(\boldsymbol{A}) \neq \emptyset$ independently of the rank of $\boldsymbol{A}$; if $m = n$, then $\mathcal{N}(\boldsymbol{A}) \neq \emptyset$ only in the case of $\varrho(\boldsymbol{A}) = r < m$.

If $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$ and $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{A}^T)$, then $\boldsymbol{y}^T\boldsymbol{x} = 0$, i.e., the vectors in the null space of $\boldsymbol{A}$ are orthogonal to each vector in the range space of the transpose of $\boldsymbol{A}$. It can be shown that the set of vectors orthogonal to each vector of the range space of $\boldsymbol{A}^T$ coincides with the null space of $\boldsymbol{A}$, whereas the set of vectors orthogonal to each vector in the null space of $\boldsymbol{A}^T$ coincides with the range space of $\boldsymbol{A}$. In symbols:

$$\mathcal{N}(\boldsymbol{A}) \equiv \mathcal{R}^{\perp}(\boldsymbol{A}^T) \qquad \mathcal{R}(\boldsymbol{A}) \equiv \mathcal{N}^{\perp}(\boldsymbol{A}^T) \tag{A.40}$$

where $\perp$ denotes the *orthogonal complement* of a subspace.

If the matrix $\boldsymbol{A}$ in (A.35) is square and idempotent, the matrix represents the *projection* of space $\mathcal{X}$ into a subspace.

A linear transformation allows the definition of the *norm* of a matrix $\boldsymbol{A}$ induced by the norm defined for a vector $\boldsymbol{x}$ as follows. In view of the property

$$\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|\ \|\boldsymbol{x}\|, \tag{A.41}$$

the norm of $\boldsymbol{A}$ can be defined as

$$\|\boldsymbol{A}\| = \sup_{\boldsymbol{x} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \tag{A.42}$$

which can also be computed as

$$\max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|.$$

A direct consequence of (A.41) is the property

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\ \|\boldsymbol{B}\|. \tag{A.43}$$

A different norm of a matrix is the *Frobenius norm* defined as

$$\|\boldsymbol{A}\|_F = \Big(\mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})\Big)^{1/2} \tag{A.44}$$

A.5 Eigenvalues and Eigenvectors

Consider the linear transformation on a vector $\boldsymbol{u}$ established by an $(n \times n)$ square matrix $\boldsymbol{A}$. If the vector resulting from the transformation has the same direction of $\boldsymbol{u}$ (with $\boldsymbol{u} \neq \boldsymbol{0}$), then

$$\boldsymbol{A}\boldsymbol{u} = \lambda\boldsymbol{u}. \tag{A.45}$$

The equation in (A.45) can be rewritten in matrix form as

$$(\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u} = \boldsymbol{0}. \tag{A.46}$$

For the homogeneous system of equations in (A.46) to have a solution different from the trivial one $\boldsymbol{u} = \boldsymbol{0}$, it must be

$$\det(\lambda\boldsymbol{I} - \boldsymbol{A}) = 0 \tag{A.47}$$

which is termed a *characteristic equation*. Its solutions $\lambda_1, \ldots, \lambda_n$ are the *eigenvalues* of matrix $\boldsymbol{A}$; they coincide with the eigenvalues of matrix $\boldsymbol{A}^T$. On the assumption of distinct eigenvalues, the n vectors $\boldsymbol{u}_i$ satisfying the equation

$$(\lambda_i\boldsymbol{I} - \boldsymbol{A})\boldsymbol{u}_i = \boldsymbol{0} \qquad i = 1, \ldots, n \tag{A.48}$$

are said to be the *eigenvectors* associated with the eigenvalues λ_i.

The matrix $\boldsymbol{U}$ formed by the column vectors $\boldsymbol{u}_i$ is invertible and constitutes a basis in the space of dimension n. Further, the *similarity transformation* established by $\boldsymbol{U}$

$$\boldsymbol{\Lambda} = \boldsymbol{U}^{-1}\boldsymbol{A}\boldsymbol{U} \tag{A.49}$$

is so that $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$. It follows that $\det(\boldsymbol{A}) = \prod_{i=1}^{n} \lambda_i$.

If the matrix $\boldsymbol{A}$ is *symmetric*, its eigenvalues are real and $\boldsymbol{\Lambda}$ can be written as

$$\boldsymbol{\Lambda} = \boldsymbol{U}^T\boldsymbol{A}\boldsymbol{U}; \tag{A.50}$$

hence, the eigenvector matrix $\boldsymbol{U}$ is orthogonal.

A.6 Bilinear Forms and Quadratic Forms

A *bilinear form* in the variables x_i and y_j is the scalar

$$B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j$$

which can be written in matrix form

$$B(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y} = \boldsymbol{y}^T \boldsymbol{A}^T \boldsymbol{x} \tag{A.51}$$

where $\boldsymbol{x} = [\, x_1 \quad x_2 \quad \dots \quad x_m \,]^T$, $\boldsymbol{y} = [\, y_1 \quad y_2 \quad \dots \quad y_n \,]^T$, and $\boldsymbol{A}$ is the $(m \times n)$ matrix of the coefficients a_{ij} representing the core of the form.

A special case of bilinear form is the *quadratic form*

$$Q(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \tag{A.52}$$

where $\boldsymbol{A}$ is an $(n \times n)$ square matrix. Hence, for computation of (A.52), the matrix $\boldsymbol{A}$ can be replaced with its symmetric part $\boldsymbol{A}_s$ given by (A.6). It follows that if $\boldsymbol{A}$ is a *skew-symmetric* matrix, then

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \boldsymbol{0} \qquad \forall \boldsymbol{x}.$$

The quadratic form (A.52) is said to be *positive definite* if

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0 \quad \forall \boldsymbol{x} \neq \boldsymbol{0} \qquad \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = 0 \quad \boldsymbol{x} = \boldsymbol{0}. \tag{A.53}$$

The matrix $\boldsymbol{A}$ core of the form is also said to be *positive definite*. Analogously, a quadratic form is said to be *negative definite* if it can be written as $-Q(\boldsymbol{x}) = -\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ where $Q(\boldsymbol{x})$ is positive definite.

A necessary condition for a square matrix to be positive definite is that its elements on the diagonal are strictly positive. Further, in view of (A.50), the eigenvalues of a positive definite matrix are all positive. If the eigenvalues are not known, a necessary and sufficient condition for a symmetric matrix to be positive definite is that its principal minors are strictly positive (*Sylvester criterion*). It follows that a positive definite matrix is full-rank and thus it is always invertible.

A symmetric positive definite matrix $\boldsymbol{A}$ can always be decomposed as

$$\boldsymbol{A} = \boldsymbol{U}^T \boldsymbol{\Lambda} \boldsymbol{U} \tag{A.54}$$

where $\boldsymbol{U}$ is an orthogonal matrix of eigenvectors ($\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}$) and $\boldsymbol{\Lambda}$ is the diagonal matrix of the eigenvalues of $\boldsymbol{A}$.

Let $\lambda_{\min}(\boldsymbol{A})$ and $\lambda_{\max}(\boldsymbol{A})$ respectively denote the smallest and largest eigenvalues of a positive definite matrix $\boldsymbol{A}$ ($\lambda_{\min}, \lambda_{\max} > 0$). Then, the quadratic form in (A.52) satisfies the following inequality:

$$\lambda_{\min}(\boldsymbol{A}) \|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \leq \lambda_{\max}(\boldsymbol{A}) \|\boldsymbol{x}\|^2. \tag{A.55}$$

An $(n \times n)$ square matrix $\boldsymbol{A}$ is said to be *positive semi-definite* if

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0 \qquad \forall \boldsymbol{x}. \tag{A.56}$$

This definition implies that $\varrho(\boldsymbol{A}) = r < n$, and thus r eigenvalues of $\boldsymbol{A}$ are positive and $n - r$ are null. Therefore, a positive semi-definite matrix $\boldsymbol{A}$ has a null space of finite dimension, and specifically the form vanishes when $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{A})$. A typical example of a positive semi-definite matrix is the matrix $\boldsymbol{A} = \boldsymbol{H}^T \boldsymbol{H}$ where $\boldsymbol{H}$ is an $(m \times n)$ matrix with $m < n$. In an analogous way, a *negative semi-definite* matrix can be defined.

Given the *bilinear form* in (A.51), the *gradient* of the form with respect to $\boldsymbol{x}$ is given by

$$\nabla_{\boldsymbol{x}} B(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\partial B(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \right)^T = \boldsymbol{A} \boldsymbol{y}, \tag{A.57}$$

whereas the gradient of B with respect to $\boldsymbol{y}$ is given by

$$\nabla_{\boldsymbol{y}} B(\boldsymbol{x}, \boldsymbol{y}) = \left(\frac{\partial B(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{y}} \right)^T = \boldsymbol{A}^T \boldsymbol{x}. \tag{A.58}$$

Given the *quadratic form* in (A.52) with $\boldsymbol{A}$ *symmetric*, the *gradient* of the form with respect to $\boldsymbol{x}$ is given by

$$\nabla_{\boldsymbol{x}} Q(\boldsymbol{x}) = \left(\frac{\partial Q(\boldsymbol{x})}{\partial \boldsymbol{x}} \right)^T = 2 \boldsymbol{A} \boldsymbol{x}. \tag{A.59}$$

Further, if $\boldsymbol{x}$ and $\boldsymbol{A}$ are differentiable functions of t, then

$$\dot{Q}(x) = \frac{d}{dt} Q(\boldsymbol{x}(t)) = 2 \boldsymbol{x}^T \boldsymbol{A} \dot{\boldsymbol{x}} + \boldsymbol{x}^T \dot{\boldsymbol{A}} \boldsymbol{x}; \tag{A.60}$$

if $\boldsymbol{A}$ is constant, then the second term obviously vanishes.

A.7 Pseudo-inverse

The inverse of a matrix can be defined only when the matrix is square and nonsingular. The inverse operation can be extended to the case of non-square matrices. Consider a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$ with $\varrho(\boldsymbol{A}) = \min\{m, n\}$

If $m < n$, a *right inverse* of $\boldsymbol{A}$ can be defined as the matrix $\boldsymbol{A}_r$ of dimensions $(n \times m)$ so that

$$\boldsymbol{A} \boldsymbol{A}_r = \boldsymbol{I}_m.$$

If instead $m > n$, a *left inverse* of $\boldsymbol{A}$ can be defined as the matrix $\boldsymbol{A}_l$ of dimensions $(n \times m)$ so that

$$\boldsymbol{A}_l \boldsymbol{A} = \boldsymbol{I}_n.$$

If $\boldsymbol{A}$ has more columns than rows ($m < n$) and has rank m, a special right inverse is the matrix

$$\boldsymbol{A}_r^\dagger = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1} \tag{A.61}$$

which is termed *right pseudo-inverse*, since $\boldsymbol{A}\boldsymbol{A}_r^\dagger = \boldsymbol{I}_m$. If $\boldsymbol{W}_r$ is an $(n \times n)$ *positive definite* matrix, a *weighted* right pseudo-inverse is given by

$$\boldsymbol{A}_r^\dagger = \boldsymbol{W}_r^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{W}_r^{-1}\boldsymbol{A}^T)^{-1}. \tag{A.62}$$

If $\boldsymbol{A}$ has more rows than columns ($m > n$) and has rank n, a special left inverse is the matrix

$$\boldsymbol{A}_l^\dagger = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T \tag{A.63}$$

which is termed *left pseudo-inverse*, since $\boldsymbol{A}_l^\dagger\boldsymbol{A} = \boldsymbol{I}_n$.[3] If $\boldsymbol{W}_l$ is an $(m \times m)$ *positive definite* matrix, a *weighted* left pseudo-inverse is given by

$$\boldsymbol{A}_l^\dagger = (\boldsymbol{A}^T\boldsymbol{W}_l\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{W}_l. \tag{A.64}$$

The pseudo-inverse is very useful to invert a linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with $\boldsymbol{A}$ a full-rank matrix. If $\boldsymbol{A}$ is a square nonsingular matrix, then obviously $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$ and then $\boldsymbol{A}_l^\dagger = \boldsymbol{A}_r^\dagger = \boldsymbol{A}^{-1}$.

If $\boldsymbol{A}$ has more columns than rows ($m < n$) and has rank m, then the solution $\boldsymbol{x}$ for a given $\boldsymbol{y}$ is not unique; it can be shown that the expression

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{y} + (\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{k}, \tag{A.65}$$

with $\boldsymbol{k}$ an arbitrary $(n \times 1)$ vector and $\boldsymbol{A}^\dagger$ as in (A.61), is a solution to the system of linear equations established by (A.35). The term $\boldsymbol{A}^\dagger\boldsymbol{y} \in \mathcal{N}^\perp(\boldsymbol{A}) \equiv \mathcal{R}(\boldsymbol{A}^T)$ minimizes the norm of the solution $\|\boldsymbol{x}\|$. The term $(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{k}$ is the projection of $\boldsymbol{k}$ in $\mathcal{N}(\boldsymbol{A})$ and is termed *homogeneous solution*; as $\boldsymbol{k}$ varies, all the solutions to the homogeneous equation system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ associated with (A.35) are generated.

On the other hand, if $\boldsymbol{A}$ has more rows than columns ($m > n$), the equation in (A.35) has no solution; it can be shown that an *approximate* solution is given by

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{y} \tag{A.66}$$

where $\boldsymbol{A}^\dagger$ as in (A.63) minimizes $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|$. If instead $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{A})$, then (A.66) is a real solution.

Notice that the use of the weighted (left or right) pseudo-inverses in the solution to the linear equation systems leads to analogous results where the minimized norms are weighted according to the metrics defined by matrices $\boldsymbol{W}_r$ and $\boldsymbol{W}_l$, respectively.

The results of this section can be easily extended to the case of (square or nonsquare) matrices $\boldsymbol{A}$ not having full-rank. In particular, the expression (A.66) (with the pseudo-inverse computed by means of the singular value decomposition of $\boldsymbol{A}$) gives the minimum-norm vector among all those minimizing $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|$.

[3] Subscripts l and r are usually omitted whenever the use of a left or right pseudo-inverse is clear from the context.

A.8 Singular Value Decomposition

For a nonsquare matrix it is not possible to define eigenvalues. An extension of the eigenvalue concept can be obtained by singular values. Given a matrix $\boldsymbol{A}$ of dimensions $(m \times n)$, the matrix $\boldsymbol{A}^T\boldsymbol{A}$ has n nonnegative eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ (ordered from the largest to the smallest) which can be expressed in the form

$$\lambda_i = \sigma_i^2 \qquad \sigma_i \geq 0.$$

The scalars $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are said to be the *singular values* of matrix $\boldsymbol{A}$. The *singular value decomposition* (SVD) of matrix $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{A.67}$$

where $\boldsymbol{U}$ is an $(m \times m)$ orthogonal matrix

$$\boldsymbol{U} = [\,\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \ldots \quad \boldsymbol{u}_m\,], \tag{A.68}$$

$\boldsymbol{V}$ is an $(n \times n)$ orthogonal matrix

$$\boldsymbol{V} = [\,\boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \ldots \quad \boldsymbol{v}_n\,] \tag{A.69}$$

and $\boldsymbol{\Sigma}$ is an $(m \times n)$ matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{D} = \mathrm{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\} \tag{A.70}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. The number of non-null singular values is equal to the rank r of matrix $\boldsymbol{A}$.

The columns of $\boldsymbol{U}$ are the eigenvectors of the matrix $\boldsymbol{A}\boldsymbol{A}^T$, whereas the columns of $\boldsymbol{V}$ are the eigenvectors of the matrix $\boldsymbol{A}^T\boldsymbol{A}$. In view of the partitions of $\boldsymbol{U}$ and $\boldsymbol{V}$ in (A.68), (A.69), it is $\boldsymbol{A}\boldsymbol{v}_i = \sigma_i\boldsymbol{u}_i$, for $i = 1, \ldots, r$ and $\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{0}$, for $i = r+1, \ldots, n$.

Singular value decomposition is useful for analysis of the linear transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ established in (A.35). According to a geometric interpretation, the matrix $\boldsymbol{A}$ transforms the unit sphere in $\mathbb{R}^n$ defined by $\|\boldsymbol{x}\| = 1$ into the set of vectors $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ which define an *ellipsoid* of dimension r in $\mathbb{R}^m$. The singular values are the lengths of the various axes of the ellipsoid. The *condition number* of the matrix

$$\kappa = \frac{\sigma_1}{\sigma_r}$$

is related to the eccentricity of the ellipsoid and provides a measure of ill-conditioning ($\kappa \gg 1$) for numerical solution of the system established by (A.35).

It is worth noticing that the numerical procedure of singular value decomposition is commonly adopted to compute the (right or left) pseudo-inverse $\boldsymbol{A}^\dagger$, even in the case of a matrix $\boldsymbol{A}$ not having full rank. In fact, from (A.67), (A.70) it is

$$\boldsymbol{A}^\dagger = \boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T \tag{A.71}$$

with

$$\boldsymbol{\Sigma}^{\dagger} = \begin{bmatrix} \boldsymbol{D}^{\dagger} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{D}^{\dagger} = \operatorname{diag}\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}\right\}. \tag{A.72}$$

Bibliography

A reference text on linear algebra is [169]. For matrix computation see [88]. The properties of pseudo-inverse matrices are discussed in [24].

B

Rigid-body Mechanics

The goal of this appendix is to recall some fundamental concepts of *rigid body mechanics* which are preliminary to the study of manipulator *kinematics*, *statics* and *dynamics*.

B.1 Kinematics

A *rigid body* is a system characterized by the constraint that the distance between any two points is always constant.

Consider a rigid body $\mathcal{B}$ moving with respect to an orthonormal reference frame O–xyz of unit vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, called *fixed frame*. The rigidity assumption allows the introduction of an orthonormal frame O'–$x'y'z'$ attached to the body, called *moving frame*, with respect to which the position of any point of $\mathcal{B}$ is independent of time. Let $\boldsymbol{x}'(t)$, $\boldsymbol{y}'(t)$, $\boldsymbol{z}'(t)$ be the unit vectors of the moving frame expressed in the fixed frame at time t.

The orientation of the moving frame O'–$x'y'z'$ at time t with respect to the fixed frame O–xyz can be expressed by means of the *orthogonal* (3×3) matrix

$$\boldsymbol{R}(t) = \begin{bmatrix} \boldsymbol{x}'^{T}(t)\boldsymbol{x} & \boldsymbol{y}'^{T}(t)\boldsymbol{x} & \boldsymbol{z}'^{T}(t)\boldsymbol{x} \\ \boldsymbol{x}'^{T}(t)\boldsymbol{y} & \boldsymbol{y}'^{T}(t)\boldsymbol{y} & \boldsymbol{z}'^{T}(t)\boldsymbol{y} \\ \boldsymbol{x}'^{T}(t)\boldsymbol{z} & \boldsymbol{y}'^{T}(t)\boldsymbol{z} & \boldsymbol{z}'^{T}(t)\boldsymbol{z} \end{bmatrix}, \tag{B.1}$$

which is termed *rotation matrix* defined in the orthonormal special group $SO(3)$ of the (3×3) matrices with orthonormal columns and determinant equal to 1. The columns of the matrix in (B.1) represent the components of the unit vectors of the moving frame when expressed in the fixed frame, whereas the rows represent the components of the unit vectors of the fixed frame when expressed in the moving frame.

Let $\boldsymbol{p}'$ be the *constant* position vector of a generic point P of $\mathcal{B}$ in the moving frame O'–$x'y'z'$. The motion of P with respect to the fixed frame O–xyz is described by the equation

$$\boldsymbol{p}(t) = \boldsymbol{p}_{O'}(t) + \boldsymbol{R}(t)\boldsymbol{p}', \tag{B.2}$$

where $\boldsymbol{p}_{O'}(t)$ is the position vector of origin O' of the moving frame with respect to the fixed frame.

Notice that a position vector is a *bound vector* since its line of application and point of application are both prescribed, in addition to its direction; the point of application typically coincides with the origin of a reference frame. Therefore, to transform a bound vector from a frame to another, both translation and rotation between the two frames must be taken into account.

If the positions of the points of $\mathcal{B}$ in the moving frame are known, it follows from (B.2) that the motion of each point of $\mathcal{B}$ with respect to the fixed frame is uniquely determined once the position of the origin and the orientation of the moving frame with respect to the fixed frame are specified in time. The origin of the moving frame is determined by *three* scalar functions of time. Since the orthonormality conditions impose six constraints on the nine elements of matrix $\boldsymbol{R}(t)$, the *orientation* of the moving frame depends only on *three* independent scalar functions, three being the minimum number of parameters to represent $SO(3)$.[1]

Therefore, a rigid body motion is described by arbitrarily specifying *six* scalar functions of time, which describe the body *pose* (position + orientation). The resulting rigid motions belong to the *special Euclidean group* $SE(3) = \mathbb{R}^3 \times SO(3)$.

The expression in (B.2) continues to hold if the position vector $\boldsymbol{p}_{O'}(t)$ of the origin of the moving frame is replaced with the position vector of any other point of $\mathcal{B}$, i.e.,

$$\boldsymbol{p}(t) = \boldsymbol{p}_Q(t) + \boldsymbol{R}(t)(\boldsymbol{p}' - \boldsymbol{p}'_Q) \tag{B.3}$$

where $\boldsymbol{p}_Q(t)$ and $\boldsymbol{p}'_Q$ are the position vectors of a point Q of $\mathcal{B}$ in the fixed and moving frames, respectively.

In the following, for simplicity of notation, the dependence on the time variable t will be dropped.

Differentiating (B.3) with respect to time gives the known velocity composition rule

$$\dot{\boldsymbol{p}} = \dot{\boldsymbol{p}}_Q + \boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q), \tag{B.4}$$

where $\boldsymbol{\omega}$ is the *angular velocity* of rigid body $\mathcal{B}$. Notice that $\boldsymbol{\omega}$ is a *free vector* since its point of application is not prescribed. To transform a free vector from a frame to another, only rotation between the two frames must be taken into account.

By recalling the definition of the skew-symmetric operator $\boldsymbol{S}(\cdot)$ in (A.32), the expression in (B.4) can be rewritten as

$$\begin{aligned}\dot{\boldsymbol{p}} &= \dot{\boldsymbol{p}}_Q + \boldsymbol{S}(\boldsymbol{\omega})(\boldsymbol{p} - \boldsymbol{p}_Q) \\ &= \dot{\boldsymbol{p}}_Q + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}(\boldsymbol{p}' - \boldsymbol{p}'_Q).\end{aligned}$$

[1] The minimum number of parameters represent a special orthonormal group $SO(m)$ is equal to $m(m-1)/2$.

Comparing this equation with the formal time derivative of (B.3) leads to the result

$$\dot{\boldsymbol{R}} = \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}. \tag{B.5}$$

In view of (B.4), the *elementary displacement* of a point P of the rigid body $\mathcal{B}$ in the time interval $(t, t+dt)$ is

$$\begin{aligned}d\boldsymbol{p} = \dot{\boldsymbol{p}}dt &= \big(\dot{\boldsymbol{p}}_Q + \boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q)\big)dt \\ &= d\boldsymbol{p}_Q + \boldsymbol{\omega}dt \times (\boldsymbol{p} - \boldsymbol{p}_Q).\end{aligned} \tag{B.6}$$

Differentiating (B.4) with respect to time yields the following expression for acceleration:

$$\ddot{\boldsymbol{p}} = \ddot{\boldsymbol{p}}_Q + \dot{\boldsymbol{\omega}} \times (\boldsymbol{p} - \boldsymbol{p}_Q) + \boldsymbol{\omega} \times \big(\boldsymbol{\omega} \times (\boldsymbol{p} - \boldsymbol{p}_Q)\big). \tag{B.7}$$

B.2 Dynamics

Let ρdV be the mass of an elementary particle of a rigid body $\mathcal{B}$, where ρ denotes the density of the particle of volume dV. Also let $V_{\mathcal{B}}$ be the body volume and $m = \int_{V_{\mathcal{B}}} \rho dV$ its *total mass* assumed to be constant. If $\boldsymbol{p}$ denotes the position vector of the particle of mass ρdV in the frame O–xyz, the *centre of mass* of $\mathcal{B}$ is defined as the point C whose position vector is

$$\boldsymbol{p}_C = \frac{1}{m} \int_{V_{\mathcal{B}}} \boldsymbol{p} \rho dV. \tag{B.8}$$

In the case when $\mathcal{B}$ is the union of n distinct parts of mass $m_1, \ldots, m_n$ and centres of mass $\boldsymbol{p}_{C1} \ldots \boldsymbol{p}_{Cn}$, the centre of mass of $\mathcal{B}$ can be computed as

$$\boldsymbol{p}_C = \frac{1}{m} \sum_{i=1}^{n} m_i \boldsymbol{p}_{Ci}$$

with $m = \sum_{i=1}^{n} m_i$.

Let r be a line passing by O and $d(\boldsymbol{p})$ the distance from r of the particle of $\mathcal{B}$ of mass ρdV and position vector $\boldsymbol{p}$. The *moment of inertia* of body $\mathcal{B}$ with respect to line r is defined as the positive scalar

$$I_r = \int_{V_{\mathcal{B}}} d^2(\boldsymbol{p}) \rho dV.$$

Let $\boldsymbol{r}$ denote the unit vector of line r; then, the moment of inertia of $\mathcal{B}$ with respect to line r can be expressed as

$$I_r = \boldsymbol{r}^T \left(\int_{V_{\mathcal{B}}} \boldsymbol{S}^T(\boldsymbol{p}) \boldsymbol{S}(\boldsymbol{p}) \rho dV \right) \boldsymbol{r} = \boldsymbol{r}^T \boldsymbol{I}_O \boldsymbol{r}, \tag{B.9}$$

where $\boldsymbol{S}(\cdot)$ is the skew-symmetric operator in (A.31), and the *symmetric*, *positive definite* matrix

$$\boldsymbol{I}_O = \begin{bmatrix} \int_{V_\mathcal{B}}(p_y^2+p_z^2)\rho dV & -\int_{V_\mathcal{B}} p_x p_y \rho dV & -\int_{V_\mathcal{B}} p_x p_z \rho dV \\ * & \int_{V_\mathcal{B}}(p_x^2+p_z^2)\rho dV & -\int_{V_\mathcal{B}} p_y p_z \rho dV \\ * & * & \int_{V_\mathcal{B}}(p_x^2+p_y^2)\rho dV \end{bmatrix}$$
$$= \begin{bmatrix} I_{Oxx} & -I_{Oxy} & -I_{Oxz} \\ * & I_{Oyy} & -I_{Oyz} \\ * & * & I_{Ozz} \end{bmatrix} \tag{B.10}$$

is termed *inertia tensor* of body $\mathcal{B}$ relative to pole O.[2] The (positive) elements $I_{Oxx}, I_{Oyy}, I_{Ozz}$ are the *inertia moments* with respect to three coordinate axes of the reference frame, whereas the elements I_{Oxy}, I_{Oxz}, I_{Oyz} (of any sign) are said to be *products of inertia*.

The expression of the inertia tensor of a rigid body $\mathcal{B}$ depends both on the pole and the reference frame. If orientation of the reference frame with origin at O is changed according to a rotation matrix $\boldsymbol{R}$, the inertia tensor $\boldsymbol{I}'_O$ in the new frame is related to $\boldsymbol{I}_O$ by the relationship

$$\boldsymbol{I}_O = \boldsymbol{R}\boldsymbol{I}'_O\boldsymbol{R}^T. \tag{B.11}$$

The way an inertia tensor is transformed when the pole is changed can be inferred by the following equation, also known as *Steiner theorem* or parallel axis theorem:

$$\boldsymbol{I}_O = \boldsymbol{I}_C + m\boldsymbol{S}^T(\boldsymbol{p}_C)\boldsymbol{S}(\boldsymbol{p}_C), \tag{B.12}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass of $\mathcal{B}$, when expressed in a frame parallel to the frame with origin at O and with origin at the centre of mass C.

Since the inertia tensor is a symmetric positive definite matrix, there always exists a reference frame in which the inertia tensor attains a diagonal form; such a frame is said to be a *principal frame* (relative to pole O) and its coordinate axes are said to be *principal axes*. In the case when pole O coincides with the centre of mass, the frame is said to be a *central frame* and its axes are said to be *central axes*.

Notice that if the rigid body is moving with respect to the reference frame with origin at O, then the elements of the inertia tensor $\boldsymbol{I}_O$ become a function of time. With respect to a pole and a reference frame attached to the body (moving frame), instead, the elements of the inertia tensor represent six structural constants of the body which are known once the pole and reference frame have been specified.

[2] The symbol '$*$' has been used to avoid rewriting the symmetric elements.

Let $\dot{\boldsymbol{p}}$ be the velocity of a particle of $\mathcal{B}$ of elementary mass ρdV in frame O–xyz. The *linear momentum* of body $\mathcal{B}$ is defined as the vector

$$\boldsymbol{l} = \int_{V_\mathcal{B}} \dot{\boldsymbol{p}}\rho dV = m\dot{\boldsymbol{p}}_C. \tag{B.13}$$

Let Ω be any point in space and $\boldsymbol{p}_\Omega$ its position vector in frame O–xyz; then, the *angular momentum* of body $\mathcal{B}$ relative to pole Ω is defined as the vector

$$\boldsymbol{k}_\Omega = \int_{V_\mathcal{B}} \dot{\boldsymbol{p}} \times (\boldsymbol{p}_\Omega - \boldsymbol{p})\rho dV.$$

The pole can be either fixed or moving with respect to the reference frame. The angular momentum of a rigid body has the following notable expression:

$$\boldsymbol{k}_\Omega = \boldsymbol{I}_C\boldsymbol{\omega} + m\dot{\boldsymbol{p}}_C \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_C), \tag{B.14}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass, when expressed in a frame parallel to the reference frame with origin at the centre of mass.

The *forces* acting on a generic system of material particles can be distinguished into *internal* forces and *external* forces.

The internal forces, exerted by one part of the system on another, have null linear and angular momentum and thus they do not influence rigid body motion.

The external forces, exerted on the system by an agency outside the system, in the case of a rigid body $\mathcal{B}$ are distinguished into *active* forces and *reaction* forces.

The active forces can be either *concentrated* forces or *body* forces. The former are applied to specific points of $\mathcal{B}$, whereas the latter act on all elementary particles of the body. An example of body force is the *gravitational force* which, for any elementary particle of mass ρdV, is equal to $\boldsymbol{g}_0\rho dV$ where $\boldsymbol{g}_0$ is the gravity acceleration vector.

The reaction forces are those exerted because of surface contact between two or more bodies. Such forces can be distributed on the contact surfaces or they can be assumed to be concentrated.

For a rigid body $\mathcal{B}$ subject to gravitational force, as well as to active and or reaction forces $\boldsymbol{f}_1 \dots \boldsymbol{f}_n$ concentrated at points $\boldsymbol{p}_1 \dots \boldsymbol{p}_n$, the *resultant* of the external forces $\boldsymbol{f}$ and the *resultant moment* $\boldsymbol{\mu}_\Omega$ with respect to a pole Ω are respectively

$$\boldsymbol{f} = \int_{V_\mathcal{B}} \boldsymbol{g}_0\rho dV + \sum_{i=1}^{n} \boldsymbol{f}_i = m\boldsymbol{g}_0 + \sum_{i=1}^{n} \boldsymbol{f}_i \tag{B.15}$$

$$\boldsymbol{\mu}_\Omega = \int_{V_\mathcal{B}} \boldsymbol{g}_0 \times (\boldsymbol{p}_\Omega - \boldsymbol{p})\rho dV + \sum_{i=1}^{n} \boldsymbol{f}_i \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_i)$$
$$= m\boldsymbol{g}_0 \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_C) + \sum_{i=1}^{n} \boldsymbol{f}_i \times (\boldsymbol{p}_\Omega - \boldsymbol{p}_i). \tag{B.16}$$

In the case when $\boldsymbol{f}$ and $\boldsymbol{\mu}_\Omega$ are known and it is desired to compute the resultant moment with respect to a point Ω' other than Ω, the following relation holds:

$$\boldsymbol{\mu}_{\Omega'} = \boldsymbol{\mu}_\Omega + \boldsymbol{f} \times (\boldsymbol{p}_{\Omega'} - \boldsymbol{p}_\Omega). \tag{B.17}$$

Consider now a generic system of material particles subject to *external forces* of resultant $\boldsymbol{f}$ and resultant moment $\boldsymbol{\mu}_\Omega$. The motion of the system in a frame O–xyz is established by the following *fundamental principles of dynamics* (Newton laws of motion):

$$\boldsymbol{f} = \dot{\boldsymbol{l}} \tag{B.18}$$

$$\boldsymbol{\mu}_\Omega = \dot{\boldsymbol{k}}_\Omega \tag{B.19}$$

where Ω is a pole fixed or coincident with the centre of mass C of the system. These equations hold for any mechanical system and can be used even in the case of variable mass. For a system with constant mass, computing the time derivative of the momentum in (B.18) gives *Newton equations of motion* in the form

$$\boldsymbol{f} = m\ddot{\boldsymbol{p}}_C, \tag{B.20}$$

where the quantity on the right-hand side represents the *resultant of inertia forces*.

If, besides the assumption of constant mass, the assumption of rigid system holds too, the expression in (B.14) of the angular momentum with (B.19) yield *Euler equations of motion* in the form

$$\boldsymbol{\mu}_\Omega = \boldsymbol{I}_\Omega \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}_\Omega \boldsymbol{\omega}), \tag{B.21}$$

where the quantity on the right-hand side represents the *resultant moment of inertia forces*.

For a system constituted by a set of rigid bodies, the external forces obviously do not include the reaction forces exerted between the bodies belonging to the same system.

B.3 Work and Energy

Given a force $\boldsymbol{f}_i$ applied at a point of position $\boldsymbol{p}_i$ with respect to frame O–xyz, the *elementary work* of the force $\boldsymbol{f}_i$ on the displacement $d\boldsymbol{p}_i = \dot{\boldsymbol{p}}_i dt$ is defined as the scalar

$$dW_i = \boldsymbol{f}_i^T d\boldsymbol{p}_i.$$

For a rigid body $\mathcal{B}$ subject to a system of forces of resultant $\boldsymbol{f}$ and resultant moment $\boldsymbol{\mu}_Q$ with respect to any point Q of $\mathcal{B}$, the elementary work on the rigid displacement (B.6) is given by

$$dW = (\boldsymbol{f}^T \dot{\boldsymbol{p}}_Q + \boldsymbol{\mu}_Q^T \boldsymbol{\omega}) dt = \boldsymbol{f}^T d\boldsymbol{p}_Q + \boldsymbol{\mu}_Q^T \boldsymbol{\omega} dt. \tag{B.22}$$

The *kinetic energy* of a body $\mathcal{B}$ is defined as the scalar quantity

$$\mathcal{T} = \frac{1}{2} \int_{V_\mathcal{B}} \dot{\boldsymbol{p}}^T \dot{\boldsymbol{p}} \rho dV$$

which, for a rigid body, takes on the notable expression

$$\mathcal{T} = \frac{1}{2} m \dot{\boldsymbol{p}}_C^T \dot{\boldsymbol{p}}_C + \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{I}_C \boldsymbol{\omega} \tag{B.23}$$

where $\boldsymbol{I}_C$ is the inertia tensor relative to the centre of mass expressed in a frame parallel to the reference frame with origin at the centre of mass.

A system of position forces, i.e., the forces depending only on the positions of the points of application, is said to be *conservative* if the work done by each force is independent of the trajectory described by the point of application of the force but it depends only on the initial and final positions of the point of application. In this case, the elementary work of the system of forces is equal to minus the total differential of a scalar function termed *potential energy*, i.e.,

$$dW = -d\mathcal{U}. \tag{B.24}$$

An example of a conservative system of forces on a rigid body is the gravitational force, with which is associated the potential energy

$$\mathcal{U} = -\int_{V_\mathcal{B}} \boldsymbol{g}_0^T \boldsymbol{p} \rho dV = -m \boldsymbol{g}_0^T \boldsymbol{p}_C. \tag{B.25}$$

B.4 Constrained Systems

Consider a system $\mathcal{B}_r$ of r rigid bodies and assume that all the elements of $\mathcal{B}_r$ can reach any position in space. In order to find uniquely the position of all the points of the system, it is necessary to assign a vector $\boldsymbol{x} = [\, x_1 \quad \dots \quad x_p \,]^T$ of $6r = p$ parameters, termed *configuration*. These parameters are termed *Lagrange* or *generalized coordinates* of the *unconstrained* system $\mathcal{B}_r$, and p determines the number of *degrees of freedom* (DOFs).

Any limitation on the mobility of the system $\mathcal{B}_r$ is termed *constraint*. A constraint acting on $\mathcal{B}_r$ is said to be *holonomic* if it is expressed by a system of equations

$$\boldsymbol{h}(\boldsymbol{x}, t) = \boldsymbol{0}, \tag{B.26}$$

where $\boldsymbol{h}$ is a vector of dimensions $(s \times 1)$, with $s < m$. On the other hand, a constraint in the form $\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}, t) = \boldsymbol{0}$ which is nonintegrable is said to be *nonholonomic*. For simplicity, only equality (or *bilateral*) constraints are considered. If the equations in (B.26) do not explicitly depend on time, the constraint is said to be *scleronomic*.

On the assumption that $\boldsymbol{h}$ has continuous and continuously differentiable components, and its Jacobian $\partial \boldsymbol{h} / \partial \boldsymbol{x}$ has full rank, the equations in (B.26)

allow the elimination of s out of m coordinates of the system $\mathcal{B}_r$. With the remaining $n = m - s$ coordinates it is possible to determine uniquely the configurations of $\mathcal{B}_r$ satisfying the constraints (B.26). Such coordinates are the *Lagrange* or *generalized coordinates* and n is the number of *degrees of freedom* of the *unconstrained* system $\mathcal{B}_r$.[3]

The motion of a system $\mathcal{B}_r$ with n DOFs and holonomic equality constraints can be described by equations of the form

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{q}(t), t), \tag{B.27}$$

where $\boldsymbol{q}(t) = [\, q_1(t) \quad \dots \quad q_n(t) \,]^T$ is a vector of Lagrange coordinates.

The *elementary displacement* of system (B.27) relative to the interval $(t, t+dt)$ is defined as

$$d\boldsymbol{x} = \frac{\partial \boldsymbol{x}(\boldsymbol{q}, t)}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} dt + \frac{\partial \boldsymbol{x}(\boldsymbol{q}, t)}{\partial t} dt. \tag{B.28}$$

The *virtual displacement* of system (B.27) at time t, relative to an increment $\delta\boldsymbol{\lambda}$, is defined as the quantity

$$\delta\boldsymbol{x} = \frac{\partial \boldsymbol{x}(\boldsymbol{q}, t)}{\partial \boldsymbol{q}} \delta\boldsymbol{q}. \tag{B.29}$$

The difference between the elementary displacement and the virtual displacement is that the former is relative to an actual motion of the system in an interval $(t, t + dt)$ which is consistent with the constraints, while the latter is relative to an imaginary motion of the system when the constraints are made invariant and equal to those at time t.

For a system with time-invariant constraints, the equations of motion (B.27) become

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{q}(t)), \tag{B.30}$$

and then, by setting $\delta\boldsymbol{\lambda} = d\boldsymbol{\lambda} = \dot{\boldsymbol{\lambda}} dt$, the virtual displacements (B.29) coincide with the elementary displacements (B.28).

To the concept of virtual displacement can be associated that of *virtual work* of a system of forces, by considering a virtual displacement instead of an elementary displacement.

If external forces are distinguished into *active forces* and *reaction forces*, a direct consequence of the principles of dynamics (B.18), (B.19) applied to the system of rigid bodies $\mathcal{B}_r$ is that, for each virtual displacement, the following relation holds:

$$\delta W_m + \delta W_a + \delta W_h = 0, \tag{B.31}$$

where δW_m, δW_a, δW_h are the total virtual works done by the inertia, active, reaction forces, respectively.

[3] In general, the Lagrange coordinates of a constrained system have a local validity; in certain cases, such as the joint variables of a manipulator, they can have a global validity.

In the case of *frictionless* equality constraints, reaction forces are exerted orthogonally to the contact surfaces and the virtual work is always null. Hence, (B.31) reduces to

$$\delta W_m + \delta W_a = 0. \tag{B.32}$$

For a steady system, inertia forces are identically null. Then the condition for the equilibrium of system $\mathcal{B}_r$ is that the virtual work of the active forces is identically null on any virtual displacement, which gives the fundamental equation of *statics* of a constrained system

$$\delta W_a = 0 \tag{B.33}$$

known as *principle of virtual work*. Expressing (B.33) in terms of the increment $\delta\boldsymbol{\lambda}$ of generalized coordinates leads to

$$\delta W_a = \boldsymbol{\zeta}^T \delta\boldsymbol{q} = 0 \tag{B.34}$$

where $\boldsymbol{\zeta}$ denotes the $(n \times 1)$ vector of active *generalized* forces.

In the dynamic case, it is worth distinguishing active forces into *conservative* (that can be derived from a potential) and *nonconservative*. The virtual work of conservative forces is given by

$$\delta W_c = -\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}} \delta\boldsymbol{q}, \tag{B.35}$$

where $\mathcal{U}(\boldsymbol{\lambda})$ is the total potential energy of the system. The work of nonconservative forces can be expressed in the form

$$\delta W_{nc} = \boldsymbol{\xi}^T \delta\boldsymbol{q}, \tag{B.36}$$

where $\boldsymbol{\xi}$ denotes the vector of nonconservative generalized forces. It follows that the vector of active generalized forces is

$$\boldsymbol{\zeta} = \boldsymbol{\xi} - \left(\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}} \right)^T. \tag{B.37}$$

Moreover, the work of inertia forces can be computed from the total kinetic energy of system $\mathcal{T}$ as

$$\delta W_m = \left(\frac{\partial \mathcal{T}}{\partial \boldsymbol{q}} - \frac{d}{dt} \frac{\partial \mathcal{T}}{\partial \dot{\boldsymbol{q}}} \right) \delta\boldsymbol{q}. \tag{B.38}$$

Substituting (B.35), (B.36), (B.38) into (B.32) and observing that (B.32) holds true for any increment $\delta\boldsymbol{\lambda}$ leads to *Lagrange equations*

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}} \right)^T - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}} \right)^T = \boldsymbol{\xi}, \tag{B.39}$$

where

$$\mathcal{L} = \mathcal{T} - \mathcal{U} \tag{B.40}$$

is the *Lagrangian* function of the system. The equations in (B.39) completely describe the dynamic behaviour of an n-DOF system with holonomic equality constraints.

The sum of kinetic and potential energy of a system with time-invariant constraints is termed *Hamiltonian* function

$$\mathcal{H} = \mathcal{T} + \mathcal{U}. \tag{B.41}$$

Conservation of energy dictates that the time derivative of the Hamiltonian must balance the power generated by the nonconservative forces acting on the system, i.e.,

$$\frac{d\mathcal{H}}{dt} = \boldsymbol{\xi}^T \dot{\boldsymbol{q}}. \tag{B.42}$$

In view of (B.37), (B.41), the equation in (B.42) becomes

$$\frac{d\mathcal{T}}{dt} = \boldsymbol{\zeta}^T \dot{\boldsymbol{q}}. \tag{B.43}$$

Bibliography

The fundamental concepts of rigid-body mechanics and constrained systems can be found in classical texts such as [87, 154, 224]. An authoritative reference on rigid-body system dynamics is [187].

C

Feedback Control

As a premise to the study of manipulator decentralized control and centralized control, the fundamental principles of *feedback control* of *linear systems* are recalled, and an approach to the determination of control laws for *nonlinear systems* based on the use of *Lyapunov functions* is presented.

C.1 Control of Single-input/Single-output Linear Systems

According to classical *automatic control* theory of *linear time-invariant single-input/single-output systems*, in order to servo the output $y(t)$ of a system to a reference $r(t)$, it is worth adopting a *negative feedback control* structure. This structure indeed allows the use of approximate mathematical models to describe the input/output relationship of the system to control, since negative feedback has a potential for reducing the effects of system parameter variations and nonmeasurable disturbance inputs $d(t)$ on the output.

This structure can be represented in the *domain of complex variable* s as in the block scheme of Fig. C.1, where $G(s)$, $H(s)$ and $C(s)$ are the transfer functions of the system to control, the transducer and the controller, respectively. From this scheme it is easy to derive

$$Y(s) = W(s)R(s) + W_D(s)D(s), \tag{C.1}$$

where

$$W(s) = \frac{C(s)G(s)}{1 + C(s)G(s)H(s)} \tag{C.2}$$

is the *closed-loop input/output transfer function* and

$$W_D(s) = \frac{G(s)}{1 + C(s)G(s)H(s)} \tag{C.3}$$

is the *disturbance/output transfer function*.

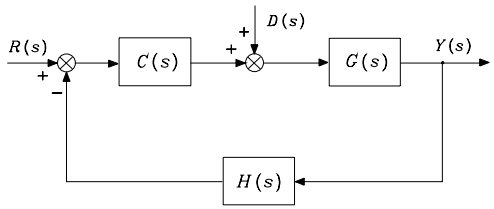

Fig. C.1. Feedback control structure

The goal of the controller design is to find a control structure $C(s)$ ensuring that the output variable $Y(s)$ tracks a reference input $R(s)$. Further, the controller should guarantee that the effects of the disturbance input $D(s)$ on the output variable are suitably reduced. The goal is then twofold, namely, *reference tracking* and *disturbance rejection*.

The basic problem for controller design consists of the determination of an action $C(s)$ which can make the system *asymptotically stable*. In the absence of positive or null real part pole/zero and zero/pole cancellation in the *open-loop* function $F(s) = C(s)G(s)H(s)$, a necessary and sufficient condition for asymptotic stability is that the *poles* of $W(s)$ and $W_D(s)$ have all *negative real parts*; such poles coincide with the zeros of the rational transfer function $1 + F(s)$. Testing for this condition can be performed by resorting to stability criteria, thus avoiding computation of the function zeros.

Routh criterion allows the determination of the sign of the real parts of the zeros of the function $1 + F(s)$ by constructing a table with the coefficients of the polynomial at the numerator of $1 + F(s)$ (*characteristic polynomial*).

Routh criterion is easy to apply for testing stability of a feedback system, but it does not provide a direct relationship between the open-loop function and stability of the closed-loop system. It is then worth resorting to *Nyquist criterion* which is based on the representation, in the complex plane, of the open-loop transfer function $F(s)$ evaluated in the *domain of real angular frequency* ($s = j\omega, -\infty < \omega < +\infty$).

Drawing of Nyquist plot and computation of the number of circles made by the vector representing the complex number $1 + F(j\omega)$ when ω continuously varies from $-\infty$ to $+\infty$ allows a test on whether or not the *closed-loop* system is asymptotically stable. It is also possible to determine the number of positive, null and negative real part roots of the characteristic polynomial, similarly to application of Routh criterion. Nonetheless, Nyquist criterion is based on the plot of the open-loop transfer function, and thus it allows the determination of a direct relationship between this function and closed-loop system stability. It is then possible from an examination of the Nyquist plot to draw suggestions on the controller structure $C(s)$ which ensures closed-loop system asymptotic stability.

If the closed-loop system is asymptotically stable, the *steady-state response* to a sinusoidal input $r(t)$, with $d(t) = 0$, is sinusoidal, too. In this case, the function $W(s)$, evaluated for $s = j\omega$, is termed *frequency response function*; the frequency response function of a feedback system can be assimilated to that of a low-pass filter with the possible occurrence of a *resonance peak* inside its *bandwidth*.

As regards the transducer, this should be chosen so that its bandwidth is much greater than the feedback system bandwidth, in order to ensure a nearly instantaneous response for any value of ω inside the bandwidth of $W(j\omega)$. Therefore, setting $H(j\omega) \approx H_0$ and assuming that the *loop gain* $|C(j\omega)G(j\omega)H_0| \gg 1$ in the same bandwidth, the expression in (C.1) for $s = j\omega$ can be approximated as

$$Y(j\omega) \approx \frac{R(j\omega)}{H_0} + \frac{D(j\omega)}{C(j\omega)H_0}.$$

Assuming $R(j\omega) = H_0 Y_d(j\omega)$ leads to

$$Y(j\omega) \approx Y_d(j\omega) + \frac{D(j\omega)}{C(j\omega)H_0}; \tag{C.4}$$

i.e., the output tracks the desired output $Y_d(j\omega)$ and the frequency components of the disturbance in the bandwidth of $W(j\omega)$ produce an effect on the output which can be reduced by increasing $|C(j\omega)H_0|$. Furthermore, if the disturbance input is a constant, the steady-state output is not influenced by the disturbance as long as $C(s)$ has at least a pole at the origin.

Therefore, a feedback control system is capable of establishing a proportional relationship between the desired output and the actual output, as evidenced by (C.4). This equation, however, requires that the frequency content of the input (desired output) be inside the frequency range for which the loop gain is much greater than unity.

The previous considerations show the advantage of including a *proportional action* and an *integral action* in the controller $C(s)$, leading to the transfer function

$$C(s) = K_I \frac{1 + sT_I}{s} \tag{C.5}$$

of a *proportional-integral controller* (PI); T_I is the time constant of the integral action and the quantity $K_I T_I$ is called proportional sensitivity.

The adoption of a PI controller is effective for low-frequency response of the system, but it may involve a reduction of *stability margins* and/or a reduction of closed-loop system bandwidth. To avoid these drawbacks, a *derivative action* can be added to the proportional and integral actions, leading to the transfer function

$$C(s) = K_I \frac{1 + sT_I + s^2 T_D T_I}{s} \tag{C.6}$$

of a *proportional-integral-derivative controller* (PID); T_D denotes the time constant of the derivative action. Notice that physical realizability of (C.6)

demands the introduction of a high-frequency pole which little influences the input/output relationship in the system bandwidth. The transfer function in (C.6) is characterized by the presence of two zeros which provide a stabilizing action and an enlargement of the closed-loop system bandwidth. Bandwidth enlargement implies shorter *response time* of the system, in terms of both variations of the reference signal and recovery action of the feedback system to output variations induced by the disturbance input.

The parameters of the adopted control structure should be chosen so as to satisfy requirements on the system behaviour at *steady state* and during the *transient*. Classical tools to determine such parameters are the *root locus* in the domain of the complex variable s or the *Nichols chart* in the domain of the real angular frequency ω. The two tools are conceptually equivalent. Their potential is different in that root locus allows a control law to be found which assigns the exact parameters of the closed-loop system time response, whereas Nichols chart allows a controller to be specified which confers good transient and steady-state behaviour to the system response.

A feedback system with strict requirements on the steady-state and transient behaviour, typically, has a response that can be assimilated to that of a *second-order system*. In fact, even for closed-loop functions of greater order, it is possible to identify a pair of complex conjugate poles whose real part absolute value is smaller than the real part absolute values of the other poles. Such a pair of poles is *dominant* in that its contribution to the transient response prevails over that of the other poles. It is then possible to approximate the input/output relationship with the transfer function

$$W(s) = \frac{k_W}{1 + \dfrac{2\zeta s}{\omega_n} + \dfrac{s^2}{\omega_n^2}} \tag{C.7}$$

which has to be realized by a proper choice of the controller. Regarding the values to assign to the parameters characterizing the transfer function in (C.7), the following remarks are in order. The constant k_W represents the input/output *steady-state gain*, which is equal to $1/H_0$ if $C(s)G(s)H_0$ has at least a pole at the origin. The *natural frequency* ω_n is the modulus of the complex conjugate poles, whose real part is given by $-\zeta\omega_n$ where ζ is the *damping ratio* of the pair of poles.

The influence of parameters ζ and ω_n on the closed-loop frequency response can be evaluated in terms of the resonance peak magnitude

$$M_r = \frac{1}{2\zeta\sqrt{1-\zeta^2}},$$

occurring at the resonant frequency

$$\omega_r = \omega_n\sqrt{1-2\zeta^2},$$

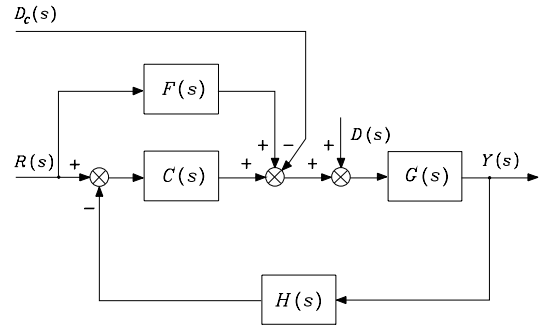

Fig. C.2. Feedback control structure with feedforward compensation

and of the 3 dB bandwidth

$$\omega_3 = \omega_n\sqrt{1-2\zeta^2+\sqrt{2-4\zeta^2+4\zeta^4}}.$$

A step input is typically used to characterize the transient response in the time domain. The influence of parameters ζ and ω_n on the *step response* can be evaluated in terms of the percentage of *overshoot*

$$s\% = 100\exp(-\pi\zeta/\sqrt{1-\zeta^2}),$$

of the *rise time*

$$t_r \approx \frac{1.8}{\omega_n}$$

and of the *settling time* within 1%

$$t_s = \frac{4.6}{\zeta\omega_n}.$$

The adoption of a *feedforward compensation* action represents a feasible solution both for tracking a time-varying reference input and for enhancing rejection of the effects of a disturbance on the output. Consider the general scheme in Fig. C.2. Let $R(s)$ denote a given input reference and $D_c(s)$ denote a computed estimate of the disturbance $D(s)$; the introduction of the feedforward action yields the input/output relationship

$$Y(s) = \left(\frac{C(s)G(s)}{1+C(s)G(s)H(s)} + \frac{F(s)G(s)}{1+C(s)G(s)H(s)}\right)R(s) + \frac{G(s)}{1+C(s)G(s)H(s)}\big(D(s)-D_c(s)\big). \tag{C.8}$$

By assuming that the desired output is related to the reference input by a constant factor K_d and regarding the transducer as an instantaneous system ($H(s) \approx H_0 = 1/K_d$) for the current operating conditions, the choice

$$F(s) = \frac{K_d}{G(s)} \tag{C.9}$$

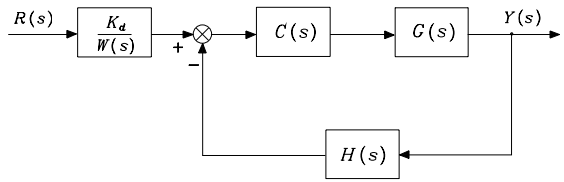

Fig. C.3. Feedback control structure with inverse model technique

yields the input/output relationship

$$Y(s) = Y_d(s) + \frac{G(s)}{1 + C(s)G(s)H_0}\big(D(s) - D_c(s)\big). \tag{C.10}$$

If $|C(j\omega)G(j\omega)H_0| \gg 1$, the effect of the disturbance on the output is further reduced by means of an accurate estimate of the disturbance.

Feedforward compensation technique may lead to a solution, termed *inverse model control*, illustrated in the scheme of Fig. C.3. It should be remarked, however, that such a solution is based on dynamics cancellation, and thus it can be employed only for a minimum-phase system, i.e., a system whose poles and zeros have all strictly negative real parts. Further, one should consider physical realizability issues as well as effects of parameter variations which prevent perfect cancellation.

C.2 Control of Nonlinear Mechanical Systems

If the system to control does not satisfy the linearity property, the control design problem becomes more complex. The fact that a *system* is qualified as *nonlinear*, whenever linearity does not hold, leads to understanding how it is not possible to resort to general techniques for control design, but it is necessary to face the problem for each class of nonlinear systems which can be defined through imposition of special properties.

On the above premise, the control design problem of nonlinear systems described by the dynamic model

$$\boldsymbol{H}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{u} \tag{C.11}$$

is considered, where $[\,\boldsymbol{x}^T \quad \dot{\boldsymbol{x}}^T\,]^T$ denotes the $(2n \times 1)$ *state* vector of the system, $\boldsymbol{u}$ is the $(n \times 1)$ *input* vector, $\boldsymbol{H}(\boldsymbol{x})$ is an $(n \times n)$ *positive definite* (and thus invertible) matrix depending on $\boldsymbol{x}$, and $\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ is an $(n \times 1)$ vector depending on state. Several *mechanical systems* can be reduced to this class, including manipulators with rigid links and joints.

The *control* law can be found through a nonlinear compensating action obtained by choosing the following *nonlinear state feedback* law (*inverse dynamics* control):

$$\boldsymbol{u} = \widehat{\boldsymbol{H}}(\boldsymbol{x})\boldsymbol{v} + \widehat{\boldsymbol{h}}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{C.12}$$

where $\widehat{\boldsymbol{H}}(\boldsymbol{x})$ and $\widehat{\boldsymbol{h}}(\boldsymbol{x})$ respectively denote the *estimates* of the terms $\boldsymbol{H}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$, computed on the basis of measures on the system state, and $\boldsymbol{v}$ is a new control input to be defined later. In general, it is

$$\widehat{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}) + \Delta\boldsymbol{H}(\boldsymbol{x}) \tag{C.13}$$

$$\widehat{\boldsymbol{h}}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \Delta\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}}) \tag{C.14}$$

because of the unavoidable modelling approximations or as a consequence of an intentional simplification in the compensating action. Substituting (C.12) into (C.11) and accounting for (C.13), (C.14) yields

$$\ddot{\boldsymbol{x}} = \boldsymbol{v} + \boldsymbol{z}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{v}) \tag{C.15}$$

where

$$\boldsymbol{z}(\boldsymbol{x}, \dot{\boldsymbol{x}}, \boldsymbol{v}) = \boldsymbol{H}^{-1}(\boldsymbol{x})\big(\Delta\boldsymbol{H}(\boldsymbol{x})\boldsymbol{v} + \Delta\boldsymbol{h}(\boldsymbol{x}, \dot{\boldsymbol{x}})\big).$$

If *tracking* of a trajectory $(\boldsymbol{x}_d(t), \dot{\boldsymbol{x}}_d(t), \ddot{\boldsymbol{x}}_d(t))$ is desired, the tracking error can be defined as

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{x}_d - \boldsymbol{x} \\ \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}} \end{bmatrix} \tag{C.16}$$

and it is necessary to derive the error dynamics equation to study convergence of the actual state to the desired one. To this end, the choice

$$\boldsymbol{v} = \ddot{\boldsymbol{x}}_d + \boldsymbol{w}(\boldsymbol{e}), \tag{C.17}$$

substituted into (C.15), leads to the error equation

$$\dot{\boldsymbol{e}} = \boldsymbol{F}\boldsymbol{e} - \boldsymbol{G}\boldsymbol{w}(\boldsymbol{e}) - \boldsymbol{G}\boldsymbol{z}(\boldsymbol{e}, \boldsymbol{x}_d, \dot{\boldsymbol{x}}_d, \ddot{\boldsymbol{x}}_d), \tag{C.18}$$

where the $(2n \times 2n)$ and $(2n \times n)$ matrices, respectively,

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} \boldsymbol{O} \\ \boldsymbol{I} \end{bmatrix}$$

follow from the error definition in (C.16). Control law design consists of finding the error function $\boldsymbol{w}(\boldsymbol{e})$ which makes (C.18) *globally asymptotically stable*,[1] i.e.,

$$\lim_{t \to \infty} \boldsymbol{e}(t) = \boldsymbol{0}.$$

In the case of *perfect* nonlinear compensation ($\boldsymbol{z}(\cdot) = \boldsymbol{0}$), the simplest choice of the control action is the *linear* one

$$\begin{aligned} \boldsymbol{w}(\boldsymbol{e}) &= -\boldsymbol{K}_P(\boldsymbol{x}_d - \boldsymbol{x}) - \boldsymbol{K}_D(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}}) \\ &= [\,-\boldsymbol{K}_P \quad -\boldsymbol{K}_D\,]\,\boldsymbol{e}, \end{aligned} \tag{C.19}$$

[1] *Global* asymptotic stability is invoked to remark that the equilibrium state is asymptotically stable for any perturbation.

where asymptotic stability of the error equation is ensured by choosing *positive definite* matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$. The error transient behaviour is determined by the eigenvalues of the matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix} \tag{C.20}$$

characterizing the error dynamics

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e}. \tag{C.21}$$

If compensation is *imperfect*, then $\boldsymbol{z}(\cdot)$ cannot be neglected and the error equation in (C.18) takes on the general form

$$\dot{\boldsymbol{e}} = \boldsymbol{f}(\boldsymbol{e}). \tag{C.22}$$

It may be worth choosing the control law $\boldsymbol{w}(\boldsymbol{e})$ as the sum of a nonlinear term and a linear term of the kind in (C.19); in this case, the error equation can be written as

$$\dot{\boldsymbol{e}} = \boldsymbol{A}\boldsymbol{e} + \boldsymbol{k}(\boldsymbol{e}), \tag{C.23}$$

where $\boldsymbol{A}$ is given by (C.20) and $\boldsymbol{k}(\boldsymbol{e})$ is available to make the system globally asymptotically stable. The equations in (C.22), (C.23) express nonlinear differential equations of the error. To test for stability and obtain advise on the choice of suitable control actions, one may resort to *Lyapunov direct method* illustrated below.

C.3 Lyapunov Direct Method

The philosophy of the *Lyapunov direct method* is the same as that of most methods used in control engineering to study stability, namely, testing for stability without solving the differential equations describing the dynamic system.

This method can be presented in short on the basis of the following reasoning. If it is possible to associate an energy-based description with a (linear or nonlinear) autonomous dynamic system and, for each system state with the exception of the equilibrium state, the time rate of such energy is negative, then energy decreases along any system trajectory until it attains its minimum at the equilibrium state; this argument justifies an intuitive concept of stability.

With reference to (C.22), by setting $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, the *equilibrium state* is $\boldsymbol{e} = \boldsymbol{0}$. A scalar function $V(\boldsymbol{e})$ of the system state, continuous together with its first derivative, is defined a *Lyapunov function* if the following properties hold:

$$\begin{aligned} V(\boldsymbol{e}) &> 0 \qquad && \forall \boldsymbol{e} \neq \boldsymbol{0} \\ V(\boldsymbol{e}) &= 0 \qquad && \boldsymbol{e} = \boldsymbol{0} \\ \dot{V}(\boldsymbol{e}) &< 0 \qquad && \forall \boldsymbol{e} \neq \boldsymbol{0} \\ V(\boldsymbol{e}) &\to \infty \qquad && \|\boldsymbol{e}\| \to \infty. \end{aligned}$$

The existence of such a function ensures *global asymptotic stability* of the equilibrium $\boldsymbol{e} = \boldsymbol{0}$. In practice, the equilibrium $\boldsymbol{e} = \boldsymbol{0}$ is globally asymptotically stable if a positive definite, radially unbounded function $\boldsymbol{V}(\boldsymbol{e})$ is found so that its time derivative along the system trajectories is negative definite.

If positive definiteness of $V(\boldsymbol{e})$ is realized by the adoption of a *quadratic form*, i.e.,

$$V(\boldsymbol{e}) = \boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} \tag{C.24}$$

with $\boldsymbol{Q}$ a symmetric positive definite matrix, then in view of (C.22) it follows

$$\dot{V}(\boldsymbol{e}) = 2\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{f}(\boldsymbol{e}). \tag{C.25}$$

If $\boldsymbol{f}(\boldsymbol{e})$ is so as to render the function $\dot{V}(\boldsymbol{e})$ negative definite, the function $V(\boldsymbol{e})$ is a *Lyapunov function*, since the choice (C.24) allows system global asymptotic stability to be proved. If $\dot{V}(\boldsymbol{e})$ in (C.25) is not negative definite for the given $V(\boldsymbol{e})$, nothing can be inferred on the stability of the system, since the Lyapunov method gives only a *sufficient* condition. In such cases one should resort to different choices of $V(\boldsymbol{e})$ in order to find, if possible, a negative definite $\dot{V}(\boldsymbol{e})$.

In the case when the property of negative definiteness does not hold, but $\dot{V}(\boldsymbol{e})$ is only *negative semi-definite*

$$\dot{V}(\boldsymbol{e}) \leq 0,$$

global asymptotic stability of the equilibrium state is ensured if the only system trajectory for which $\dot{V}(\boldsymbol{e})$ is *identically* null ($\dot{V}(\boldsymbol{e}) \equiv 0$) is the equilibrium trajectory $\boldsymbol{e} \equiv \boldsymbol{0}$ (a consequence of *La Salle theorem*).

Finally, consider the stability problem of the nonlinear system in the form (C.23); under the assumption that $\boldsymbol{k}(\boldsymbol{0}) = \boldsymbol{0}$, it is easy to verify that $\boldsymbol{e} = \boldsymbol{0}$ is an equilibrium state for the system. The choice of a Lyapunov function candidate as in (C.24) leads to the following expression for its derivative:

$$\dot{V}(\boldsymbol{e}) = \boldsymbol{e}^T(\boldsymbol{A}^T\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A})\boldsymbol{e} + 2\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{k}(\boldsymbol{e}). \tag{C.26}$$

By setting

$$\boldsymbol{A}^T\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A} = -\boldsymbol{P}, \tag{C.27}$$

the expression in (C.26) becomes

$$\dot{V}(\boldsymbol{e}) = -\boldsymbol{e}^T\boldsymbol{P}\boldsymbol{e} + 2\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{k}(\boldsymbol{e}). \tag{C.28}$$

The matrix equation in (C.27) is said to be a *Lyapunov equation*; for any choice of a symmetric positive definite matrix $\boldsymbol{P}$, the solution matrix $\boldsymbol{Q}$ exists

and is symmetric positive definite if and only if the eigenvalues of $\boldsymbol{A}$ have all negative real parts. Since matrix $\boldsymbol{A}$ in (C.20) verifies such condition, it is always possible to assign a positive definite matrix $\boldsymbol{P}$ and find a positive definite matrix solution $\boldsymbol{Q}$ to (C.27). It follows that the first term on the right-hand side of (C.28) is negative definite and the stability problem is reduced to searching a control law so that $\boldsymbol{k}(\boldsymbol{e})$ renders the total $\dot{V}(\boldsymbol{e})$ negative (semi-)definite.

It should be underlined that La Salle theorem does not hold for *time-varying* systems (also termed *non-autonomous*) in the form

$$\dot{\boldsymbol{e}} = f(\boldsymbol{e}, t).$$

In this case, a conceptually analogous result which might be useful is the following, typically referred to as *Barbalat lemma* — of which it is indeed a consequence. Given a scalar function $V(\boldsymbol{e}, t)$ so that

1. $V(\boldsymbol{e}, t)$ is lower bounded
2. $\dot{V}(\boldsymbol{e}, t) \leq 0$
3. $\dot{V}(\boldsymbol{e}, t)$ is *uniformly continuous*

then it is $\lim_{t\to\infty} \dot{V}(\boldsymbol{e}, t) = 0$. Conditions 1 and 2 imply that $V(\boldsymbol{e}, t)$ has a bounded limit for $t \to \infty$. Since it is not easy to verify the property of uniform continuity from the definition, Condition 3 is usually replaced by

3'. $\ddot{V}(\boldsymbol{e}, t)$ is bounded

which is sufficient to guarantee validity of Condition 3. Barbalat lemma can obviously be used for time-invariant (autonomous) dynamic systems as an alternative to La Salle theorem, with respect to which some conditions are relaxed; in particular, $V(\boldsymbol{e})$ needs not necessarily be positive definite.

Bibliography

Linear systems analysis can be found in classical texts such as [61]. For the control of these systems see [82, 171]. For the analysis of nonlinear systems see [109]. Control of nonlinear mechanical systems is dealt with in [215].

D

Differential Geometry

The analysis of mechanical systems subject to nonholonomic constraints, such as wheeled mobile robots, requires some basic concepts of differential geometry and nonlinear controllability theory, that are briefly recalled in this appendix.

D.1 Vector Fields and Lie Brackets

For simplicity, the case of vectors $\boldsymbol{x} \in \mathbb{R}^n$ is considered. The tangent space at $\boldsymbol{x}$ (intuitively, the space of velocities of trajectories passing through $\boldsymbol{x}$) is hence denoted by $T_{\boldsymbol{x}}(\mathbb{R}^n)$. The presented notions are however valid in the more general case in which a *differentiable manifold* (i.e., a space that is locally diffeomorphic to $\mathbb{R}^n$) is considered in place of a Euclidean space.

A *vector field* $\boldsymbol{g} : \mathbb{R}^n \mapsto T_x(\mathbb{R}^n)$ is a mapping that assigns to each point $\boldsymbol{x} \in \mathbb{R}^n$ a tangent vector $\boldsymbol{g}(\boldsymbol{x}) \in T_x(\mathbb{R}^n)$. In the following it is always assumed that vector fields are *smooth*, i.e., such that the associated mappings are of class C^∞.

If the vector field $\boldsymbol{g}(\boldsymbol{x})$ is used to define a differential equation as in

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}), \tag{D.1}$$

the *flow* $\boldsymbol{\phi}_t^{\boldsymbol{g}}(\boldsymbol{x})$ of $\boldsymbol{g}$ is the mapping that associates to each point $\boldsymbol{x}$ the value at time t of the solution of (D.1) evolving from $\boldsymbol{x}$ at time 0, or

$$\frac{d}{dt}\boldsymbol{\phi}_t^{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{\phi}_t^{\boldsymbol{g}}(\boldsymbol{x})). \tag{D.2}$$

The family of mappings $\{\boldsymbol{\phi}_t^{\boldsymbol{g}}\}$ is a one-parameter (i.e., t) group under the composition operator

$$\boldsymbol{\phi}_{t_1}^{\boldsymbol{g}} \circ \boldsymbol{\phi}_{t_2}^{\boldsymbol{g}} = \boldsymbol{\phi}_{t_1+t_2}^{\boldsymbol{g}}.$$

For example, for time-invariant linear systems it is $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and the flow is the linear operator $\boldsymbol{\phi}_t^{\boldsymbol{g}} = e^{\boldsymbol{A}t}$.

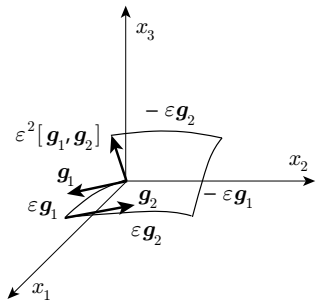

Fig. D.1. The net displacement of system (D.4) under the input sequence (D.5) is directed as the Lie bracket of the two vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$

Given two vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, the composition of their flows is non-commutative in general:

$$\phi_t^{\boldsymbol{g}_1} \circ \phi_s^{\boldsymbol{g}_2} \neq \phi_s^{\boldsymbol{g}_2} \circ \phi_t^{\boldsymbol{g}_1}.$$

The vector field $[\boldsymbol{g}_1, \boldsymbol{g}_2]$ defined as

$$[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x}) = \frac{\partial \boldsymbol{g}_2}{\partial x}\boldsymbol{g}_1(\boldsymbol{x}) - \frac{\partial \boldsymbol{g}_1}{\partial x}\boldsymbol{g}_2(\boldsymbol{x}) \tag{D.3}$$

is called *Lie bracket* of $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. The two vector field $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ *commute* if $[\boldsymbol{g}_1, \boldsymbol{g}_2] = 0$.

The Lie bracket operation has an interesting interpretation. Consider the driftless dynamic system

$$\dot{\boldsymbol{x}} = \boldsymbol{g}_1(\boldsymbol{x})u_1 + \boldsymbol{g}_2(\boldsymbol{x})u_2 \tag{D.4}$$

associated with the vector fields $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. If the inputs u_1 and u_2 are never active simultaneously, the solution of the differential equation (D.4) can be obtained by composing the flows of $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. In particular, consider the following input sequence:

$$u(t) = \begin{cases} u_1(t) = +1, u_2(t) = 0 & t \in [0, \varepsilon) \\ u_1(t) = 0, u_2(t) = +1 & t \in [\varepsilon, 2\varepsilon) \\ u_1(t) = -1, u_2(t) = 0 & t \in [2\varepsilon, 3\varepsilon) \\ u_1(t) = 0, u_2(t) = -1 & t \in [3\varepsilon, 4\varepsilon), \end{cases} \tag{D.5}$$

where ε is an infinitesimal time interval. The solution of (D.4) at time $t = 4\varepsilon$ can be obtained by following first the flow of $\boldsymbol{g}_1$, then of $\boldsymbol{g}_2$, then of $-\boldsymbol{g}_1$, and finally of $-\boldsymbol{g}_2$ (see Fig. D.1). By computing $\boldsymbol{x}(\varepsilon)$ through a series expansion at $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ along $\boldsymbol{g}_1$, then $\boldsymbol{x}(2\varepsilon)$ as a series expansion at $\boldsymbol{x}(\varepsilon)$ along $\boldsymbol{g}_2$, and so on, one obtains

$$\begin{aligned} \boldsymbol{x}(4\varepsilon) &= \phi_\varepsilon^{-\boldsymbol{g}_2} \circ \phi_\varepsilon^{-\boldsymbol{g}_1} \circ \phi_\varepsilon^{\boldsymbol{g}_2} \circ \phi_\varepsilon^{\boldsymbol{g}_1}(\boldsymbol{x}_0) \\ &= \boldsymbol{x}_0 + \varepsilon^2 \left(\frac{\partial \boldsymbol{g}_2}{\partial x}\boldsymbol{g}_1(\boldsymbol{x}_0) - \frac{\partial \boldsymbol{g}_1}{\partial x}\boldsymbol{g}_2(\boldsymbol{x}_0) \right) + O(\varepsilon^3). \end{aligned}$$

If $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ commute, the net displacement resulting from the input sequence (D.5) is zero.

The above expression shows that, at each point $\boldsymbol{x}$, infinitesimal motion of the driftless system (D.4) is possible not only in the directions belonging to the linear span of $\boldsymbol{g}_1(\boldsymbol{x})$ and $\boldsymbol{g}_2(\boldsymbol{x})$, but also in the direction of their Lie bracket $[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x})$. It can be proven that more complicated input sequences can be used to generate motion in the direction of higher-order Lie brackets, such as $[\boldsymbol{g}_1, [\boldsymbol{g}_1, \boldsymbol{g}_2]]$.

Similar constructive procedures can be given for systems with a *drift*[1] vector field, such as the following:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1(\boldsymbol{x})u_1 + \boldsymbol{g}_2(\boldsymbol{x})u_2. \tag{D.6}$$

Using appropriate input sequences, it is possible to generate motion in the direction of Lie brackets involving the vector field $\boldsymbol{f}$ as well as $\boldsymbol{g}_j$, $j = 1, 2$.

Example D.1

For a single-input linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{b}\,u,$$

the drift and input vector fields are $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{b}$, respectively. The following Lie brackets:

$$\begin{aligned} -[\boldsymbol{f}, \boldsymbol{g}] &= \boldsymbol{A}\boldsymbol{b} \\ [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}]] &= \boldsymbol{A}^2\boldsymbol{b} \\ -[\boldsymbol{f}, [\boldsymbol{f}, [\boldsymbol{f}, \boldsymbol{g}]]] &= \boldsymbol{A}^3\boldsymbol{b} \\ &\vdots \end{aligned}$$

represent well-known directions in which it is possible to move the system.

The *Lie derivative* of the scalar function $\alpha : \mathbb{R}^n \mapsto \mathbb{R}$ along vector field $\boldsymbol{g}$ is defined as

$$L_{\boldsymbol{g}}\,\alpha(\boldsymbol{x}) = \frac{\partial \alpha}{\partial \boldsymbol{x}}\,\boldsymbol{g}(\boldsymbol{x}). \tag{D.7}$$

The following properties of Lie brackets are useful in computation:

$$\begin{aligned} &[\boldsymbol{f}, \boldsymbol{g}] = -[\boldsymbol{g}, \boldsymbol{f}] && \text{(skew-symmetry)} \\ &[\boldsymbol{f}, [\boldsymbol{g}, \boldsymbol{h}]] + [\boldsymbol{h}, [\boldsymbol{f}, \boldsymbol{g}]] + [\boldsymbol{g}, [\boldsymbol{h}, \boldsymbol{f}]] = 0 && \text{(Jacobi identity)} \\ &[\alpha\boldsymbol{f}, \beta\boldsymbol{g}] = \alpha\beta[\boldsymbol{f}, \boldsymbol{g}] + \alpha(L_{\boldsymbol{f}}\beta)\boldsymbol{g} - \beta(L_{\boldsymbol{g}}\alpha)\boldsymbol{f} && \text{(chain rule)} \end{aligned}$$

[1] This term emphasizes how the presence of $\boldsymbol{f}$ will in general force the system to move ($\dot{\boldsymbol{x}} \neq \boldsymbol{0}$) even in the absence of inputs.

with $\alpha, \beta: \mathbb{R}^n \mapsto \mathbb{R}$. The vector space $\mathcal{V}(\mathbb{R}^n)$ of smooth vector fields on $\mathbb{R}^n$, equipped with the Lie bracket operation, is a *Lie algebra*.

The *distribution* Δ associated with the m vector fields $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$ is the mapping that assigns to each point $\boldsymbol{x} \in \mathbb{R}^n$ the subspace of $T_x(\mathbb{R}^n)$ defined as

$$\Delta(\boldsymbol{x}) = \text{span}\{\boldsymbol{g}_1(\boldsymbol{x}), \ldots, \boldsymbol{g}_m(\boldsymbol{x})\}. \tag{D.8}$$

Often, a shorthand notation is used:

$$\Delta = \text{span}\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}.$$

The distribution Δ is *nonsingular* if $\dim \Delta(\boldsymbol{x}) = r$, with r constant for all $\boldsymbol{x}$. In this case, r is called the *dimension* of the distribution. Moreover, Δ is called *involutive* if it is closed under the Lie bracket operation:

$$[\boldsymbol{g}_i, \boldsymbol{g}_j] \in \Delta \qquad \forall \boldsymbol{g}_i, \boldsymbol{g}_j \in \Delta.$$

The *involutive closure* $\bar{\Delta}$ of a distribution Δ is its closure under the Lie bracket operation. Hence, Δ is involutive if and only if $\bar{\Delta} = \Delta$. Note that the distribution $\Delta = \text{span}\{\boldsymbol{g}\}$ associated with a single vector field is always involutive, because $[\boldsymbol{g}, \boldsymbol{g}](\boldsymbol{x}) = \boldsymbol{0}$.

Example D.2

The distribution

$$\Delta = \text{span}\{\boldsymbol{g}_1, \boldsymbol{g}_2\} = \text{span}\left\{ \begin{bmatrix} \cos x_3 \\ \sin x_3 \\ 0 \end{bmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

is nonsingular and has dimension 2. It is not involutive, because the Lie bracket

$$[\boldsymbol{g}_1, \boldsymbol{g}_2](\boldsymbol{x}) = \begin{bmatrix} \sin x_3 \\ -\cos x_3 \\ 0 \end{bmatrix}$$

is always linearly independent of $\boldsymbol{g}_1(\boldsymbol{x})$ and $\boldsymbol{g}_2(\boldsymbol{x})$. Its involutive closure is therefore

$$\bar{\Delta} = \text{span}\{\boldsymbol{g}_1, \boldsymbol{g}_2, [\boldsymbol{g}_1, \boldsymbol{g}_2]\}.$$

D.2 Nonlinear Controllability

Consider a nonlinear dynamic system of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \sum_{j=1}^{m} \boldsymbol{g}_j(\boldsymbol{x}) u_j, \tag{D.9}$$

that is called *affine* in the inputs u_j. The state $\boldsymbol{x}$ takes values in $\mathbb{R}^n$, while each component u_j of the control input $\boldsymbol{u} \in \mathbb{R}^m$ takes values in the class $\mathcal{U}$ of piecewise-constant functions.

Denote by $\boldsymbol{x}(t, 0, \boldsymbol{x}_0, \boldsymbol{u})$ the solution of (D.9) at time $t \geq 0$, corresponding to an input $\boldsymbol{u}: [0, t] \to \mathcal{U}$ and an initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$. Such a solution exists and is unique provided that the drift vector field $\boldsymbol{f}$ and the input vector fields $\boldsymbol{g}_j$ are of class C^∞. System (D.9) is said to be *controllable* if, for any choice of $\boldsymbol{x}_1, \boldsymbol{x}_2$ in $\mathbb{R}^n$, there exists a time instant T and an input $\boldsymbol{u}: [0, T] \to \mathcal{U}$ such that $\boldsymbol{x}(T, 0, \boldsymbol{x}_1, \boldsymbol{u}) = \boldsymbol{x}_2$.

The *accessibility algebra* $\mathcal{A}$ of system (D.9) is the smallest subalgebra of $\mathcal{V}(\mathbb{R}^n)$ that contains $\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m$. By definition, all the Lie brackets that can be generated using these vector fields belong to $\mathcal{A}$. The *accessibility distribution* $\Delta_{\mathcal{A}}$ of system (D.9) is defined as

$$\Delta_{\mathcal{A}} = \text{span}\{\boldsymbol{v} | \boldsymbol{v} \in \mathcal{A}\}. \tag{D.10}$$

In other words, $\Delta_{\mathcal{A}}$ is the involutive closure of $\Delta = \text{span}\{\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$.

The computation of $\Delta_{\mathcal{A}}$ may be organized as an iterative procedure

$$\Delta_{\mathcal{A}} = \text{span}\left\{\boldsymbol{v} | \boldsymbol{v} \in \Delta_i, \forall i \geq 1\right\},$$

with

$$\begin{aligned} \Delta_1 &= \Delta = \text{span}\{\boldsymbol{f}, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\} \\ \Delta_i &= \Delta_{i-1} + \text{span}\{[\boldsymbol{g}, \boldsymbol{v}] | \, \boldsymbol{g} \in \Delta_1, \boldsymbol{v} \in \Delta_{i-1}\}, \quad i \geq 2. \end{aligned}$$

This procedure stops after κ steps, where κ is the smallest integer such that $\Delta_{\kappa+1} = \Delta_\kappa = \Delta_{\mathcal{A}}$. This number is called the *nonholonomy degree* of the system and is related to the 'level' of Lie brackets that must be included in $\Delta_{\mathcal{A}}$. Since $\dim \Delta_{\mathcal{A}} \leq n$, it is $\kappa \leq n - m$ necessarily.

If system (D.9) is driftless

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x}) u_i, \tag{D.11}$$

the accessibility distribution $\Delta_{\mathcal{A}}$ associated with vector fields $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m$ characterizes its controllability. In particular, system (D.11) is controllable if and only if the following *accessibility rank condition* holds:

$$\dim \Delta_{\mathcal{A}}(\boldsymbol{x}) = n. \tag{D.12}$$

Note that for driftless systems the iterative procedure for building $\Delta_{\mathcal{A}}$ starts with $\Delta_1 = \Delta = \text{span}\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_m\}$, and therefore $\kappa \leq n - m + 1$.

For systems in the general form (D.9), condition (D.12) is only necessary for controllability. There are, however, two notable exceptions:

- If system (D.11) is controllable, the system with drift obtained by performing a *dynamic extension* of (D.11)

$$\dot{\boldsymbol{x}} = \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{x}) v_i \tag{D.13}$$

$$\dot{v}_i = u_i, \qquad i = 1, \ldots, m, \tag{D.14}$$

  i.e., by adding an integrator on each input channel, is also controllable.
- For a linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \sum_{j=1}^{m} \boldsymbol{b}_j u_j = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

  (D.12) becomes

$$\varrho\left([\, \boldsymbol{B} \quad \boldsymbol{A}\boldsymbol{B} \quad \boldsymbol{A}^2\boldsymbol{B} \quad \ldots \quad \boldsymbol{A}^{n-1}\boldsymbol{B} \,]\right) = n, \tag{D.15}$$

  i.e., the well-known necessary and sufficient condition for controllability due to Kalman.

Bibliography

The concepts briefly recalled in this appendix can be studied in detail in various tests of differential geometry [94, 20] and nonlinear control theory [104, 168, 195].

References

1. C. Abdallah, D. Dawson, P. Dorato, M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24–30, 1991.
2. M. Aicardi, G. Casalino, A. Bicchi, A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.
3. J.S. Albus, H.G. McCain, R. Lumia, *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*, NBS tech. note 1235, Gaithersburg, MD, 1987.
4. C.H. An, C.G. Atkeson, J.M. Hollerbach, *Model-Based Control of a Robot Manipulator*, MIT Press, Cambridge, MA, 1988.
5. R.J. Anderson, M.W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 549–556, 1988.
6. J. Angeles, *Spatial Kinematic Chains: Analysis, Synthesis, Optimization*, Springer-Verlag, Berlin, 1982.
7. S. Arimoto, F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," in *Robotics Research: The First International Symposium*, M. Brady, R. Paul (Eds.), MIT Press, Cambridge, MA, pp. 783–799, 1984.
8. R.C. Arkin, *Behavior-Based Robotics*, MIT Press, Cambridge, MA, 1998.
9. B. Armstrong-Hélouvry, *Control of Machines with Friction*, Kluwer, Boston, MA, 1991.
10. H. Asada, J.-J.E. Slotine, *Robot Analysis and Control*, Wiley, New York, 1986.
11. H. Asada, K. Youcef-Toumi, "Analysis and design of a direct-drive arm with a five-bar-link parallel drive mechanism," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 225–230, 1984.
12. H. Asada, K. Youcef-Toumi, *Direct-Drive Robots*, MIT Press, Cambridge, MA, 1987.
13. C.G. Atkeson, C.H. An, J.M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.
14. J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proc. 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, pp. 722–728, 1985.

15. A. Balestrino, G. De Maria, L. Sciavicco, "An adaptive model following control for robotic manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 105, pp. 143–151, 1983.
16. A. Balestrino, G. De Maria, L. Sciavicco, B. Siciliano, "An algorithmic approach to coordinate transformation for robotic manipulators," *Advanced Robotics*, vol. 2, pp. 327–344, 1988.
17. J. Barraquand, J.-C. Latombe, "Robot motion planning: A distributed representation approach," *International Journal of Robotics Research*, vol. 10, pp. 628–649, 1991.
18. G. Bastin, G. Campion, B. D'Andréa-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 47–62, 1996.
19. A.K. Bejczy, *Robot Arm Dynamics and Control*, memo. TM 33-669, Jet Propulsion Laboratory, California Institute of Technology, 1974.
20. W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, Orlando, FL, 1986.
21. J. Borenstein, H.R. Everett, L. Feng, *Navigating Mobile Robots: Systems and Techniques*, A K Peters, Wellesley, MA, 1996.
22. B.K.K. Bose, *Modern Power Electronics and AC Drives*, Prentice-Hall, Englewood Cliffs, NJ, 2001.
23. O. Bottema, B. Roth, *Theoretical Kinematics*, North Holland, Amsterdam, 1979.
24. T.L. Boullion, P.L. Odell, *Generalized Inverse Matrices*, Wiley, New York, 1971.
25. M. Brady, "Artificial intelligence and robotics," *Artificial Intelligence*, vol. 26, pp. 79–121, 1985.
26. M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Pérez, M.T. Mason, (Eds.), *Robot Motion: Planning and Control*, MIT Press, Cambridge, MA, 1982.
27. H. Bruyninckx, J. De Schutter, "Specification of force-controlled actions in the "task frame formalism" — A synthesis," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 581–589, 1996.
28. H. Bruyninckx, S. Dumey, S. Dutré, J. De Schutter, "Kinematic models for model-based compliant motion in the presence of uncertainty," *International Journal of Robotics Research*, vol. 14, pp. 465–482, 1995.
29. F. Caccavale, P. Chiacchio, "Identification of dynamic parameters and feedforward control for a conventional industrial manipulator," *Control Engineering Practice*, vol. 2, pp. 1039–1050, 1994.
30. F. Caccavale, C. Natale, B. Siciliano, L. Villani, "Resolved-acceleration control of robot manipulators: A critical review with experiments," *Robotica*, vol. 16, pp. 565–573, 1998.
31. F. Caccavale, C. Natale, B. Siciliano, L. Villani, "Six-DOF impedance control based on angle/axis representations," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 289–300, 1999.
32. F. Caccavale, C. Natale, B. Siciliano, L. Villani, "Robot impedance control with nondiagonal stiffness," *IEEE Transactions on Automatic Control*, vol. 44, pp. 1943–1946, 1999.
33. J.F. Canny, *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA, 1988.

34. C. Canudas de Wit, H. Khennouf, C. Samson, O.J. Sørdalen, "Nonlinear control design for mobile robots," in *Recent Trends in Mobile Robots*, Y.F. Zheng, (Ed.), pp. 121–156, World Scientific Publisher, Singapore, 1993.
35. F. Chaumette, "Image moments: A general and useful set of features for visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 21, pp. 1116-1127, 2005.
36. F. Chaumette, S. Hutchinson, "Visual servo control. Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
37. P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *International Journal of Robotics Research*, vol. 10, pp. 410–425, 1991.
38. P. Chiacchio, S. Chiaverini, L. Sciavicco, B. Siciliano, "Influence of gravity on the manipulability ellipsoid for robot arms," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 723–727, 1992.
39. P. Chiacchio, F. Pierrot, L. Sciavicco, B. Siciliano, "Robust design of independent joint controllers with experimentation on a high-speed parallel robot," *IEEE Transactions on Industrial Electronics*, vol. 40, pp. 393–403, 1993.
40. S. Chiaverini, L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 4, pp. 361–373, 1993.
41. S. Chiaverini, B. Siciliano, "The unit quaternion: A useful tool for inverse kinematics of robot manipulators," *Systems Analysis Modelling Simulation*, vol. 35, pp. 45–60, 1999.
42. S. Chiaverini, B. Siciliano, O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *IEEE Transactions on Control Systems Technology*, vol. 2, pp. 123–134, 1994.
43. S. Chiaverini, B. Siciliano, L. Villani, "Force/position regulation of compliant robot manipulators," *IEEE Transactions on Automatic Control*, vol. 39, pp. 647–652, 1994.
44. S.L. Chiu, "Task compatibility of manipulator postures," *International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.
45. H. Choset, K.M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki, S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.
46. J.C.K. Chou, "Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 53–64, 1992.
47. A.I. Comport, E. Marchand, M. Pressigout, F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 615–628, 2006.
48. P.I. Corke, *Visual Control of Robots: High-Performance Visual Servoing*, Research Studies Press, Taunton, UK, 1996.
49. P. Corke, S. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 507-515, 2001.
50. M. Corless, G. Leitmann, "Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems," *IEEE Transactions on Automatic Control*, vol. 26, pp. 1139–1144, 1981.

51. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA, 2001.
52. J.J. Craig, *Adaptive Control of Mechanical Manipulators*, Addison-Wesley, Reading, MA, 1988.
53. J.J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed., Pearson Prentice Hall, Upper Saddle River, NJ, 2004.
54. C. De Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
55. T.L. De Fazio, D.S. Seltzer, D.E. Whitney, "The instrumented Remote Center of Compliance," *Industrial Robot*, vol. 11, pp. 238–242, 1984.
56. A. De Luca, *A Spline Generator for Robot Arms*, tech. rep. RAL 68, Rensselaer Polytechnic Institute, Department of Electrical, Computer, and Systems Engineering, 1986.
57. A. De Luca, C. Manes, "Modeling robots in contact with a dynamic environment," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 542–548, 1994.
58. A. De Luca, G. Oriolo, C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond, (Ed.), Springer-Verlag, Berlin, Germany, 1998.
59. A. De Luca, G. Oriolo, B. Siciliano, "Robot redundancy resolution at the acceleration level," *Laboratory Robotics and Automation*, vol. 4, pp. 97–106, 1992.
60. J. Denavit, R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *ASME Journal of Applied Mechanics*, vol. 22, pp. 215–221, 1955.
61. P.M. DeRusso, R.J. Roy, C.M. Close, A.A. Desrochers, *State Variables for Engineers*, 2nd ed., Wiley, New York, 1998.
62. J. De Schutter, H. Bruyninckx, S. Dutré, J. De Geeter, J. Katupitiya, S. Demey, T. Lefebvre, "Estimating first-order geometric parameters and monitoring contact transitions during force-controlled compliant motions," *International Journal of Robotics Research*, vol. 18, pp. 1161–1184, 1999.
63. J. De Schutter, H. Bruyninckx, W.-H. Zhu, M.W. Spong, "Force control: A bird's eye view," in *Control Problems in Robotics and Automation*, B. Siciliano, K.P. Valavanis, (Ed.), pp. 1–17, Springer-Verlag, London, UK, 1998.
64. J. De Schutter, H. Van Brussel, "Compliant robot motion I. A formalism for specifying compliant motion tasks," *International Journal of Robotics Research*, vol. 7, no. 4, pp. 3–17, 1988.
65. J. De Schutter, H. Van Brussel, "Compliant robot motion II. A control approach based on external control loops," *International Journal of Robotics Research*, vol. 7, no. 4, pp. 18–33, 1988.
66. K.L. Doty, C. Melchiorri, C. Bonivento, "A theory of generalized inverses applied to robotics," *International Journal of Robotics Research*, vol. 12, pp. 1–19, 1993.
67. S. Dubowsky, D.T. DesForges, "The application of model referenced adaptive control to robotic manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 101, pp. 193–200, 1979.
68. C. Edwards, L. Galloway, "A single-point calibration technique for a six-degree–of–freedom articulated arm," *International Journal of Robotics Research*, vol. 13, pp. 189–199, 1994.
69. O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 471–475, 1987.

70. S.D. Eppinger, W.P. Seering, "Introduction to dynamic models for robot force control," *IEEE Control Systems Magazine*, vol. 7, no. 2, pp. 48–52, 1987.
71. B. Espiau, F. Chaumette, P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 313–326, 1992.
72. H.R. Everett, *Sensors for Mobile Robots: Theory and Application*, AK Peters, Wellesley, MA, 1995.
73. G.E. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, 5th ed., Morgan Kaufmann Publishers, San Francisco, CA, 2001.
74. E.D. Fasse, P.C. Breedveld, "Modelling of elastically coupled bodies: Parts I–II", *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 120, pp. 496–506, 1998.
75. O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Boston, MA, 1993.
76. R. Featherstone, "Position and velocity transformations between robot end-effector coordinates and joint angles," *International Journal of Robotics Research*, vol. 2, no. 2, pp. 35–45, 1983.
77. R. Featherstone, *Robot Dynamics Algorithms*, Kluwer, Boston, MA, 1987.
78. R. Featherstone, O. Khatib, "Load independence of the dynamically consistent inverse of the Jacobian matrix," *International Journal of Robotics Research*, vol. 16, pp. 168–170, 1997.
79. J. Feddema, O. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 691–700, 1989.
80. M. Fliess, J. Lévine, P. Martin, P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, pp. 1327–1361, 1995.
81. J. Fraden, *Handbook of Modern Sensors: Physics, Designs, and Applications*, Springer, New York, 2004.
82. G.F. Franklin, J.D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 5th ed., Prentice-Hall, Lebanon, IN, 2005.
83. E. Freund, "Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 65–78, 1982.
84. L.-C. Fu, T.-L. Liao, "Globally stable robust tracking of nonlinear systems using variable structure control with an application to a robotic manipulator," *IEEE Transactions on Automatic Control*, vol. 35, pp. 1345–1350, 1990.
85. M. Gautier, W. Khalil, "Direct calculation of minimum set of inertial parameters of serial robots," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 368–373, 1990.
86. A.A. Goldenberg, B. Benhabib, R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal of Robotics and Automation*, vol. 1, pp. 14–20, 1985.
87. H. Goldstein, C.P. Poole, J.L. Safko, *Classical Mechanics*, 3rd ed., Addison-Wesley, Reading, MA, 2002.
88. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
89. M.C. Good, L.M. Sweet, K.L. Strobel, "Dynamic models for control system design of integrated robot and drive systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 53–59, 1985.

90. D.M. Gorinevski, A.M. Formalsky, A.Yu. Schneider, *Force Control of Robotics Systems*, CRC Press, Boca Raton, FL, 1997.
91. W.A. Gruver, B.I. Soroka, J.J. Craig, T.L. Turner, "Industrial robot programming languages: A comparative evaluation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 14, pp. 565–570, 1984.
92. G. Hager, W. Chang, A. Morse, "Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 30–39, 1995.
93. R.M. Haralick, L.G. Shapiro, *Computer and Robot Vision*, vols. 1 & 2, Addison-Wesley, Reading, MA, 1993.
94. S. Helgason, *Differential Geometry and Symmetric Spaces*, Academic Press, New York, NY, 1962.
95. N. Hogan, "Impedance control: An approach to manipulation: Part I — Theory," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–7, 1985.
96. J.M. Hollerbach, "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, pp. 730–736, 1980.
97. J.M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 106, pp. 102–106, 1984.
98. J.M. Hollerbach, "A survey of kinematic calibration," in *The Robotics Review 1*, O. Khatib, J.J. Craig, and T. Lozano-Pérez (Eds.), MIT Press, Cambridge, MA, pp. 207–242, 1989.
99. J.M. Hollerbach, G. Sahar, "Wrist-partitioned inverse kinematic accelerations and manipulator dynamics," *International Journal of Robotics Research*, vol. 2, no. 4, pp. 61–76, 1983.
100. R. Horowitz, M. Tomizuka, "An adaptive control scheme for mechanical manipulators — Compensation of nonlinearity and decoupling control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 127–135, 1986.
101. T.C.S. Hsia, T.A. Lasky, Z. Guo, "Robust independent joint controller design for industrial robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 38, pp. 21–25, 1991.
102. P. Hsu, J. Hauser, S. Sastry, "Dynamic control of redundant manipulators," *Journal of Robotic Systems*, vol. 6, pp. 133–148, 1989.
103. S. Hutchinson, G. Hager, P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 651–670, 1996.
104. A. Isidori, *Nonlinear Control Systems*, 3rd ed., Springer-Verlag, London, UK, 1995.
105. H. Kazerooni, P.K. Houpt, T.B. Sheridan, "Robust compliant motion of manipulators, Part I: The fundamental concepts of compliant motion," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 83–92, 1986.
106. J.L. Jones, A.M. Flynn, *Mobile Robots: Inspiration to Implementation*, AK Peters, Wellesley, MA, 1993.
107. L.E. Kavraki, P. Svestka, J.-C. Latombe, M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, 1996.

108. R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, pp. 39–48, 2000.
109. H.K. Khalil, *Nonlinear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 2002.
110. W. Khalil, F. Bennis, "Symbolic calculation of the base inertial parameters of closed-loop robots," *International Journal of Robotics Research*, vol. 14, pp. 112–128, 1995.
111. W. Khalil, E. Dombre, *Modeling, Identification and Control of Robots*, Hermes Penton Ltd, London, 2002.
112. W. Khalil, J.F. Kleinfinger, "Minimum operations and minimum parameters of the dynamic model of tree structure robots," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 517–526, 1987.
113. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
114. O. Khatib, "A unified approach to motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 43–53, 1987.
115. P.K. Khosla, "Categorization of parameters in the dynamic robot model," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 261–268, 1989.
116. P.K. Khosla, T. Kanade, "Parameter identification of robot dynamics," in *Proceedings of 24th IEEE Conference on Decision and Control*, Fort Lauderdale, FL, pp. 1754–1760, 1985.
117. P.K. Khosla, T. Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators," *International Journal of Robotics Research*, vol. 7, no. 1, pp. 18–28, 1988.
118. C.A. Klein, C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 245–250, 1983.
119. D.E. Koditschek, "Natural motion for robot arms," *Proc. 23th IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 733–735, 1984.
120. A.J. Koivo, *Fundamentals for Control of Robotic Manipulators*, Wiley, New York, 1989.
121. K. Kreutz, "On manipulator control by exact linearization," *IEEE Transactions on Automatic Control*, vol. 34, pp. 763–767, 1989.
122. J.-C. Latombe, *Robot Motion Planning*, Kluwer, Boston, MA, 1991.
123. J.-P. Laumond, (Ed.), *Robot Motion Planning and Control*, Springer-Verlag, Berlin, 1998.
124. S.M. LaValle, *Planning Algorithms*, Cambridge University Press, New York, 2006.
125. S.M. LaValle, J.J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *New Directions in Algorithmic and Computational Robotics*, B.R. Donald, K. Lynch, D. Rus, (Eds.), AK Peters, Wellesley, MA, pp. 293–308, 2001.
126. M.B. Leahy, G.N. Saridis, "Compensation of industrial manipulator dynamics," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 73–84, 1989.
127. C.S.G. Lee, "Robot kinematics, dynamics and control," *IEEE Computer*, vol. 15, no. 12, pp. 62–80, 1982.
128. W. Leonhard, *Control of Electrical Drives*, Springer-Verlag, New York, 2001.

129. A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 868–871, 1977.
130. K.Y. Lim, M. Eslami, "Robust adaptive controller designs for robot manipulator systems," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 54–66, 1987.
131. C.S. Lin, P.R. Chang, J.Y.S. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on Automatic Control*, vol. 28, pp. 1066–1073, 1983.
132. S.K. Lin, "Singularity of a nonlinear feedback control scheme for robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, pp. 134–139, 1989.
133. H. Lipkin, J. Duffy, "Hybrid twist and wrench control for a robotic manipulator," *ASME Journal of Mechanism, Transmissions, and Automation Design*, vol. 110, pp. 138–144, 1988.
134. V. Lippiello, B. Siciliano, L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Transactions on Robotics and Automation*, vol. 23, pp. 73–86, 2007.
135. D.A. Lizárraga, "Obstructions to the existence of universal stabilizers for smooth control systems," *Mathematics of Control, Signals, and Systems*, vol. 16, pp. 255–277, 2004.
136. J. Lončarić, "Normal forms of stiffness and compliance matrices," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 567–572, 1987.
137. T. Lozano-Pérez, "Automatic planning of manipulator transfer movements," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 681–698, 1981.
138. T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computing*, vol. 32, pp. 108–120, 1983.
139. T. Lozano-Pérez, "Robot programming," *Proceedings IEEE*, vol. 71, pp. 821–841, 1983.
140. T. Lozano-Pérez, M.T. Mason, R.H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
141. J.Y.S. Luh, "Conventional controller design for industrial robots: A tutorial," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 298–316, 1983.
142. J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "On-line computational scheme for mechanical manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, 1980.
143. J.Y.S. Luh, M.W. Walker, R.P.C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, pp. 468–474, 1980.
144. J.Y.S. Luh, Y.-F. Zheng, "Computation of input generalized forces for robots with closed kinematic chain mechanisms," *IEEE Journal of Robotics and Automation* vol. 1, pp. 95–103, 1985.
145. V.J. Lumelsky, *Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World*, Wiley, Hoboken, NJ, 2006.
146. Y. Ma, S. Soatto, J. Kosecka, S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, Springer, New York, 2003.

147. A.A. Maciejewski, C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.
148. E. Malis, F. Chaumette, S. Boudet, "2-1/2D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 238–250, 1999.
149. B.R. Markiewicz, *Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator*, memo. TM 33-601, JPL, Pasadena, CA, 1973.
150. M.T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 418–432, 1981.
151. J.M. McCarthy, *An Introduction to Theoretical Kinematics*, MIT Press, Cambridge, MA, 1990.
152. N.H. McClamroch, D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Transactions on Automatic Control*, vol. 33, pp. 419–426, 1988.
153. R.T. M'Closkey, R.M. Murray, "Exponential stabilization of driftless nonlinear control systems using homogeneous feedback," *IEEE Transactions on Automatic Control*, vol. 42, pp. 614–628, 1997.
154. L. Meirovitch, *Dynamics and Control of Structures*, Wiley, New York, 1990.
155. C. Melchiorri, *Traiettorie per Azionamenti Elettrici*, Progetto Leonardo, Bologna, I, 2000.
156. N. Manring, *Hydraulic Control Systems*, Wiley, New York, 2005.
157. R. Middleton, G.C. Goodwin, "Adaptive computed torque control for rigid link manipulators," *Systems & Control Letters*, vol. 10, pp. 9–16, 1988.
158. R.R. Murphy, *Introduction to AI Robotics*, MIT Press, Cambridge, MA, 2000.
159. R.M. Murray, Z. Li, S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, CA. 1994.
160. Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, Addison-Wesley, Reading, MA, 1991.
161. Y. Nakamura, H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.
162. Y. Nakamura, H. Hanafusa, "Optimal redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.
163. Y. Nakamura, H. Hanafusa, T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
164. J.I. Neimark, F.A. Fufaev, *Dynamics of Nonholonomic Systems*, American Mathematical Society, Providence, RI, 1972.
165. I. Nevins, D.E. Whitney, "The force vector assembler concept," *Proc. First CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Udine, I, 1973.
166. F. Nicolò, J. Katende, "A robust MRAC for industrial robots," *Proc. 2nd IASTED International Symposium on Robotics and Automation*, pp. 162–171, Lugano, Switzerland, 1983.
167. S. Nicosia, P. Tomei, "Model reference adaptive control algorithms for industrial robots," *Automatica*, vol. 20, pp. 635–644, 1984.

168. H. Nijmeijer, A. van de Schaft, *Nonlinear Dynamical Control Systems*, Springer-Verlag, Berlin, Germany, 1990.
169. B. Noble, *Applied Linear Algebra*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1987.
170. C. O'Dúnlaing, C.K. Yap, "A retraction method for planning the motion of a disc," *Journal of Algorithms*, vol. 6, pp. 104–111, 1982.
171. K. Ogata, *Modern Control Engineering*, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.
172. D.E. Orin, R.B. McGhee, M. Vukobratović, G. Hartoch, "Kinematic and kinetic analysis of open-chain linkages utilizing Newton–Euler methods," *Mathematical Biosciences* vol. 43, pp. 107–130, 1979.
173. D.E. Orin, W.W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 66–75, 1984.
174. G. Oriolo, A. De Luca, M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 835–852, 2002.
175. R. Ortega, M.W. Spong, "Adaptive motion control of rigid robots: A tutorial," *Automatica*, vol. 25, pp. 877–888, 1989.
176. T. Patterson, H. Lipkin, "Duality of constrained elastic manipulation," *Proc. 1991 IEEE International Conference on Robotics and Automation*, pp. 2820–2825, Sacramento, CA, 1991.
177. T. Patterson, H. Lipkin, "Structure of robot compliance," *ASME Journal of Mechanical Design*, vol. 115, pp. 576–580, 1993.
178. R.P. Paul, *Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm*, memo. AIM 177, Stanford Artificial Intelligence Laboratory, 1972.
179. R.P. Paul, "Manipulator Cartesian path control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 702–711, 1979.
180. R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, Cambridge, MA, 1981.
181. R.P. Paul, B.E. Shimano, G. Mayer, "Kinematic control equations for simple manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 449–455, 1981.
182. R.P. Paul, H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *International Journal of Robotics Research*, vol. 5, no. 2, pp. 32–44, 1986.
183. D.L. Pieper, *The Kinematics of Manipulators Under Computer Control* memo. AIM 72, Stanford Artificial Intelligence Laboratory, 1968.
184. M.H. Raibert, J.J. Craig, "Hybrid position/force control of manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 103, pp. 126–133, 1981.
185. E. Rimon, D.E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Proc. 1989 IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, pp. 21–26, 1989.
186. E.I. Rivin, *Mechanical Design of Robots*, McGraw-Hill, New York, 1987.
187. R.E. Roberson, R. Schwertassek, *Dynamics of Multibody Systems*, Springer-Verlag, Berlin, Germany, 1988.
188. Z. Roth, B.W. Mooring, B. Ravani, "An overview of robot calibration," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 377–386, 1987.

189. S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed., Prentice Hall, Englewood Cliffs, NJ, 2003.
190. J.K. Salisbury, "Active stiffness control of a manipulator in Cartesian coordinates," *Proc. 19th IEEE Conference on Decision and Control*, pp. 95–100, Albuquerque, NM, 1980.
191. J.K. Salisbury, J.J. Craig, "Articulated hands: Force control and kinematic issues," *International Journal of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.
192. C. Samson, "Robust control of a class of nonlinear systems and applications to robotics," *International Journal of Adaptive Control and Signal Processing*, vol. 1, pp. 49–68, 1987.
193. C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.
194. C. Samson, M. Le Borgne, B. Espiau, *Robot Control: The Task Function Approach*, Clarendon Press, Oxford, UK, 1991.
195. S. Sastry, *Nonlinear Systems: Analysis, Stability and Control*, Springer-Verlag, Berlin, Germany, 1999.
196. V.D. Scheinman, *Design of a Computer Controlled Manipulator*, memo. AIM 92, Stanford Artificial Intelligence Laboratory, 1969.
197. J.T. Schwartz, M. Sharir, "On the 'piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, vol. 4, pp. 298–351, 1983.
198. L. Sciavicco, B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, pp. 550–559, 1986.
199. L. Sciavicco, B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 403–410, 1988.
200. L. Sciavicco, B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd ed., Springer, London, UK, 2000.
201. L. Sciavicco, B. Siciliano, L. Villani, "Lagrange and Newton–Euler dynamic modeling of a gear-driven rigid robot manipulator with inclusion of motor inertia effects," *Advanced Robotics*, vol. 10, pp. 317–334, 1996.
202. R. Sedgewick, *Algorithms*, 2nd ed., Addison-Wesley, Reading, MA, 1988.
203. H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 472–490, 1989.
204. S.W. Shepperd, S.W., "Quaternion from rotation matrix," *AIAA Journal of Guidance and Control*, vol. 1, pp. 223–224, 1978.
205. R. Shoureshi, M.E. Momot, M.D. Roesler, "Robust control for manipulators with uncertain dynamics," *Automatica*, vol. 26, pp. 353–359, 1990.
206. B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *Journal of Intelligent and Robotic Systems*, vol. 3, pp. 201–212, 1990.
207. B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint based robot control," *Robotica*, vol. 8, pp. 231–243, 1990.
208. B. Siciliano, J.-J.E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Proc. 5th International Conference on Advanced Robotics*, Pisa, I, pp. 1211–1216, 1991.

209. B. Siciliano, L. Villani, *Robot Force Control*, Kluwer, Boston, MA, 2000.
210. R. Siegwart, I.R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, MA, 2004.
211. D.B. Silver, "On the equivalence of Lagrangian and Newton–Euler dynamics for manipulators," *International Journal of Robotics Research*, vol. 1, no. 2, pp. 60–70, 1982.
212. J.-J.E. Slotine, "The robust control of robot manipulators," *International Jorunal of Robotics Research*, vol. 4, no. 2, pp. 49–64, 1985.
213. J.-J.E. Slotine, "Putting physics in control — The example of robotics," *IEEE Control Systems Magazine*, vol. 8, no. 6, pp. 12–18, 1988.
214. J.-J.E. Slotine, W. Li, "On the adaptive control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.
215. J.-J.E. Slotine, W. Li, *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
216. M.W. Spong, "On the robust control of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1782–1786, 1992.
217. M.W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, Wiley, New York, 2006.
218. M.W. Spong, R. Ortega, R. Kelly, "Comments on "Adaptive manipulator control: A case study"," *IEEE Transactions on Automatic Control*, vol. 35, pp. 761–762, 1990.
219. M.W. Spong, M. Vidyasagar, "Robust linear compensator design for nonlinear robotic control," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 345–351, 1987.
220. SRI International, *Robot Design Handbook*, G.B. Andeen, (Ed.), McGraw-Hill, New York, 1988.
221. Y. Stepanenko, M. Vukobratović, "Dynamics of articulated open-chain active mechanisms," *Mathematical Biosciences*, vol. 28, pp. 137–170, 1976.
222. Y. Stepanenko, J. Yuan, "Robust adaptive control of a class of nonlinear mechanical systems with unbounded and fast varying uncertainties," *Automatica*, vol. 28, pp. 265–276, 1992.
223. S. Stramigioli, *Modeling and IPC Control of Interactive Mechanical Systems — A Coordinate Free Approach*, Springer, London, UK, 2001.
224. K.R. Symon, *Mechanics*, 3rd ed., Addison-Wesley, Reading, MA, 1971.
225. K. Takase, R. Paul, E. Berg, "A structured approach to robot programming and teaching," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, pp. 274–289, 1981.
226. M. Takegaki, S. Arimoto, "A new feedback method for dynamic control of manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 119–125, 1981.
227. T.-J. Tarn, A.K. Bejczy, X. Yun, Z. Li, "Effect of motor dynamics on nonlinear feedback robot arm control," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 114–122, 1991.
228. T.-J. Tarn, Y. Wu, N. Xi, A. Isidori, "Force regulation and contact transition control," *IEEE Control Systems Magazine*, vol. 16, no. 1, pp. 32–40, 1996.
229. R.H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM Journal of Research and Development*, vol. 23, pp. 424–436, 1979.
230. R.H. Taylor, D.D. Grossman, "An integrated robot system architecture," *Proceedings IEEE*, vol. 71, pp. 842–856, 1983.

231. S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.
232. L.W. Tsai, A.P. Morgan, "Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods," *ASME Journal of Mechanisms, Transmission, and Automation in Design*, vol. 107, pp. 189–200, 1985.
233. R. Tsai, "A versatile camera calibration technique for high accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Transactions on Robotics and Automation*, vol. 3, pp. 323–344, 1987.
234. J.J.Uicker, "Dynamic force analysis of spatial linkages," *ASME Journal of Applied Mechanics*, vol. 34, pp. 418–424, 1967.
235. L. Villani, C. Canudas de Wit, B. Brogliato, "An exponentially stable adaptive control for force and position tracking of robot manipulators," *IEEE Transactions on Automatic Control*, vol. 44, pp. 798–802, 1999.
236. M. Vukobratović, "Dynamics of active articulated mechanisms and synthesis of artificial motion," *Mechanism and Machine Theory*, vol. 13, pp. 1–56, 1978.
237. M.W. Walker, D.E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 205–211, 1982.
238. C.W. Wampler, "Manipulator inverse kinematic solutions based on damped least-squares solutions," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, pp. 93–101, 1986.
239. L. Weiss, A. Sanderson, C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 404–417, 1987.
240. D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Transactions on Man-Machine Systems*, vol. 10, pp. 47–53, 1969.
241. D.E. Whitney, "Force feedback control of manipulator fine motions," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 99, pp. 91–97, 1977.
242. D.E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 65–77, 1982.
243. D.E. Whitney, "Historical perspective and state of the art in robot force control," *International Journal of Robotics Research*, vol. 6, no. 1, pp. 3–14, 1987.
244. W. Wilson, C. Hulls, G. Bell, "Relative end-effector control using Cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 684–696, 1996.
245. T. Yoshikawa, "Manipulability of robotic mechanisms," *International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.
246. T. Yoshikawa, "Dynamic manipulability ellipsoid of robot manipulators," *Journal of Robotic Systems*, vol. 2, pp. 113–124, 1985.
247. T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators — Description of hand constraints and calculation of joint driving force," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 386–392, 1987.
248. T. Yoshikawa, *Foundations of Robotics*, MIT Press, Boston, MA, 1990.
249. T. Yoshikawa, T. Sugie, N. Tanaka, "Dynamic hybrid position/force control of robot manipulators — Controller design and experiment," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 699–705, 1988.

250. J.S.-C. Yuan, "Closed-loop manipulator control using quaternion feedback," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 434–440, 1988.

Index